



TR-4998: Oracle HA in AWS EC2 mit Pacemaker Clustering und FSx ONTAP

NetApp database solutions

NetApp
August 18, 2025

Inhalt

TR-4998: Oracle HA in AWS EC2 mit Pacemaker Clustering und FSx ONTAP	1
Zweck	1
Publikum	1
Test- und Validierungsumgebung für Lösungen	2
Architektur	2
Hardware- und Softwarekomponenten	2
Aktiv/Passiv-Konfiguration der Oracle-Datenbank in der AWS EC2/FSx-Laborumgebung	3
Wichtige Faktoren für die Bereitstellungsüberlegungen	3
Lösungsbereitstellung	4
Voraussetzungen für die Bereitstellung	4
Bereitstellung von EC2-Instances und Amazon FSx ONTAP Speicherclustern	4
Pacemaker-Cluster-Setup	7
Pacemaker-Cluster-Fencing-Konfiguration	12
Bereitstellen einer Oracle-Datenbank im PCS-Cluster	15
Konfigurieren von Oracle-Ressourcen für die PCS-Verwaltung	30
HA-Validierung nach der Bereitstellung	40
Oracle-Backup, -Wiederherstellung und -Klonen mit SnapCenter	52
Wo Sie weitere Informationen finden	53

TR-4998: Oracle HA in AWS EC2 mit Pacemaker Clustering und FSx ONTAP

Allen Cao, Niyaz Mohamed, NetApp

Diese Lösung bietet eine Übersicht und Details zum Aktivieren der Oracle-Hochverfügbarkeit (HA) in AWS EC2 mit Pacemaker-Clustering auf Redhat Enterprise Linux (RHEL) und Amazon FSx ONTAP für die Datenbankspeicher-HA über das NFS-Protokoll.

Zweck

Viele Kunden, die Oracle in der öffentlichen Cloud selbst verwalten und ausführen möchten, müssen einige Herausforderungen bewältigen. Eine dieser Herausforderungen besteht darin, eine hohe Verfügbarkeit für die Oracle-Datenbank zu gewährleisten. Traditionell verlassen sich Oracle-Kunden auf eine Oracle-Datenbankfunktion namens „Real Application Cluster“ oder RAC für die Unterstützung von Aktiv-Aktiv-Transaktionen auf mehreren Clusterknoten. Ein ausgefallener Knoten würde die Anwendungsverarbeitung nicht blockieren. Leider ist die Oracle RAC-Implementierung in vielen gängigen öffentlichen Clouds wie AWS EC2 nicht ohne Weiteres verfügbar oder wird nicht unterstützt. Durch die Nutzung des integrierten Pacemaker-Clusterings (PCS) in RHEL und Amazon FSx ONTAP erhalten Kunden ohne Oracle RAC-Lizenzkosten eine praktikable Alternative für Aktiv-Passiv-Clustering sowohl auf der Datenverarbeitung als auch auf der Speicherung, um unternehmenskritische Oracle-Datenbank-Workloads in der AWS-Cloud zu unterstützen.

Diese Dokumentation zeigt die Details zum Einrichten des Pacemaker-Clusterings auf RHEL, zum Bereitstellen einer Oracle-Datenbank auf EC2 und Amazon FSx ONTAP mit NFS-Protokoll, zum Konfigurieren von Oracle-Ressourcen in Pacemaker für HA und zum Abschluss der Demo mit einer Validierung unter den am häufigsten auftretenden HA-Szenarien. Die Lösung bietet außerdem Informationen zum schnellen Sichern, Wiederherstellen und Klonen von Oracle-Datenbanken mit dem NetApp SnapCenter UI-Tool.

Diese Lösung ist für die folgenden Anwendungsfälle geeignet:

- Einrichtung und Konfiguration des Pacemaker HA-Clusters in RHEL.
- Oracle-Datenbank-HA-Bereitstellung in AWS EC2 und Amazon FSx ONTAP.

Publikum

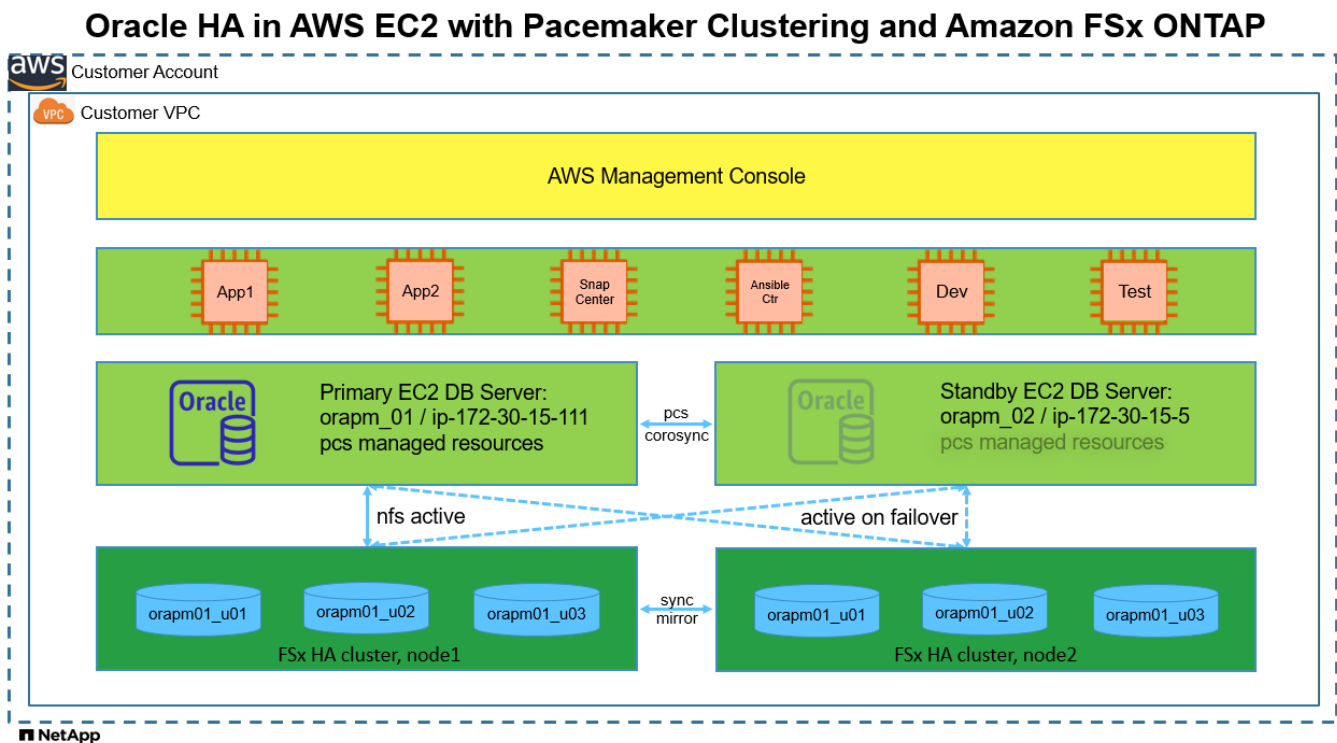
Diese Lösung ist für folgende Personen gedacht:

- Ein DBA, der Oracle in AWS EC2 und Amazon FSx ONTAP bereitstellen möchte.
- Ein Datenbanklösungsarchitekt, der Oracle-Workloads in AWS EC2 und Amazon FSx ONTAP testen möchte.
- Ein Speicheradministrator, der eine Oracle-Datenbank in AWS EC2 und Amazon FSx ONTAP bereitstellen und verwalten möchte.
- Ein Anwendungsbesitzer, der eine Oracle-Datenbank in AWS EC2 und Amazon FSx ONTAP einrichten möchte.

Test- und Validierungsumgebung für Lösungen

Die Tests und Validierungen dieser Lösung wurden in einer Laborumgebung durchgeführt, die möglicherweise nicht der endgültigen Bereitstellungsumgebung entspricht. Siehe den Abschnitt [Wichtige Faktoren für die Bereitstellungsüberlegungen](#) für weitere Informationen.

Architektur



Hardware- und Softwarekomponenten

Hardware		
Amazon FSx ONTAP Speicher	Aktuelle von AWS angebotene Version	Single-AZ in us-east-1, 1024 GiB Kapazität, 128 MB/s Durchsatz
EC2-Instanzen für DB-Server	t2.xlarge/4vCPU/16G	Zwei EC2 T2 xlarge EC2-Instanzen, eine als primärer DB-Server und die andere als Standby-DB-Server
VM für Ansible-Controller	4 vCPUs, 16 GiB RAM	Eine Linux-VM zum Ausführen der automatisierten AWS EC2/FSx-Bereitstellung und Oracle-Bereitstellung auf NFS
Software		
RedHat Linux	RHEL Linux 8.6 (LVM) – x64 Gen2	RedHat-Abonnement zum Testen bereitgestellt

Oracle-Datenbank	Version 19.18	RU-Patch p34765931_190000_Linux-x86-64.zip angewendet
Oracle OPatch	Version 12.2.0.1.36	Neuester Patch p6880880_190000_Linux-x86-64.zip
Schrittmacher	Version 0.10.18	Hochverfügbarkeits-Add-On für RHEL 8.0 von RedHat
NFS	Version 3.0	Oracle dNFS aktiviert
Ansible	Kern 2.16.2	Python 3.6.8

Aktiv/Passiv-Konfiguration der Oracle-Datenbank in der AWS EC2/FSx-Laborumgebung

Server	Datenbank	DB-Speicher
Primärknoten: orapm01/ip-172.30.15.111	NTAP(NTAP_PDB1,NTAP_PDB2,NTAP_PDB3)	/u01, /u02, /u03 NFS-Mounts auf Amazon FSx ONTAP Volumes
Standby-Knoten: orapm02/ip-172.30.15.5	NTAP(NTAP_PDB1,NTAP_PDB2,NTAP_PDB3) beim Failover	/u01, /u02, /u03 NFS-Mounts beim Failover

Wichtige Faktoren für die Bereitstellungsüberlegungen

- *** Amazon FSx ONTAP HA.*** Amazon FSx ONTAP wird standardmäßig in einem HA-Paar von Speichercontrollern in einer oder mehreren Verfügbarkeitszonen bereitgestellt. Es bietet Speicherredundanz auf aktive/passive Weise für unternehmenskritische Datenbank-Workloads. Das Speicher-Failover ist für den Endbenutzer transparent. Im Falle eines Speicher-Failovers ist kein Benutzereingriff erforderlich.
- **PCS-Ressourcengruppe und Ressourcenbestellung.** Eine Ressourcengruppe ermöglicht die Ausführung mehrerer abhängiger Ressourcen auf demselben Clusterknoten. Die Ressourcenreihenfolge erzwingt die Startreihenfolge der Ressourcen und die Reihenfolge beim Herunterfahren in umgekehrter Reihenfolge.
- **Bevorzugter Knoten.** Der Pacemaker-Cluster wird absichtlich im Aktiv-/Passiv-Clustering bereitgestellt (keine Anforderung von Pacemaker) und ist mit dem FSx ONTAP -Clustering synchronisiert. Die aktive EC2-Instanz wird als bevorzugter Knoten für Oracle-Ressourcen konfiguriert, sofern sie mit einer Standortbeschränkung verfügbar ist.
- **Zaunverzögerung auf Standby-Knoten.** In einem PCS-Cluster mit zwei Knoten wird ein Quorum künstlich auf 1 festgelegt. Im Falle eines Kommunikationsproblems zwischen den Clusterknoten könnte einer der Knoten versuchen, den anderen Knoten abzuschirmen, was möglicherweise zu Datenbeschädigungen führen kann. Durch das Einrichten einer Verzögerung auf dem Standby-Knoten wird das Problem gemildert und es wird ermöglicht, dass der primäre Knoten weiterhin Dienste bereitstellt, während der Standby-Knoten abgeschirmt ist.
- **Überlegung zur Bereitstellung mehrerer AZs.** Die Lösung wird in einer einzigen Verfügbarkeitszone bereitgestellt und validiert. Für die Multi-AZ-Bereitstellung werden zusätzliche AWS-Netzwerkressourcen benötigt, um die PCS-Floating-IP zwischen den Verfügbarkeitszonen zu verschieben.
- **Oracle-Datenbankspeicherlayout.** In dieser Lösungsdemonstration stellen wir drei Datenbankvolumes für die Testdatenbank NTAP bereit, um Oracle-Binärdateien, Daten und Protokolle zu hosten. Die Volumes

werden auf dem Oracle DB-Server als /u01 – Binär, /u02 – Daten und /u03 – Protokoll über NFS gemountet. Aus Redundanzgründen sind auf den Einhängepunkten /u02 und /u03 doppelte Steuerdateien konfiguriert.

- **dNFS-Konfiguration.** Durch die Verwendung von dNFS (verfügbar seit Oracle 11g) kann eine Oracle-Datenbank, die auf einer DB-VM ausgeführt wird, deutlich mehr E/A-Vorgänge ausführen als der native NFS-Client. Bei der automatisierten Oracle-Bereitstellung wird dNFS standardmäßig auf NFSv3 konfiguriert.
- **Datenbanksicherung.** NetApp bietet eine SnapCenter software zum Sichern, Wiederherstellen und Klonen von Datenbanken mit einer benutzerfreundlichen Benutzeroberfläche. NetApp empfiehlt die Implementierung eines solchen Verwaltungstools, um schnelle Snapshot-Backups (unter einer Minute), schnelle Datenbankwiederherstellungen (in Minuten) und Datenbankklone zu erreichen.

Lösungsbereitstellung

Die folgenden Abschnitte enthalten schrittweise Anleitungen zur Bereitstellung und Konfiguration von Oracle Database HA in AWS EC2 mit Pacemaker-Clustering und Amazon FSx ONTAP zum Schutz der Datenbankspeicherung.

Voraussetzungen für die Bereitstellung

Für die Bereitstellung sind die folgenden Voraussetzungen erforderlich.

1. Ein AWS-Konto wurde eingerichtet und die erforderlichen VPC- und Netzwerksegmente wurden innerhalb Ihres AWS-Kontos erstellt.
2. Stellen Sie eine Linux-VM als Ansible-Controllerknoten bereit, auf der die neueste Version von Ansible und Git installiert ist. Weitere Einzelheiten finden Sie unter folgendem Link: "[Erste Schritte mit der NetApp Lösungsautomatisierung](#)" im Abschnitt -
`Setup the Ansible Control Node for CLI deployments on RHEL / CentOS oder`
`Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.`

Aktivieren Sie die SSH-Authentifizierung mit öffentlichem/privatem Schlüssel zwischen dem Ansible-Controller und den DB-VMs der EC2-Instanz.

Bereitstellung von EC2-Instances und Amazon FSx ONTAP Speicherclustern

Obwohl EC2-Instanzen und Amazon FSx ONTAP manuell über die AWS-Konsole bereitgestellt werden können, wird empfohlen, das auf NetApp Terraform basierende Automatisierungs-Toolkit zu verwenden, um die Bereitstellung von EC2-Instanzen und FSx ONTAP -Speicherclustern zu automatisieren. Nachfolgend finden Sie die detaillierten Verfahren.

1. Klonen Sie von AWS CloudShell oder Ansible Controller VM eine Kopie des Automatisierungs-Toolkits für EC2 und FSx ONTAP.

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_aws_fsx_ec2_deploy.git
```



Wenn das Toolkit nicht von AWS CloudShell ausgeführt wird, ist eine AWS CLI-Authentifizierung mit Ihrem AWS-Konto unter Verwendung des AWS-Benutzerkontozugriffs/geheimen Schlüsselpaars erforderlich.

2. Überprüfen Sie die im Toolkit enthaltene Datei README.md. Überarbeiten Sie main.tf und die zugehörigen Parameterdateien nach Bedarf für die erforderlichen AWS-Ressourcen.

An example of main.tf:

```
resource "aws_instance" "orapm01" {
  ami                = var.ami
  instance_type      = var.instance_type
  subnet_id          = var.subnet_id
  key_name            = var.ssh_key_name

  root_block_device {
    volume_type      = "gp3"
    volume_size      = var.root_volume_size
  }

  tags = {
    Name              = var.ec2_tag1
  }
}

resource "aws_instance" "orapm02" {
  ami                = var.ami
  instance_type      = var.instance_type
  subnet_id          = var.subnet_id
  key_name            = var.ssh_key_name

  root_block_device {
    volume_type      = "gp3"
    volume_size      = var.root_volume_size
  }
}
```

```

tags = {
  Name          = var.ec2_tag2
}

resource "aws_fsx_ontap_file_system" "fsx_01" {
  storage_capacity      = var.fs_capacity
  subnet_ids            = var.subnet_ids
  preferred_subnet_id   = var.preferred_subnet_id
  throughput_capacity   = var.fs_throughput
  fsx_admin_password    = var.fsxadmin_password
  deployment_type       = var.deployment_type

  disk_iops_configuration {
    iops      = var.iops
    mode      = var.iops_mode
  }

  tags = {
    Name = var.fsx_tag
  }
}

resource "aws_fsx_ontap_storage_virtual_machine" "svm_01" {
  file_system_id =
aws_fsx_ontap_file_system.fsx_01.id
  name           = var.svm_name
  svm_admin_password = var.vsadmin_password
}

```

- Validieren und führen Sie den Terraform-Plan aus. Bei einer erfolgreichen Ausführung würden zwei EC2-Instanzen und ein FSx ONTAP -Speichercluster im AWS-Zielkonto erstellt. Die Automatisierungsausgabe zeigt die IP-Adresse der EC2-Instanz und die Endpunkte des FSx ONTAP Clusters an.

```
terraform plan -out=main.plan
```

```
terraform apply main.plan
```

Damit sind die EC2-Instanzen und die FSx ONTAP -Bereitstellung für Oracle abgeschlossen.

Das High Availability Add-On für RHEL ist ein Clustersystem, das Zuverlässigkeit, Skalierbarkeit und Verfügbarkeit für kritische Produktionsdienste wie Oracle-Datenbankdienste bietet. In dieser Anwendungsfalldemonstration wird ein Pacemaker-Cluster mit zwei Knoten eingerichtet und konfiguriert, um die Hochverfügbarkeit einer Oracle-Datenbank in einem Aktiv/Passiv-Clustering-Szenario zu unterstützen.

Melden Sie sich bei EC2-Instanzen als EC2-Benutzer an und führen Sie die folgenden Aufgaben aus **both** EC2-Instanzen:

1. Entfernen Sie den AWS Red Hat Update Infrastructure (RHUI)-Client.

```
sudo -i yum -y remove rh-amazon-rhui-client*
```

2. Registrieren Sie die EC2-Instanz-VMs bei Red Hat.

```
sudo subscription-manager register --username xxxxxxxx --password  
'xxxxxxx' --auto-attach
```

3. Aktivieren Sie RHEL-RPMs mit hoher Verfügbarkeit.

```
sudo subscription-manager config --rhsm.manage_repos=1
```

```
sudo subscription-manager repos --enable=rhel-8-for-x86_64  
-highavailability-rpms
```

4. Installieren Sie den Herzschrittmacher und den Zaunagenten.

```
sudo yum update -y
```

```
sudo yum install pcs pacemaker fence-agents-aws
```

5. Erstellen Sie auf allen Clusterknoten ein Kennwort für den hacluster-Benutzer. Verwenden Sie für alle Knoten dasselbe Passwort.

```
sudo passwd hacluster
```

6. Starten Sie den PCS-Dienst und aktivieren Sie ihn, damit er beim Booten gestartet wird.

```
sudo systemctl start pcsd.service
```

```
sudo systemctl enable pcsd.service
```

7. Validieren Sie den PCSD-Dienst.

```
sudo systemctl status pcsd
```

```
[ec2-user@ip-172-30-15-5 ~]$ sudo systemctl status pcsd
● pcsd.service - PCS GUI and remote configuration interface
   Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled;
 vendor preset: disabled)
   Active: active (running) since Tue 2024-09-10 18:50:22 UTC; 33s
 ago
     Docs: man:pcsd(8)
           man:pcs(8)
  Main PID: 65302 (pcsd)
    Tasks: 1 (limit: 100849)
  Memory: 24.0M
   CGroup: /system.slice/pcsd.service
           └─65302 /usr/libexec/platform-python -Es /usr/sbin/pcsd

Sep 10 18:50:21 ip-172-30-15-5.ec2.internal systemd[1]: Starting PCS
GUI and remote configuration interface...
Sep 10 18:50:22 ip-172-30-15-5.ec2.internal systemd[1]: Started PCS
GUI and remote configuration interface.
```

8. Fügen Sie Clusterknoten zu Hostdateien hinzu.

```
sudo vi /etc/hosts
```

```
[ec2-user@ip-172-30-15-5 ~]$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhostdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localhostdomain6

# cluster nodes
172.30.15.111 ip-172-30-15-111.ec2.internal
172.30.15.5   ip-172-30-15-5.ec2.internal
```

9. Installieren und konfigurieren Sie awscli für die Verbindung zum AWS-Konto.

```
sudo yum install awscli
```

```
sudo aws configure
```

```
[ec2-user@ip-172-30-15-111 ]# sudo aws configure
AWS Access Key ID [None]: XXXXXXXXXXXXXXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXX
Default region name [None]: us-east-1
Default output format [None]: json
```

10. Installieren Sie das Paket „resource-agents“, falls es noch nicht installiert ist.

```
sudo yum install resource-agents
```

An only one Führen Sie die folgenden Aufgaben des Clusterknotens aus, um einen PC-Cluster zu erstellen.

1. Authentifizieren Sie den PC-Benutzer hacluster.

```
sudo pcs host auth ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal
```

```
[ec2-user@ip-172-30-15-111 ~]$ sudo pcs host auth ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal
Username: hacluster
Password:
ip-172-30-15-111.ec2.internal: Authorized
ip-172-30-15-5.ec2.internal: Authorized
```

2. Erstellen Sie den PC-Cluster.

```
sudo pcs cluster setup ora_ec2nfsx ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal
```

```
[ec2-user@ip-172-30-15-111 ~]$ sudo pcs cluster setup ora_ec2nfsx
ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal
No addresses specified for host 'ip-172-30-15-5.ec2.internal', using
'ip-172-30-15-5.ec2.internal'
No addresses specified for host 'ip-172-30-15-111.ec2.internal',
using 'ip-172-30-15-111.ec2.internal'
Destroying cluster on hosts: 'ip-172-30-15-111.ec2.internal', 'ip-
172-30-15-5.ec2.internal'...
ip-172-30-15-5.ec2.internal: Successfully destroyed cluster
ip-172-30-15-111.ec2.internal: Successfully destroyed cluster
Requesting remove 'pcsd settings' from 'ip-172-30-15-
111.ec2.internal', 'ip-172-30-15-5.ec2.internal'
ip-172-30-15-111.ec2.internal: successful removal of the file 'pcsd
settings'
ip-172-30-15-5.ec2.internal: successful removal of the file 'pcsd
settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'ip-172-30-15-
111.ec2.internal', 'ip-172-30-15-5.ec2.internal'
ip-172-30-15-111.ec2.internal: successful distribution of the file
'corosync authkey'
ip-172-30-15-111.ec2.internal: successful distribution of the file
'pacemaker authkey'
ip-172-30-15-5.ec2.internal: successful distribution of the file
'corosync authkey'
ip-172-30-15-5.ec2.internal: successful distribution of the file
'pacemaker authkey'
Sending 'corosync.conf' to 'ip-172-30-15-111.ec2.internal', 'ip-172-
30-15-5.ec2.internal'
ip-172-30-15-111.ec2.internal: successful distribution of the file
'corosync.conf'
ip-172-30-15-5.ec2.internal: successful distribution of the file
'corosync.conf'
Cluster has been successfully set up.
```

3. Aktivieren Sie den Cluster.

```
sudo pcs cluster enable --all
```

```
[ec2-user@ip-172-30-15-111 ~]$ sudo pcs cluster enable --all
ip-172-30-15-5.ec2.internal: Cluster Enabled
ip-172-30-15-111.ec2.internal: Cluster Enabled
```

4. Starten und validieren Sie den Cluster.

```
sudo pcs cluster start --all
```

```
sudo pcs status
```

```
[ec2-user@ip-172-30-15-111 ~]$ sudo pcs status
Cluster name: ora_ec2nfsx

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Wed Sep 11 15:43:23 2024 on ip-172-30-15-111.ec2.internal
  * Last change: Wed Sep 11 15:43:06 2024 by hacluster via hacluster on ip-172-30-15-111.ec2.internal
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal ]

Full List of Resources:
  * No resources

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Damit ist die Einrichtung und Erstkonfiguration des Pacemaker-Clusters abgeschlossen.

Pacemaker-Cluster-Fencing-Konfiguration

Die Pacemaker-Fencing-Konfiguration ist für einen Produktionscluster obligatorisch. Es stellt sicher, dass ein fehlerhafter Knoten in Ihrem AWS EC2-Cluster automatisch isoliert wird. Dadurch wird verhindert, dass der Knoten die Ressourcen des Clusters verbraucht, die Funktionalität des Clusters beeinträchtigt oder gemeinsam genutzte Daten beschädigt. Dieser Abschnitt demonstriert die Konfiguration der Cluster-Fencing-Funktion mithilfe des Fencing-Agenten `fence_aws`.

1. Geben Sie als Root-Benutzer die folgende AWS-Metadatenabfrage ein, um die Instanz-ID für jeden EC2-Instanzknoten abzurufen.

```
echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id)
```

```
[root@ip-172-30-15-111 ec2-user]# echo $(curl -s  
http://169.254.169.254/latest/meta-data/instance-id)  
i-0d8e7a0028371636f
```

```
or just get instance-id from AWS EC2 console
```

2. Geben Sie den folgenden Befehl ein, um das Fencing-Gerät zu konfigurieren. Verwenden Sie den Befehl `pcmk_host_map`, um den RHEL-Hostnamen der Instanz-ID zuzuordnen. Verwenden Sie den AWS-Zugriffsschlüssel und den geheimen AWS-Zugriffsschlüssel des AWS-Benutzerkontos, das Sie zuvor für die AWS-Authentifizierung verwendet haben.

```
sudo pcs stonith \  
create clusterfence fence_aws access_key=XXXXXXXXXXXXXXXXX  
secret_key=XXXXXXXXXXXXXXXXX \  
region=us-east-1 pcmk_host_map="ip-172-30-15-111.ec2.internal:i-  
0d8e7a0028371636f;ip-172-30-15-5.ec2.internal:i-0bc54b315afb20a2e" \  
power_timeout=240 pcmk_reboot_timeout=480 pcmk_reboot_retries=4
```

3. Validieren Sie die Fencing-Konfiguration.

```
pcs status
```

```
[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Wed Sep 11 21:17:18 2024 on ip-172-30-15-111.ec2.internal
  * Last change: Wed Sep 11 21:16:40 2024 by root via root on ip-172-30-15-111.ec2.internal
  * 2 nodes configured
  * 1 resource instance configured

Node List:
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal ]

Full List of Resources:
  * clusterfence          (stonith:fence_aws):      Started ip-172-30-15-111.ec2.internal

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

4. Setzen Sie „stonith-action“ auf „Aus“, anstatt auf Clusterebene einen Neustart durchzuführen.

```
pcs property set stonith-action=off
```

```
[root@ip-172-30-15-111 ec2-user]# pcs property config
Cluster Properties:
  cluster-infrastructure: corosync
  cluster-name: ora_ec2nfsx
  dc-version: 2.1.7-5.1.el8_10-0f7f88312
  have-watchdog: false
  last-lrm-refresh: 1726257586
  stonith-action: off
```



Wenn „stonith-action“ auf „Aus“ gesetzt ist, wird der eingezäunte Clusterknoten zunächst heruntergefahren. Nach dem in stonith power_timeout definierten Zeitraum (240 Sekunden) wird der eingezäunte Knoten neu gestartet und tritt dem Cluster wieder bei.

5. Stellen Sie die Zaunverzögerung für den Standby-Knoten auf 10 Sekunden ein.

```
pcs stonith update clusterfence pcmk_delay_base="ip-172-30-15-111.ec2.internal:0;ip-172-30-15-5.ec2.internal:10s"
```

```
[root@ip-172-30-15-111 ec2-user]# pcs stonith config
Resource: clusterfence (class=stonith type=fence_aws)
  Attributes: clusterfence-instance_attributes
    access_key=XXXXXXXXXXXXXXXXXX
    pcmk_delay_base=ip-172-30-15-111.ec2.internal:0;ip-172-30-15-5.ec2.internal:10s
    pcmk_host_map=ip-172-30-15-111.ec2.internal:i-0d8e7a0028371636f;ip-172-30-15-5.ec2.internal:i-0bc54b315afb20a2e
    pcmk_reboot_retries=4
    pcmk_reboot_timeout=480
    power_timeout=240
    region=us-east-1
    secret_key=XXXXXXXXXXXXXXXXXX
  Operations:
    monitor: clusterfence-monitor-interval-60s
      interval=60s
```



Ausführen `pcs stonith refresh` Befehl zum Aktualisieren des gestoppten Stonith-Fence-Agenten oder zum Löschen fehlgeschlagener Stonith-Ressourcenaktionen.

Bereitstellen einer Oracle-Datenbank im PCS-Cluster

Wir empfehlen, das von NetApp bereitgestellte Ansible-Playbook zu nutzen, um Datenbankinstallations- und Konfigurationsaufgaben mit vordefinierten Parametern auf dem PCS-Cluster auszuführen. Für diese automatisierte Oracle-Bereitstellung sind vor der Ausführung des Playbooks Benutzereingaben für drei benutzerdefinierte Parameterdateien erforderlich.

- **Hosts** – Definieren Sie Ziele, gegen die das Automatisierungs-Playbook ausgeführt wird.
- **vars/vars.yml** – die globale Variablendatei, die Variablen definiert, die für alle Ziele gelten.
- **host_vars/host_name.yml** – die lokale Variablendatei, die Variablen definiert, die nur für ein benanntes Ziel gelten. In unserem Anwendungsfall sind dies die Oracle DB-Server.

Zusätzlich zu diesen benutzerdefinierten Variablendateien gibt es mehrere Standardvariablendateien, die Standardparameter enthalten, die nur bei Bedarf geändert werden müssen. Im Folgenden werden die Details der automatisierten Oracle-Bereitstellung in AWS EC2 und FSx ONTAP in einer PCS-Clusterkonfiguration gezeigt.

1. Klonen Sie aus dem Stammverzeichnis des Ansible-Controller-Administratorbenutzers eine Kopie des NetApp Oracle Deployment Automation Toolkit für NFS.

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_nfs.git
```



Der Ansible-Controller kann sich im selben VPC wie die EC2-Datenbankinstanz oder vor Ort befinden, solange eine Netzwerkverbindung zwischen ihnen besteht.

2. Füllen Sie die benutzerdefinierten Parameter in den Host-Parameterdateien aus. Nachfolgend finden Sie Beispiele für eine typische Hostdateikonfiguration.

```
[admin@ansiblectl na_oracle_deploy_nfs]$ cat hosts
#Oracle hosts
[oracle]
orapm01 ansible_host=172.30.15.111 ansible_ssh_private_key_file=ec2-user.pem
orapm02 ansible_host=172.30.15.5 ansible_ssh_private_key_file=ec2-user.pem
```

3. Füllen Sie die benutzerdefinierten Parameter in den Parameterdateien vars/vars.yml aus. Nachfolgend finden Sie Beispiele für eine typische Konfiguration der Datei vars.yml.

```

[admin@ansiblectl na_oracle_deploy_nfs]$ cat vars/vars.yml
#####
##
##### Oracle 19c deployment user configuration variables
#####
##### Consolidate all variables from ONTAP, linux and oracle
#####
#####
#####

#####
### ONTAP env specific config variables ###
#####

# Prerequisite to create three volumes in NetApp ONTAP storage from
System Manager or cloud dashboard with following naming convention:
# db_hostname_u01 - Oracle binary
# db_hostname_u02 - Oracle data
# db_hostname_u03 - Oracle redo
# It is important to strictly follow the name convention or the
automation will fail.

#####
### Linux env specific config variables ###
#####

redhat_sub_username: xxxxxxxx
redhat_sub_password: "xxxxxxx"

#####
### DB env specific install and config variables ###
#####

# Database domain name
db_domain: ec2.internal

# Set initial password for all required Oracle passwords. Change
them after installation.
initial_pwd_all: "xxxxxxx"

```

4. Füllen Sie die benutzerdefinierten Parameter in den Parameterdateien `host_vars/host_name.yml` aus. Nachfolgend finden Sie Beispiele für eine typische Konfiguration der Datei `host_vars/host_name.yml`.

```
[admin@ansiblectl na_oracle_deploy_nfs]$ cat host_vars/orapm01.yml
# User configurable Oracle host specific parameters

# Database SID. By default, a container DB is created with 3 PDBs
within the CDB
oracle_sid: NTAP

# CDB is created with SGA at 75% of memory_limit, MB. Consider how
many databases to be hosted on the node and
# how much ram to be allocated to each DB. The grand total of SGA
should not exceed 75% available RAM on node.
memory_limit: 8192

# Local NFS lif ip address to access database volumes
nfs_lif: 172.30.15.95
```



Die nfs_lif-Adresse kann aus der Ausgabe der FSx ONTAP Cluster-Endpunkte aus der automatisierten EC2- und FSx ONTAP Bereitstellung im vorherigen Abschnitt abgerufen werden.

- Erstellen Sie Datenbankvolumes aus der AWS FSx-Konsole. Stellen Sie sicher, dass Sie den Hostnamen des primären PCS-Knotens (orapm01) als Präfix für die Volumes verwenden, wie unten gezeigt.

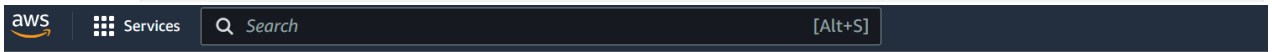
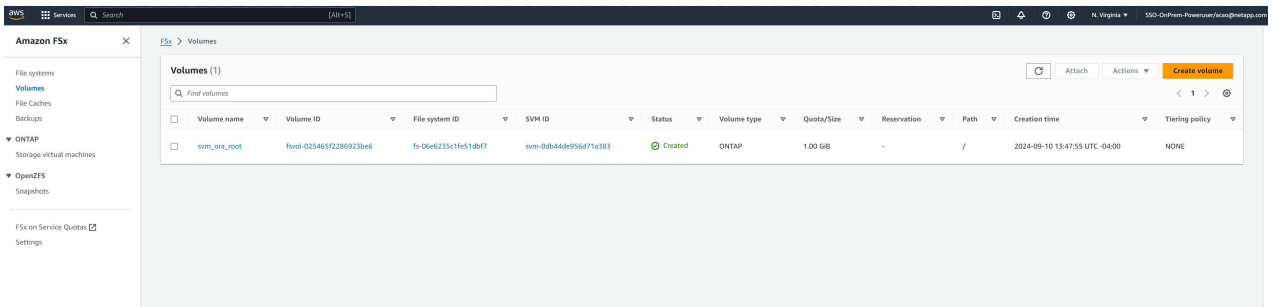
The screenshot shows the AWS Management Console. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces.

The main content area displays the 'Instances (1/2) info' page. A table lists two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Ela
orapm02	i-0bc54b315afb20a2e	Running	t2.xlarge	2/2 checks passed	View alarms +	us-east-1a	-	-	-
orapm01	i-0d8e7a0028371636f	Running	t2.xlarge	2/2 checks passed	View alarms +	us-east-1a	-	-	-

Below the table, the details for instance **i-0bc54b315afb20a2e (orapm02)** are shown. The 'Details' tab is active, displaying various attributes:

- Instance ID copied** (with a copy icon)
- Instance type**: t2.xlarge
- Public IPv4 address**: -
- Instance state**: Running
- Private IP DNS name (IPv4 only)**: ip-172-30-15-5.ec2.internal
- Instance type**: t2.xlarge
- VPC ID**: vpc-0b522d5e982a50ceb (sol-vpcvmmconn-01)
- Private IPv4 addresses**: 172.30.15.5
- Public IPv4 DNS**: -
- Elastic IP addresses**: -
- AWS Compute Optimizer finding**: Opt-in to AWS Compute Optimizer for recommendations. | Learn more



[FSx](#) > [Volumes](#) > Create volume

Create volume

File system type

☒ Amazon FSx for NetApp ONTAP

☐ Amazon FSx for OpenZFS

File system details

File system

The file system where this volume will be created.

ONTAP | fs-06e6235c1fe51dbf7 | fsx_01

Storage virtual machine

The storage virtual machine that will host this volume.

svm-0db44de956d71a383 | svm_ora

Volume details

Volume name

orapm01_u01

Maximum of 203 alphanumeric characters, plus _ .

Volume style

☒ FlexVol (recommended)

FlexVols are the standard ONTAP volume type that can be as large as 300 terabytes.

☐ FlexGroup

FlexGroups are composed of multiple hidden volumes called constituents and can be as large as 20 petabytes.

Volume size

Minimum 20 MiB; Maximum 314,572,800 MiB

50

TiB

Volume type

Select whether you're creating a Read-Write (RW) volume or a read-only Data Protection (DP) volume, which is used with SnapMirror.

☒ Read-Write (RW)

☐ Data Protection (DP)

Junction path

The location within your file system where your volume will be mounted.

/orapm01_u01

Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

☒ Enabled (recommended)

☐ Disabled

Volume security style

The security style of the volume determines whether preference is given to NTFS or UNIX ACLs for multi-protocol access.

Unix (Linux)

Snapshot policy

The snapshot policy of the volume determines the schedule on which snapshots are automatically taken of your volume.

None

Storage tiering

Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

Snapshot Only

Tiering policy cooling period

Your volume's tiering policy cooling period defines the number of days before unaccessed data is marked cold and moved to capacity pool storage. Only affects the Auto and Snapshot-only policies.

31

Default value is 31 days. Valid values are 2-183 days.

Advanced

SnapLock Configuration

Store files using a write-once-read-many (WORM) model to prevent data from being deleted or overwritten for a user-defined period.

☐ Enabled

☒ Disabled

► Tags - optional

Cancel

Create volume

FSx > Volumes

Volumes (4)

Find volumes

☐	Volume name ▾	Volume ID ▾	File system ID ▾	SVM ID ▾	Status ▾	Volume type ▾	Quota/Size ▾	Reservation ▾	Path ▾	Creation time ▾	Tiering policy ▾
☐	orapm01_u03	fsvol-06c48420c929b391b	fs-06e6235c1fe51dbf7	svm-0db44de956d71a383	Created	ONTAP	200.00 TiB	-	/orapm01_u03	2024-09-12 11:21:18 UTC -04:00	SNAPSHOT_ONLY
☐	orapm01_u02	fsvol-0aba81ad57964d955	fs-06e6235c1fe51dbf7	svm-0db44de956d71a383	Created	ONTAP	300.00 TiB	-	/orapm01_u02	2024-09-12 11:20:09 UTC -04:00	SNAPSHOT_ONLY
☐	orapm01_u01	fsvol-0e5ffddcc93a9453	fs-06e6235c1fe51dbf7	svm-0db44de956d71a383	Created	ONTAP	50.00 TiB	-	/orapm01_u01	2024-09-12 11:17:46 UTC -04:00	SNAPSHOT_ONLY
☐	svm_ora_root	fsvol-025465f2286923be6	fs-06e6235c1fe51dbf7	svm-0db44de956d71a383	Created	ONTAP	1.00 GiB	-	/	2024-09-10 13:47:55 UTC -04:00	NONE

6. Führen Sie die folgenden Installationsdateien für Oracle 19c auf dem primären PCS-Knoten EC2-Instanzverzeichnis ip-172-30-15-111.ec2.internal /tmp/archive mit der Berechtigung 777 durch.

```
installer_archives:
  - "LINUX.X64_193000_db_home.zip"
  - "p34765931_190000_Linux-x86-64.zip"
  - "p6880880_190000_Linux-x86-64.zip"
```

7. Führen Sie das Playbook für die Linux-Konfiguration aus für all nodes .

```
ansible-playbook -i hosts 2-linux_config.yml -u ec2-user -e
@vars/vars.yml
```

```
[admin@ansiblectl na_oracle_deploy_nfs]$ ansible-playbook -i hosts
2-linux_config.yml -u ec2-user -e @vars/vars.yml
```

```
PLAY [Linux Setup and Storage Config for Oracle]
```

```
*****
*****
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
*****
*****
*****
```

```
ok: [orapm01]
```

```
ok: [orapm02]
```

```
TASK [linux : Configure RedHat 7 for Oracle DB installation]
```

```
*****
*****
*****
*****
```

```
skipping: [orapm01]
```

```
skipping: [orapm02]
```

```
TASK [linux : Configure RedHat 8 for Oracle DB installation]
```

```
*****
*****
*****
*****
```

```
included:
```

```
/home/admin/na_oracle_deploy_nfs/roles/linux/tasks/rhel8_config.yml
for orapm01, orapm02
```

```
TASK [linux : Register subscriptions for RedHat Server]
```

```
*****
*****
*****
*****
```

```
ok: [orapm01]
```

```
ok: [orapm02]
```

```
.
.
.
```

8. Playbook für Oracle-Konfiguration ausführen only on primary node (Standby-Knoten in der Hosts-Datei auskommentieren).

```
ansible-playbook -i hosts 4-oracle_config.yml -u ec2-user -e
@vars/vars.yml --skip-tags "enable_db_start_shut"
```

```
[admin@ansiblectl na_oracle_deploy_nfs]$ ansible-playbook -i hosts
4-oracle_config.yml -u ec2-user -e @vars/vars.yml --skip-tags
"enable_db_start_shut"
```

```
PLAY [Oracle installation and configuration]
```

```
*****
*****
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
*****
*****
*****
```

```
ok: [orapm01]
```

```
TASK [oracle : Oracle software only install]
```

```
*****
*****
*****
*****
```

```
included:
```

```
/home/admin/na_oracle_deploy_nfs/roles/oracle/tasks/oracle_install.y
ml for orapm01
```

```
TASK [oracle : Create mount points for NFS file systems / Mount NFS
file systems on Oracle hosts]
```

```
*****
*****
*****
*****
```

```
included:
```

```
/home/admin/na_oracle_deploy_nfs/roles/oracle/tasks/oracle_mount_poi
nts.yml for orapm01
```

```
TASK [oracle : Create mount points for NFS file systems]
```

```
*****
```

```

*****
*****
*****
changed: [orapm01] => (item=/u01)
changed: [orapm01] => (item=/u02)
changed: [orapm01] => (item=/u03)
.
.
.

```

9. Nachdem die Datenbank bereitgestellt wurde, kommentieren Sie die Mounts /u01, /u02 und /u03 in /etc/fstab auf dem primären Knoten aus, da die Mount-Punkte nur von PCS verwaltet werden.

```
sudo vi /etc/fstab
```

```

[root@ip-172-30-15-111 ec2-user]# cat /etc/fstab
UUID=eaa1f38e-de0f-4ed5-a5b5-2fa9db43bb38          /          xfs
defaults            0          0
/mnt/swapfile swap swap defaults 0 0
#172.30.15.95:/orapm01_u01 /u01 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0
#172.30.15.95:/orapm01_u02 /u02 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0
#172.30.15.95:/orapm01_u03 /u03 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0

```

10. Kopieren Sie /etc/oratab /etc/orainst.loc, /home/oracle/.bash_profile auf den Standby-Knoten. Stellen Sie sicher, dass die richtigen Dateieigentümer und Berechtigungen vorhanden sind.
11. Fahren Sie die Datenbank und den Listener herunter und führen Sie die Umount-Operationen /u01, /u02 und /u03 auf dem primären Knoten durch.

```

[root@ip-172-30-15-111 ec2-user]# su - oracle
Last login: Wed Sep 18 16:51:02 UTC 2024
[oracle@ip-172-30-15-111 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Sep 18 16:51:16
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> shutdown immediate;

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.18.0.0.0
[oracle@ip-172-30-15-111 ~]$ lsnrctl stop listener.ntap

[oracle@ip-172-30-15-111 ~]$ exit
logout
[root@ip-172-30-15-111 ec2-user]# umount /u01
[root@ip-172-30-15-111 ec2-user]# umount /u02
[root@ip-172-30-15-111 ec2-user]# umount /u03

```

12. Erstellen Sie Einhängepunkte auf dem Standby-Knoten IP-172-30-15-5.

```

mkdir /u01
mkdir /u02
mkdir /u03

```

13. Mounten Sie die FSx ONTAP Datenbankvolumes auf dem Standby-Knoten IP-172-30-15-5.

```

mount -t nfs 172.30.15.95:/orapm01_u01 /u01 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsz=65536,wsz=65536

```

```
mount -t nfs 172.30.15.95:/orapm01_u02 /u02 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsiz=65536
```

```
mount -t nfs 172.30.15.95:/orapm01_u03 /u03 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsiz=65536
```

```
[root@ip-172-30-15-5 ec2-user]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	7.7G	0	7.7G	0%	/dev
tmpfs	7.7G	33M	7.7G	1%	/dev/shm
tmpfs	7.7G	17M	7.7G	1%	/run
tmpfs	7.7G	0	7.7G	0%	/sys/fs/cgroup
/dev/xvda2	50G	21G	30G	41%	/
tmpfs	1.6G	0	1.6G	0%	/run/user/1000
172.30.15.95:/orapm01_u01	48T	47T	844G	99%	/u01
172.30.15.95:/orapm01_u02	285T	285T	844G	100%	/u02
172.30.15.95:/orapm01_u03	190T	190T	844G	100%	/u03

14. Zum Oracle-Benutzer geändert, Binärdatei neu verknüpfen.

```
[root@ip-172-30-15-5 ec2-user]# su - oracle
Last login: Thu Sep 12 18:09:03 UTC 2024 on pts/0
[oracle@ip-172-30-15-5 ~]$ env | grep ORA
ORACLE_SID=NTAP
ORACLE_HOME=/u01/app/oracle/product/19.0.0/NTAP
[oracle@ip-172-30-15-5 ~]$ cd $ORACLE_HOME/bin
[oracle@ip-172-30-15-5 bin]$ ./relink
writing relink log to:
/u01/app/oracle/product/19.0.0/NTAP/install/relinkActions2024-09-
12_06-21-40PM.log
```

15. Kopieren Sie die DNFS-Bibliothek zurück in den ODM-Ordner. Durch erneutes Verknüpfen könnte die DNFS-Bibliotheksdatei verloren gehen.

```
[oracle@ip-172-30-15-5 odm]$ cd
/u01/app/oracle/product/19.0.0/NTAP/rdbms/lib/odm
[oracle@ip-172-30-15-5 odm]$ cp ../../../../lib/libnfsodm19.so .
```

16. Starten Sie die Datenbank zur Validierung auf dem Standby-Knoten IP-172-30-15-5.

```
[oracle@ip-172-30-15-5 odm]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Sep 12 18:30:04
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup;
ORACLE instance started.

Total System Global Area 6442449688 bytes
Fixed Size                  9177880 bytes
Variable Size              1090519040 bytes
Database Buffers           5335154688 bytes
Redo Buffers                7598080 bytes
Database mounted.
Database opened.
SQL> select name, open_mode from v$database;

NAME          OPEN_MODE
-----
NTAP          READ WRITE

SQL> show pdbs

      CON_ID CON_NAME                                OPEN MODE RESTRICTED
-----
      2 PDB$SEED                                READ ONLY NO
      3 NTAP_PDB1                                READ WRITE NO
      4 NTAP_PDB2                                READ WRITE NO
      5 NTAP_PDB3                                READ WRITE NO
```

17. Fahren Sie die Datenbank herunter und führen Sie ein Failback der Datenbank auf den primären Knoten IP-172-30-15-111 durch.

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit

[root@ip-172-30-15-5 ec2-user]# df -h
```

```

Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G         0  7.7G   0% /dev
tmpfs                     7.7G      33M  7.7G   1% /dev/shm
tmpfs                     7.7G      17M  7.7G   1% /run
tmpfs                     7.7G         0  7.7G   0% /sys/fs/cgroup
/dev/xvda2                 50G      21G   30G  41% /
tmpfs                     1.6G         0  1.6G   0% /run/user/1000
172.30.15.95:/orapm01_u01  48T      47T  844G  99% /u01
172.30.15.95:/orapm01_u02 285T    285T  844G 100% /u02
172.30.15.95:/orapm01_u03 190T    190T  844G 100% /u03

[root@ip-172-30-15-5 ec2-user]# umount /u01
[root@ip-172-30-15-5 ec2-user]# umount /u02
[root@ip-172-30-15-5 ec2-user]# umount /u03

[root@ip-172-30-15-111 ec2-user]# mount -t nfs
172.30.15.95:/orapm01_u01 /u01 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536
mount: (hint) your fstab has been modified, but systemd still uses
        the old version; use 'systemctl daemon-reload' to reload.
[root@ip-172-30-15-111 ec2-user]# mount -t nfs
172.30.15.95:/orapm01_u02 /u02 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536
mount: (hint) your fstab has been modified, but systemd still uses
        the old version; use 'systemctl daemon-reload' to reload.
[root@ip-172-30-15-111 ec2-user]# mount -t nfs
172.30.15.95:/orapm01_u03 /u03 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536
mount: (hint) your fstab has been modified, but systemd still uses
        the old version; use 'systemctl daemon-reload' to reload.
[root@ip-172-30-15-111 ec2-user]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G         0  7.7G   0% /dev
tmpfs                     7.8G      48M  7.7G   1% /dev/shm
tmpfs                     7.8G      33M  7.7G   1% /run
tmpfs                     7.8G         0  7.8G   0% /sys/fs/cgroup
/dev/xvda2                 50G      29G   22G  58% /
tmpfs                     1.6G         0  1.6G   0% /run/user/1000
172.30.15.95:/orapm01_u01  48T      47T  844G  99% /u01
172.30.15.95:/orapm01_u02 285T    285T  844G 100% /u02
172.30.15.95:/orapm01_u03 190T    190T  844G 100% /u03
[root@ip-172-30-15-111 ec2-user]# su - oracle
Last login: Thu Sep 12 18:13:34 UTC 2024 on pts/1
[oracle@ip-172-30-15-111 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Sep 12 18:38:46

```

```

2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup;
ORACLE instance started.

Total System Global Area 6442449688 bytes
Fixed Size                  9177880 bytes
Variable Size              1090519040 bytes
Database Buffers           5335154688 bytes
Redo Buffers                7598080 bytes
Database mounted.
Database opened.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.18.0.0.0
[oracle@ip-172-30-15-111 ~]$ lsnrctl start listener.ntap

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 12-SEP-2024
18:39:17

Copyright (c) 1991, 2022, Oracle. All rights reserved.

Starting /u01/app/oracle/product/19.0.0/NTAP/bin/tnslsnr: please
wait...

TNSLSNR for Linux: Version 19.0.0.0.0 - Production
System parameter file is
/u01/app/oracle/product/19.0.0/NTAP/network/admin/listener.ora
Log messages written to /u01/app/oracle/diag/tnslsnr/ip-172-30-15-
111/listener.ntap/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=ip-172-30-
15-111.ec2.internal) (PORT=1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=ip-172-30-
15-111.ec2.internal) (PORT=1521)))
STATUS of the LISTENER
-----
Alias                                listener.ntap

```

```
Version                TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date             12-SEP-2024 18:39:17
Uptime                 0 days 0 hr. 0 min. 0 sec
Trace Level            off
Security               ON: Local OS Authentication
SNMP                   OFF
Listener Parameter File
/u01/app/oracle/product/19.0.0/NTAP/network/admin/listener.ora
Listener Log File      /u01/app/oracle/diag/tnslnr/ip-172-30-15-
111/listener.ntap/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ip-172-30-15-
111.ec2.internal)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
The listener supports no services
The command completed successfully
```

Konfigurieren von Oracle-Ressourcen für die PCS-Verwaltung

Das Ziel der Konfiguration des Pacemaker-Clusterings besteht darin, eine aktive/passive Hochverfügbarkeitslösung für die Ausführung von Oracle in AWS EC2- und FSx ONTAP Umgebungen mit minimalem Benutzereingriff im Fehlerfall einzurichten. Im Folgenden wird die Konfiguration von Oracle-Ressourcen für die PCS-Verwaltung veranschaulicht.

1. Erstellen Sie als Root-Benutzer auf der primären EC2-Instanz IP-172-30-15-111 eine sekundäre private IP-Adresse mit einer nicht verwendeten privaten IP-Adresse im VPC-CIDR-Block als Floating-IP. Erstellen Sie dabei eine Oracle-Ressourcengruppe, zu der die sekundäre private IP-Adresse gehören soll.

```
pcs resource create privip ocf:heartbeat:awsvip  
secondary_private_ip=172.30.15.33 --group oracle
```

```
[root@ip-172-30-15-111 ec2-user]# pcs status  
Cluster name: ora_ec2nfsx  
Cluster Summary:  
  * Stack: corosync (Pacemaker is running)  
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-  
5.1.el8_10-0f7f88312) - partition with quorum  
  * Last updated: Fri Sep 13 16:25:35 2024 on ip-172-30-15-  
111.ec2.internal  
  * Last change:  Fri Sep 13 16:25:23 2024 by root via root on ip-  
172-30-15-111.ec2.internal  
  * 2 nodes configured  
  * 2 resource instances configured  
  
Node List:  
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-  
111.ec2.internal ]  
  
Full List of Resources:  
  * clusterfence          (stonith:fence_aws):      Started ip-172-30-  
15-111.ec2.internal  
  * Resource Group: oracle:  
    * privip              (ocf::heartbeat:awsvip):    Started ip-172-30-  
15-5.ec2.internal  
  
Daemon Status:  
  corosync: active/enabled  
  pacemaker: active/enabled  
  pcsd: active/enabled
```



Wenn das Privip zufällig auf dem Standby-Clusterknoten erstellt wird, verschieben Sie es wie unten gezeigt auf den primären Knoten.

2. Verschieben Sie eine Ressource zwischen Clusterknoten.

```
pcs resource move privip ip-172-30-15-111.ec2.internal
```

```
[root@ip-172-30-15-111 ec2-user]# pcs resource move privip ip-172-30-15-111.ec2.internal
```

```
Warning: A move constraint has been created and the resource 'privip' may or may not move depending on other configuration
```

```
[root@ip-172-30-15-111 ec2-user]# pcs status
```

```
Cluster name: ora_ec2nfsx
```

WARNINGS:

Following resources have been moved and their move constraints are still in place: 'privip'

Run 'pcs constraint location' or 'pcs resource clear <resource id>' to view or remove the constraints, respectively

Cluster Summary:

- * Stack: corosync (Pacemaker is running)
- * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
- * Last updated: Fri Sep 13 16:26:38 2024 on ip-172-30-15-111.ec2.internal
- * Last change: Fri Sep 13 16:26:27 2024 by root via root on ip-172-30-15-111.ec2.internal
- * 2 nodes configured
- * 2 resource instances configured

Node List:

- * Online: [ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal]

Full List of Resources:

- * clusterfence (stonith:fence_aws): Started ip-172-30-15-111.ec2.internal
- * Resource Group: oracle:
 - * privip (ocf::heartbeat:awsvip): Started ip-172-30-15-111.ec2.internal (Monitoring)

Daemon Status:

- corosync: active/enabled
- pacemaker: active/enabled
- pcsd: active/enabled

3. Erstellen Sie eine virtuelle IP (VIP) für Oracle. Die virtuelle IP wird je nach Bedarf zwischen dem primären und dem Standby-Knoten verschoben.

```
pcs resource create vip ocf:heartbeat:IPaddr2 ip=172.30.15.33  
cidr_netmask=25 nic=eth0 op monitor interval=10s --group oracle
```

```
[root@ip-172-30-15-111 ec2-user]# pcs resource create vip
ocf:heartbeat:IPaddr2 ip=172.30.15.33 cidr_netmask=25 nic=eth0 op
monitor interval=10s --group oracle
[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
```

WARNINGS:

Following resources have been moved and their move constraints are still in place: 'privip'

Run 'pcs constraint location' or 'pcs resource clear <resource id>' to view or remove the constraints, respectively

Cluster Summary:

- * Stack: corosync (Pacemaker is running)
- * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
- * Last updated: Fri Sep 13 16:27:34 2024 on ip-172-30-15-111.ec2.internal
- * Last change: Fri Sep 13 16:27:24 2024 by root via root on ip-172-30-15-111.ec2.internal
- * 2 nodes configured
- * 3 resource instances configured

Node List:

- * Online: [ip-172-30-15-5.ec2.internal ip-172-30-15-111.ec2.internal]

Full List of Resources:

- * clusterfence (stonith:fence_aws): Started ip-172-30-15-111.ec2.internal
- * Resource Group: oracle:
 - * privip (ocf::heartbeat:awsvip): Started ip-172-30-15-111.ec2.internal
 - * vip (ocf::heartbeat:IPaddr2): Started ip-172-30-15-111.ec2.internal

Daemon Status:

- corosync: active/enabled
- pacemaker: active/enabled
- pcsd: active/enabled

4. Aktualisieren Sie als Oracle-Benutzer die Dateien listener.ora und tnsnames.ora, sodass sie auf die VIP-Adresse verweisen. Starten Sie den Listener neu. Bounce-Datenbank, falls erforderlich, damit sich die Datenbank beim Listener registrieren kann.

```
vi $ORACLE_HOME/network/admin/listener.ora
```

```
vi $ORACLE_HOME/network/admin/tnsnames.ora
```

```
[oracle@ip-172-30-15-111 admin]$ cat listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/NTAP/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER.NTAP =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 172.30.15.33) (PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
  )
```

```
[oracle@ip-172-30-15-111 admin]$ cat tnsnames.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/NTAP/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.
```

```
NTAP =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = 172.30.15.33) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = NTAP.ec2.internal)
    )
  )
```

```
LISTENER_NTAP =
  (ADDRESS = (PROTOCOL = TCP) (HOST = 172.30.15.33) (PORT = 1521))
```

```
[oracle@ip-172-30-15-111 admin]$ lsnrctl status listener.ntap
```

```
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 13-SEP-2024
18:28:17
```

```
Copyright (c) 1991, 2022, Oracle. All rights reserved.
```

```
Connecting to
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=172.30.15.33) (PORT=1521)))
```

STATUS of the LISTENER

```
Alias                listener.ntap
Version              TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date           13-SEP-2024 18:15:51
Uptime               0 days 0 hr. 12 min. 25 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File
/u01/app/oracle/product/19.0.0/NTAP/network/admin/listener.ora
Listener Log File     /u01/app/oracle/diag/tnslsnr/ip-172-30-15-
111/listener.ntap/alert/log.xml
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=172.30.15.33) (PORT=1521)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=ip-172-30-15-
111.ec2.internal) (PORT=5500)) (Security=(my_wallet_directory=/u01/app
/oracle/product/19.0.0/NTAP/admin/NTAP/xdw_wallet)) (Presentation=HTT
P) (Session=RAW))
```

Services Summary...

Service "21f0b5cc1fa290e2e0636f0f1eacfd43.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "21f0b74445329119e0636f0f1eacec03.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "21f0b83929709164e0636f0f1eacacc3.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "NTAP.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "NTAPXDB.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "ntap_pdb1.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "ntap_pdb2.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

Service "ntap_pdb3.ec2.internal" has 1 instance(s).

Instance "NTAP", status READY, has 1 handler(s) for this service...

The command completed successfully

****Oracle listener now listens on vip for database connection****

5. Fügen Sie der Oracle-Ressourcengruppe die Einhängepunkte /u01, /u02 und /u03 hinzu.

```
pcs resource create u01 ocf:heartbeat:Filesystem
device='172.30.15.95:/orapm01_u01' directory='/u01' fstype='nfs'
options='rw,bg,hard,vers=3,proto=tcp,timeo=600,rsz=65536,wsz=65536' --group oracle
```

```
pcs resource create u02 ocf:heartbeat:Filesystem
device='172.30.15.95:/orapm01_u02' directory='/u02' fstype='nfs'
options='rw,bg,hard,vers=3,proto=tcp,timeo=600,rsz=65536,wsz=65536' --group oracle
```

```
pcs resource create u03 ocf:heartbeat:Filesystem
device='172.30.15.95:/orapm01_u03' directory='/u03' fstype='nfs'
options='rw,bg,hard,vers=3,proto=tcp,timeo=600,rsz=65536,wsz=65536' --group oracle
```

6. Erstellen Sie eine PCS-Monitor-Benutzer-ID in Oracle DB.

```

[root@ip-172-30-15-111 ec2-user]# su - oracle
Last login: Fri Sep 13 18:12:24 UTC 2024 on pts/0
[oracle@ip-172-30-15-111 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 13 19:08:41
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> CREATE USER c##ocfmon IDENTIFIED BY "XXXXXXXXX";

User created.

SQL> grant connect to c##ocfmon;

Grant succeeded.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.18.0.0.0

```

7. Fügen Sie der Oracle-Ressourcengruppe eine Datenbank hinzu.

```

pcs resource create ntap ocf:heartbeat:oracle sid='NTAP'
home='/u01/app/oracle/product/19.0.0/NTAP' user='oracle'
monuser='C##OCFMON' monpassword='XXXXXXXXX' monprofile='DEFAULT'
--group oracle

```

8. Fügen Sie der Oracle-Ressourcengruppe einen Datenbank-Listener hinzu.

```

pcs resource create listener ocf:heartbeat:oralsnr sid='NTAP'
listener='listener.ntap' --group=oracle

```

9. Aktualisieren Sie alle Standortbeschränkungen für Ressourcen in der Oracle-Ressourcengruppe auf den primären Knoten als bevorzugten Knoten.

```
pcs constraint location privip prefers ip-172-30-15-111.ec2.internal
pcs constraint location vip prefers ip-172-30-15-111.ec2.internal
pcs constraint location u01 prefers ip-172-30-15-111.ec2.internal
pcs constraint location u02 prefers ip-172-30-15-111.ec2.internal
pcs constraint location u03 prefers ip-172-30-15-111.ec2.internal
pcs constraint location ntap prefers ip-172-30-15-111.ec2.internal
pcs constraint location listener prefers ip-172-30-15-111.ec2.internal
```

```
[root@ip-172-30-15-111 ec2-user]# pcs constraint config
Location Constraints:
  Resource: listener
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: ntap
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: privip
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: u01
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: u02
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: u03
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
  Resource: vip
    Enabled on:
      Node: ip-172-30-15-111.ec2.internal (score:INFINITY)
Ordering Constraints:
Colocation Constraints:
Ticket Constraints:
```

10. Überprüfen Sie die Konfiguration der Oracle-Ressourcen.

```
pcs status
```

```

[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-
5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Fri Sep 13 19:25:32 2024 on ip-172-30-15-
111.ec2.internal
  * Last change:  Fri Sep 13 19:23:40 2024 by root via root on ip-
172-30-15-111.ec2.internal
  * 2 nodes configured
  * 8 resource instances configured

Node List:
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-
111.ec2.internal ]

Full List of Resources:
  * clusterfence          (stonith:fence_aws):      Started ip-172-30-
15-111.ec2.internal
  * Resource Group: oracle:
    * privip              (ocf::heartbeat:awsvip):   Started ip-172-30-
15-111.ec2.internal
    * vip                  (ocf::heartbeat:IPaddr2):  Started ip-172-30-
15-111.ec2.internal
    * u01                  (ocf::heartbeat:Filesystem): Started ip-172-30-
15-111.ec2.internal
    * u02                  (ocf::heartbeat:Filesystem): Started ip-172-30-
15-111.ec2.internal
    * u03                  (ocf::heartbeat:Filesystem): Started ip-172-30-
15-111.ec2.internal
    * ntap                 (ocf::heartbeat:oracle):   Started ip-172-30-
15-111.ec2.internal
    * listener             (ocf::heartbeat:oralsnr):  Started ip-172-30-
15-111.ec2.internal

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

```

HA-Validierung nach der Bereitstellung

Nach der Bereitstellung ist es wichtig, einige Tests und Validierungen durchzuführen, um sicherzustellen, dass der PCS Oracle-Datenbank-Failovercluster richtig konfiguriert ist und wie erwartet funktioniert. Die Testvalidierung umfasst verwaltetes Failover und simulierte unerwartete Ressourcenausfälle und deren Wiederherstellung durch den Cluster-Schutzmechanismus.

1. Validieren Sie die Knotenumzäunung, indem Sie die Umzäunung des Standby-Knotens manuell auslösen und beobachten, dass der Standby-Knoten nach einem Timeout offline geschaltet und neu gestartet wurde.

```
pcs stonith fence <standbynodename>
```

```

[root@ip-172-30-15-111 ec2-user]# pcs stonith fence ip-172-30-15-5.ec2.internal
Node: ip-172-30-15-5.ec2.internal fenced
[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Fri Sep 13 21:58:45 2024 on ip-172-30-15-111.ec2.internal
  * Last change: Fri Sep 13 21:55:12 2024 by root via root on ip-172-30-15-111.ec2.internal
  * 2 nodes configured
  * 8 resource instances configured

Node List:
  * Online: [ ip-172-30-15-111.ec2.internal ]
  * OFFLINE: [ ip-172-30-15-5.ec2.internal ]

Full List of Resources:
  * clusterfence (stonith:fence_aws): Started ip-172-30-15-111.ec2.internal
  * Resource Group: oracle:
    * privip (ocf::heartbeat:awsvip): Started ip-172-30-15-111.ec2.internal
    * vip (ocf::heartbeat:IPaddr2): Started ip-172-30-15-111.ec2.internal
    * u01 (ocf::heartbeat:Filesystem): Started ip-172-30-15-111.ec2.internal
    * u02 (ocf::heartbeat:Filesystem): Started ip-172-30-15-111.ec2.internal
    * u03 (ocf::heartbeat:Filesystem): Started ip-172-30-15-111.ec2.internal
    * ntap (ocf::heartbeat:oracle): Started ip-172-30-15-111.ec2.internal
    * listener (ocf::heartbeat:oralsnr): Started ip-172-30-15-111.ec2.internal

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

```

2. Simulieren Sie einen Datenbank-Listener-Fehler, indem Sie den Listener-Prozess beenden, und

beobachten Sie, dass PCS den Listener-Fehler überwacht und ihn innerhalb weniger Sekunden neu gestartet hat.

```
[root@ip-172-30-15-111 ec2-user]# ps -ef | grep lsnr
oracle      154895          1  0 18:15 ?                00:00:00
/u01/app/oracle/product/19.0.0/NTAP/bin/tnslsnr listener.ntap
-inherit
root        217779    120186  0 19:36 pts/0        00:00:00 grep
--color=auto lsnr
[root@ip-172-30-15-111 ec2-user]# kill -9 154895

[root@ip-172-30-15-111 ec2-user]# su - oracle
Last login: Thu Sep 19 14:58:54 UTC 2024
[oracle@ip-172-30-15-111 ~]$ lsnrctl status listener.ntap

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 13-SEP-2024
19:36:51

Copyright (c) 1991, 2022, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=172.30.15.33) (PORT=1521)))
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-00511: No listener
Linux Error: 111: Connection refused
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1521)))
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-00511: No listener
Linux Error: 111: Connection refused

[oracle@ip-172-30-15-111 ~]$ lsnrctl status listener.ntap

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 19-SEP-2024
15:00:10

Copyright (c) 1991, 2022, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=172.30.15.33) (PORT=1521)))
STATUS of the LISTENER
-----
Alias                     listener.ntap
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 -
```

```

Production
Start Date                16-SEP-2024 14:00:14
Uptime                    3 days 0 hr. 59 min. 56 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/19.0.0/NTAP/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/ip-172-30-15-
111/listener.ntap/alert/log.xml
Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=172.30.15.33) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=ip-172-30-15-
111.ec2.internal) (PORT=5500)) (Security=(my_wallet_directory=/u01/app
/oracle/product/19.0.0/NTAP/admin/NTAP/xdw_wallet)) (Presentation=HTT
P) (Session=RAW))
Services Summary...
Service "21f0b5cc1fa290e2e0636f0f1eacfd43.ec2.internal" has 1
instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "21f0b74445329119e0636f0f1eacec03.ec2.internal" has 1
instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "21f0b83929709164e0636f0f1eacacc3.ec2.internal" has 1
instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "NTAP.ec2.internal" has 1 instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "NTAPXDB.ec2.internal" has 1 instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "ntap_pdb1.ec2.internal" has 1 instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "ntap_pdb2.ec2.internal" has 1 instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...
Service "ntap_pdb3.ec2.internal" has 1 instance(s).
  Instance "NTAP", status READY, has 1 handler(s) for this
service...

```

The command completed successfully

3. Simulieren Sie einen Datenbankfehler, indem Sie den pmon-Prozess beenden, und beobachten Sie, dass PCS den Datenbankfehler überwacht und ihn innerhalb weniger Sekunden neu gestartet hat.

```
**Make a remote connection to ntap database**

[oracle@ora_01 ~]$ sqlplus
system@//172.30.15.33:1521/NTAP.ec2.internal

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 13 15:42:42
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Thu Sep 12 2024 13:37:28 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-----
HOST_NAME
-----
NTAP
ip-172-30-15-111.ec2.internal

SQL>

**Kill ntap pmon process to simulate a failure**

[root@ip-172-30-15-111 ec2-user]# ps -ef | grep pmon
oracle      159247          1  0 18:27 ?          00:00:00 ora_pmon_NTAP
root        230595    120186  0 19:44 pts/0      00:00:00 grep
--color=auto pmon
[root@ip-172-30-15-111 ec2-user]# kill -9 159247

**Observe the DB failure**
```

```

SQL> /
select instance_name, host_name from v$instance
*
ERROR at line 1:
ORA-03113: end-of-file on communication channel
Process ID: 227424
Session ID: 396 Serial number: 4913

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.18.0.0.0

**Reconnect to DB after reboot**

[oracle@ora_01 ~]$ sqlplus
system@//172.30.15.33:1521/NTAP.ec2.internal

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 13 15:47:24
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Fri Sep 13 2024 15:42:47 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-----
HOST_NAME
-----
NTAP
ip-172-30-15-111.ec2.internal

SQL>

```

4. Validieren Sie ein verwaltetes Datenbank-Failover vom Primär- zum Standby-Knoten, indem Sie den

Primärknoten in den Standby-Modus versetzen, um ein Failover der Oracle-Ressourcen auf den Standby-Knoten durchzuführen.

```
pcs node standby <nodename>
```

****Stopping Oracle resources on primary node in reverse order****

```
[root@ip-172-30-15-111 ec2-user]# pcs node standby ip-172-30-15-111.ec2.internal
```

```
[root@ip-172-30-15-111 ec2-user]# pcs status
```

Cluster name: ora_ec2nfsx

Cluster Summary:

- * Stack: corosync (Pacemaker is running)
- * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
- * Last updated: Fri Sep 13 20:01:16 2024 on ip-172-30-15-111.ec2.internal
- * Last change: Fri Sep 13 20:01:08 2024 by root via root on ip-172-30-15-111.ec2.internal
- * 2 nodes configured
- * 8 resource instances configured

Node List:

- * Node ip-172-30-15-111.ec2.internal: standby (with active resources)
- * Online: [ip-172-30-15-5.ec2.internal]

Full List of Resources:

- * clusterfence (stonith:fence_aws): Started ip-172-30-15-5.ec2.internal
- * Resource Group: oracle:
 - * privip (ocf::heartbeat:awsvip): Started ip-172-30-15-111.ec2.internal
 - * vip (ocf::heartbeat:IPaddr2): Started ip-172-30-15-111.ec2.internal
 - * u01 (ocf::heartbeat:Filesystem): Stopping ip-172-30-15-111.ec2.internal
 - * u02 (ocf::heartbeat:Filesystem): Stopped
 - * u03 (ocf::heartbeat:Filesystem): Stopped
 - * ntap (ocf::heartbeat:oracle): Stopped
 - * listener (ocf::heartbeat:oralsnr): Stopped

Daemon Status:

corosync: active/enabled
pacemaker: active/enabled

```

pcsd: active/enabled

**Starting Oracle resources on standby node in sequential order**

[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Fri Sep 13 20:01:34 2024 on ip-172-30-15-111.ec2.internal
  * Last change:  Fri Sep 13 20:01:08 2024 by root via root on ip-172-30-15-111.ec2.internal
  * 2 nodes configured
  * 8 resource instances configured

Node List:
  * Node ip-172-30-15-111.ec2.internal: standby
  * Online: [ ip-172-30-15-5.ec2.internal ]

Full List of Resources:
  * clusterfence      (stonith:fence_aws):      Started ip-172-30-15-5.ec2.internal
  * Resource Group: oracle:
    * privip          (ocf::heartbeat:awsvip):      Started ip-172-30-15-5.ec2.internal
    * vip              (ocf::heartbeat:IPaddr2):      Started ip-172-30-15-5.ec2.internal
    * u01              (ocf::heartbeat:Filesystem):    Started ip-172-30-15-5.ec2.internal
    * u02              (ocf::heartbeat:Filesystem):    Started ip-172-30-15-5.ec2.internal
    * u03              (ocf::heartbeat:Filesystem):    Started ip-172-30-15-5.ec2.internal
    * ntap             (ocf::heartbeat:oracle):      Starting ip-172-30-15-5.ec2.internal
    * listener         (ocf::heartbeat:oralsnr):      Stopped

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

**NFS mount points mounted on standby node**

```

```
[root@ip-172-30-15-5 ec2-user]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G         0  7.7G   0% /dev
tmpfs                     7.7G      33M   7.7G   1% /dev/shm
tmpfs                     7.7G      17M   7.7G   1% /run
tmpfs                     7.7G         0  7.7G   0% /sys/fs/cgroup
/dev/xvda2                 50G       21G   30G  41% /
tmpfs                     1.6G         0  1.6G   0% /run/user/1000
172.30.15.95:/orapm01_u01  48T       47T   840G  99% /u01
172.30.15.95:/orapm01_u02 285T      285T   840G 100% /u02
172.30.15.95:/orapm01_u03 190T      190T   840G 100% /u03
tmpfs                     1.6G         0  1.6G   0% /run/user/54321
```

****Database opened on standby node****

```
[oracle@ora_01 ~]$ sqlplus
system@//172.30.15.33:1521/NTAP.ec2.internal
```

```
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 13 16:34:08
2024
Version 19.18.0.0.0
```

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:

Last Successful login time: Fri Sep 13 2024 15:47:28 -04:00

Connected to:

```
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0
```

```
SQL> select name, open_mode from v$database;
```

```
NAME          OPEN_MODE
-----
NTAP          READ WRITE
```

```
SQL> select instance_name, host_name from v$instance;
```

```
INSTANCE_NAME
-----
HOST_NAME
-----
NTAP
ip-172-30-15-5.ec2.internal
```

SQL>

5. Validieren Sie ein Failback einer verwalteten Datenbank vom Standby- zum Primärknoten, indem Sie den primären Knoten aus dem Standby-Modus entfernen, und beobachten Sie, dass das Failback der Oracle-Ressourcen aufgrund der bevorzugten Knoteneinstellung automatisch erfolgt.

```
pcs node unstandby <nodename>
```

```
**Stopping Oracle resources on standby node for failback to
primary**
```

```
[root@ip-172-30-15-111 ec2-user]# pcs node unstandby ip-172-30-15-
111.ec2.internal
```

```
[root@ip-172-30-15-111 ec2-user]# pcs status
```

```
Cluster name: ora_ec2nfsx
```

```
Cluster Summary:
```

```
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-
5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Fri Sep 13 20:41:30 2024 on ip-172-30-15-
111.ec2.internal
  * Last change:  Fri Sep 13 20:41:18 2024 by root via root on ip-
172-30-15-111.ec2.internal
  * 2 nodes configured
  * 8 resource instances configured
```

```
Node List:
```

```
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-
111.ec2.internal ]
```

```
Full List of Resources:
```

```
  * clusterfence      (stonith:fence_aws):      Started ip-172-30-
15-5.ec2.internal
  * Resource Group: oracle:
    * privip          (ocf::heartbeat:awsvip):      Stopping ip-172-30-
15-5.ec2.internal
    * vip              (ocf::heartbeat:IPaddr2):      Stopped
    * u01              (ocf::heartbeat:Filesystem):    Stopped
    * u02              (ocf::heartbeat:Filesystem):    Stopped
    * u03              (ocf::heartbeat:Filesystem):    Stopped
    * ntap             (ocf::heartbeat:oracle):      Stopped
    * listener         (ocf::heartbeat:oralsnr):      Stopped
```

```
Daemon Status:
```

```

corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled

**Starting Oracle resources on primary node for failback**

[root@ip-172-30-15-111 ec2-user]# pcs status
Cluster name: ora_ec2nfsx
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: ip-172-30-15-111.ec2.internal (version 2.1.7-
5.1.el8_10-0f7f88312) - partition with quorum
  * Last updated: Fri Sep 13 20:41:45 2024 on ip-172-30-15-
111.ec2.internal
  * Last change:  Fri Sep 13 20:41:18 2024 by root via root on ip-
172-30-15-111.ec2.internal
  * 2 nodes configured
  * 8 resource instances configured

Node List:
  * Online: [ ip-172-30-15-5.ec2.internal ip-172-30-15-
111.ec2.internal ]

Full List of Resources:
  * clusterfence      (stonith:fence_aws):      Started ip-172-30-
15-5.ec2.internal
  * Resource Group: oracle:
    * privip      (ocf::heartbeat:awsvip):      Started ip-172-30-
15-111.ec2.internal
    * vip          (ocf::heartbeat:IPaddr2):      Started ip-172-30-
15-111.ec2.internal
    * u01          (ocf::heartbeat:Filesystem):    Started ip-172-30-
15-111.ec2.internal
    * u02          (ocf::heartbeat:Filesystem):    Started ip-172-30-
15-111.ec2.internal
    * u03          (ocf::heartbeat:Filesystem):    Started ip-172-30-
15-111.ec2.internal
    * ntap         (ocf::heartbeat:oracle):        Starting ip-172-30-
15-111.ec2.internal
    * listener     (ocf::heartbeat:oralsnr):        Stopped

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

```

```

**Database now accepts connection on primary node**

[oracle@ora_01 ~]$ sqlplus
system@//172.30.15.33:1521/NTAP.ec2.internal

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 13 16:46:07
2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Fri Sep 13 2024 16:34:12 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-----
HOST_NAME
-----
NTAP
ip-172-30-15-111.ec2.internal

SQL>

```

Damit ist die Oracle HA-Validierung und Lösungsdemonstration in AWS EC2 mit Pacemaker-Clustering und Amazon FSx ONTAP als Datenbankspeicher-Backend abgeschlossen.

Oracle-Backup, -Wiederherstellung und -Klonen mit SnapCenter

NetApp empfiehlt das SnapCenter UI-Tool zur Verwaltung von Oracle-Datenbanken, die in AWS EC2 und Amazon FSx ONTAP bereitgestellt werden. Siehe TR-4979 "[Vereinfachtes, selbstverwaltetes Oracle in VMware Cloud auf AWS mit gastmontiertem FSx ONTAP](#)" Abschnitt Oracle backup, restore, and clone with SnapCenter Weitere Informationen zum Einrichten von SnapCenter und zum Ausführen der Workflows zum Sichern, Wiederherstellen und Klonen von Datenbanken.

Wo Sie weitere Informationen finden

Weitere Informationen zu den in diesem Dokument beschriebenen Informationen finden Sie in den folgenden Dokumenten und/oder auf den folgenden Websites:

- ["Konfigurieren und Verwalten von Hochverfügbarkeitsclustern"](#)
- ["Amazon FSx ONTAP"](#)
- ["Bereitstellen von Oracle Direct NFS"](#)

Copyright-Informationen

Copyright © 2025 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.