



TR-4956: AWS FSx/EC2 中的自动化 PostgreSQL 高可用性部署和灾难恢复

NetApp database solutions

NetApp
August 18, 2025

目录

TR-4956: AWS FSx/EC2 中的自动化 PostgreSQL 高可用性部署和灾难恢复	1
目的	1
受众	1
解决方案测试和验证环境	1
架构	2
硬件和软件组件	2
部署考虑的关键因素	3
解决方案部署	3
自动部署的先决条件	3
配置 hosts 文件	4
配置host_vars文件夹中的host_name.yml文件	5
配置vars文件夹中的全局fsx_vars.yml文件	6
PostgreSQL 部署和 HA/DR 设置	8
PostgreSQL 数据库快照备份并复制到备用站点	9
故障转移到灾难恢复备用站点	10
故障转移测试后重新同步复制的数据库卷	10
由于 EC2 计算实例故障，从主 EC2 数据库服务器故障转移到备用 EC2 数据库服务器	10
在哪里可以找到更多信息	10

TR-4956: AWS FSx/EC2 中的自动化 PostgreSQL 高可用性部署和灾难恢复

Allen Cao、Niyaz Mohamed, NetApp

该解决方案提供了基于 FSx ONTAP 存储产品内置的 NetApp SnapMirror 技术和 AWS 中的 NetApp Ansible 自动化工具包的 PostgreSQL 数据库部署和 HA/DR 设置、故障转移、重新同步的概述和详细信息。

目的

PostgreSQL 是一个广泛使用的开源数据库，在十大最受欢迎数据库引擎中排名第四。["数据库引擎"](#)。一方面，PostgreSQL 因其免许可、开源模式而广受欢迎，同时仍具有复杂的功能。另一方面，由于它是开源的，因此缺乏关于高可用性和灾难恢复（HA/DR）领域的生产级数据库部署的详细指导，尤其是在公共云中。一般来说，建立一个具有热备用、温备用、流复制等功能的典型 PostgreSQL HA/DR 系统可能很困难。通过提升备用站点然后切换回主站点来测试 HA/DR 环境可能会对生产造成干扰。当读取工作负载部署在流式热备用系统上时，主数据库上存在有充分记录的性能问题。

在本文档中，我们演示了如何摆脱应用程序级 PostgreSQL 流式 HA/DR 解决方案，并使用存储级复制构建基于 AWS FSx ONTAP 存储和 EC2 计算实例的 PostgreSQL HA/DR 解决方案。与传统的 PostgreSQL 应用程序级 HA/DR 流复制相比，该解决方案创建了一个更简单、更可比系统，并提供了相同的结果。

该解决方案基于经过验证的成熟 NetApp SnapMirror 存储级复制技术构建，该技术可用于 AWS 原生 FSX ONTAP 云存储，实现 PostgreSQL HA/DR。使用 NetApp 解决方案团队提供的自动化工具包可以轻松实现。它提供类似的功能，同时通过基于应用程序级流的 HA/DR 解决方案消除主站点的复杂性和性能拖累。该解决方案可以轻松部署和测试，而不会影响活动主站点。

此解决方案适用于以下用例：

- 在公共 AWS 云中为 PostgreSQL 提供生产级 HA/DR 部署
- 在公共 AWS 云中测试和验证 PostgreSQL 工作负载
- 测试和验证基于 NetApp SnapMirror 复制技术的 PostgreSQL HA/DR 策略

受众

此解决方案适用于以下人群：

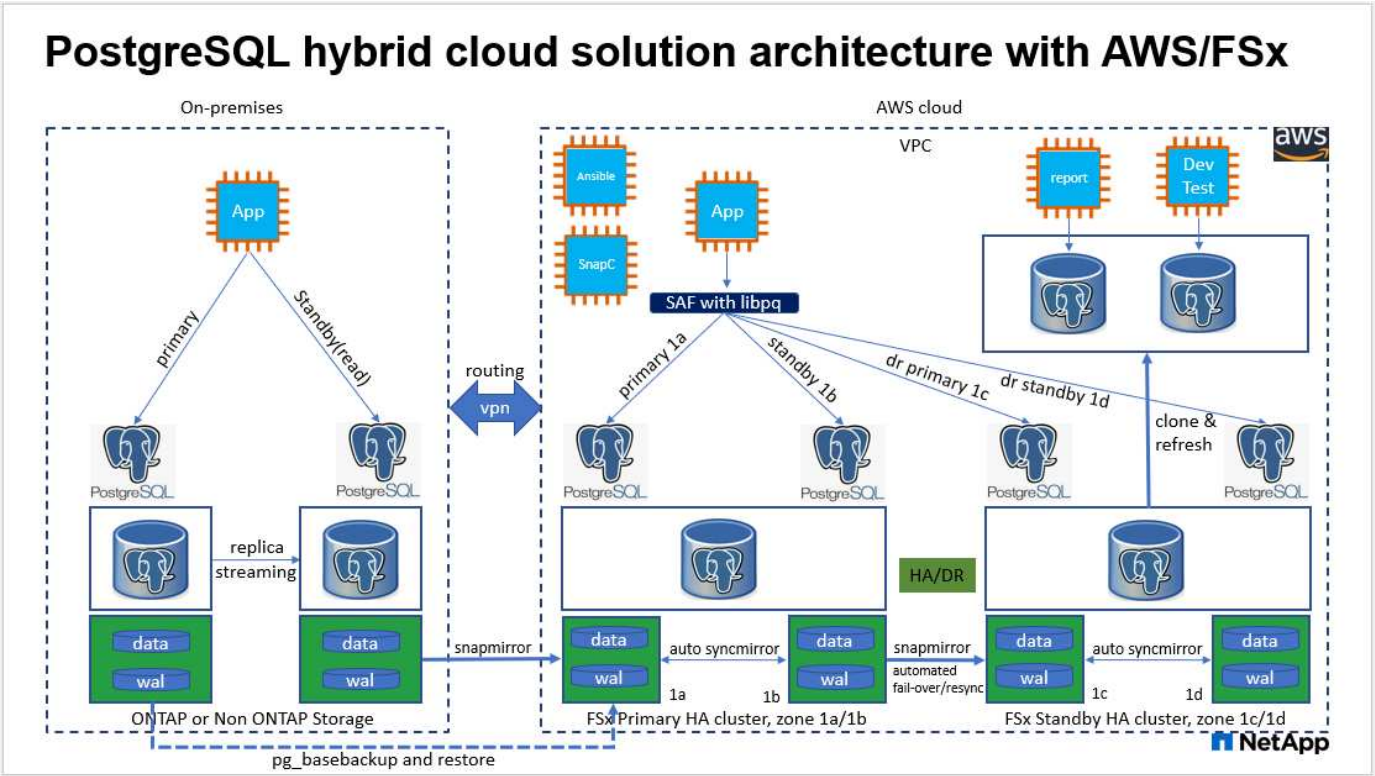
- 有兴趣在公共 AWS 云中部署具有 HA/DR 的 PostgreSQL 的 DBA。
- 有兴趣在公共 AWS 云中测试 PostgreSQL 工作负载的数据库解决方案架构师。
- 有兴趣部署和管理部署到 AWS FSx 存储的 PostgreSQL 实例的存储管理员。
- 有兴趣在 AWS FSx/EC2 中建立 PostgreSQL 环境的应用程序所有者。

解决方案测试和验证环境

该解决方案的测试和验证是在可能与最终部署环境不匹配的 AWS FSx 和 EC2 环境中进行的。有关更多信息，

请参阅[部署考虑的关键因素]。

架构



硬件和软件组件

硬件		
FSx ONTAP存储	当前版本	同一 VPC 和可用区中的两个 FSx HA 对作为主 HA 集群和备用 HA 集群
用于计算的 EC2 实例	t2.xlarge/4vCPU/16G	两个 EC2 T2 xlarge 作为主计算实例和备用计算实例
Ansible 控制器	本地 Centos VM/4vCPU/8G	用于在本地或云端托管 Ansible 自动化控制器的虚拟机
软件		
红帽Linux	RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2	部署 RedHat 订阅进行测试
Centos Linux	CentOS Linux 版本 8.2.2004（核心）	在本地实验室中部署托管 Ansible 控制器
PostgreSQL	14.5 版	自动化从 postgresql.org yum repo 中提取最新可用的 PostgreSQL 版本
Ansible	版本 2.10.3	使用需求手册安装所需集合和库的先决条件

部署考虑的关键因素

- **PostgreSQL 数据库备份、还原和恢复。** PostgreSQL 数据库支持多种备份方法，例如使用 `pg_dump` 进行逻辑备份、使用 `pg_basebackup` 或较低级别的 OS 备份命令进行物理在线备份以及存储级别一致的快照。该解决方案使用 NetApp 一致性组快照在备用站点对 PostgreSQL 数据库数据和 WAL 卷进行备份、还原和恢复。NetApp 一致性组卷快照在将 I/O 写入存储时对其进行排序，并保护数据库数据文件的完整性。
- **EC2 计算实例。** 在这些测试和验证中，我们使用 AWS EC2 `t2.xlarge` 实例类型作为 PostgreSQL 数据库计算实例。NetApp 建议在部署中使用 M5 类型的 EC2 实例作为 PostgreSQL 的计算实例，因为它针对数据库工作负载进行了优化。备用计算实例应始终与为 FSx HA 集群部署的被动（备用）文件系统部署在同一区域中。
- ***FSx 存储 HA 集群单区域或多区域部署。*** 在这些测试和验证中，我们在单个 AWS 可用区中部署了一个 FSx HA 集群。对于生产部署，NetApp 建议在两个不同的可用区部署 FSx HA 对。如果主服务器和备用服务器之间需要特定的距离，则可以在不同区域设置灾难恢复备用 HA 对，以实现业务连续性。FSx HA 集群始终在 HA 对中配置，该 HA 对在主动-被动文件系统中同步镜像，以提供存储级冗余。
- **PostgreSQL 数据和日志放置。** 典型的 PostgreSQL 部署共享相同的根目录或卷用于数据和日志文件。在我们的测试和验证中，我们将 PostgreSQL 数据和日志分成两个单独的卷以提高性能。数据目录中使用软链接指向托管 PostgreSQL WAL 日志和存档 WAL 日志的日志目录或卷。
- ***PostgreSQL 服务启动延迟计时器。*** 该解决方案使用 NFS 挂载卷来存储 PostgreSQL 数据库文件和 WAL 日志文件。在数据库主机重新启动期间，PostgreSQL 服务可能会在未安装卷的情况下尝试启动。这会导致数据库服务启动失败。PostgreSQL 数据库需要 10 到 15 秒的计时器延迟才能正确启动。
- **RPO/RTO 确保业务连续性。** FSx 数据从主服务器到备用服务器的 DR 复制基于 ASYNC，这意味着 RPO 取决于 Snapshot 备份和 SnapMirror 复制的频率。更高的 Snapshot 副本和 SnapMirror 复制频率可降低 RPO。因此，灾难发生时的潜在数据丢失和增量存储成本之间存在平衡。我们已经确定，Snapshot 副本和 SnapMirror 复制可以在低至 5 分钟的间隔内实现 RPO，并且 PostgreSQL 通常可以在一分钟内于 DR 备用站点恢复 RTO。
- ***数据库备份。*** 在将 PostgreSQL 数据库从本地数据中心实施或迁移到 AWS FSx 存储后，数据会在 FSx HA 对中自动同步镜像以进行保护。一旦发生灾难，数据将通过复制的备用站点得到进一步保护。对于长期备份保留或数据保护，NetApp 建议使用内置的 PostgreSQL `pg_basebackup` 实用程序运行可移植到 S3 blob 存储的完整数据库备份。

解决方案部署

按照下面概述的详细说明，可以使用基于 NetApp Ansible 的自动化工具包自动完成此解决方案的部署。

1. 阅读自动化工具包 `README.md` 中的说明 `"na_postgresql_aws_deploy_hadr"`。
2. 观看以下视频演示。

自动化 PostgreSQL 部署和保护

1. 配置所需的参数文件 (`hosts`, `host_vars/host_name.yml`, `fsx_vars.yml`) 通过在相关部分的模板中输入用户特定的参数。然后使用复制按钮将文件复制到 Ansible 控制器主机。

自动部署的先决条件

部署需要以下先决条件。

1. 已设置 AWS 账户，并在您的 AWS 账户内创建了必要的 VPC 和网络段。
2. 从 AWS EC2 控制台，您必须部署两个 EC2 Linux 实例，一个作为主站点上的主 PostgreSQL DB 服务器，

另一个作为备用 DR 站点上的服务器。为了在主 DR 站点和备用 DR 站点实现计算冗余，请部署两个额外的 EC2 Linux 实例作为备用 PostgreSQL DB 服务器。有关环境设置的更多详细信息，请参阅上一节中的架构图。还请查看["Linux 实例用户指南"](#)了解更多信息。

3. 从 AWS EC2 控制台部署两个 FSx ONTAP 存储 HA 集群来托管 PostgreSQL 数据库卷。如果您不熟悉 FSx 存储的部署，请参阅文档["创建 FSx ONTAP 文件系统"](#)以获得分步说明。
4. 构建一个 Centos Linux VM 来托管 Ansible 控制器。Ansible 控制器可以位于本地或 AWS 云中。如果它位于本地，则必须具有与 VPC、EC2 Linux 实例和 FSx 存储集群的 SSH 连接。
5. 按照资源中的“在 RHEL/CentOS 上为 CLI 部署设置 Ansible 控制节点”部分所述设置 Ansible 控制器["NetApp 解决方案自动化入门"](#)。
6. 从公共 NetApp GitHub 站点克隆自动化工具包的副本。

```
git clone https://github.com/NetApp-
Automation/na_postgresql_aws_deploy_hadr.git
```

1. 从工具包根目录执行先决条件剧本来安装 Ansible 控制器所需的集合和库。

```
ansible-playbook -i hosts requirements.yml
```

```
ansible-galaxy collection install -r collections/requirements.yml --force
--force-with-deps
```

1. 检索数据库主机变量文件所需的 EC2 FSx 实例参数 `host\_vars/\*` 和全局变量文件 `fsx\_vars.yml` 配置。

配置 hosts 文件

将主 FSx ONTAP 集群管理 IP 和 EC2 实例主机名输入到 hosts 文件中。

```
# Primary FSx cluster management IP address
[fsx_ontap]
172.30.15.33
```

```
# Primary PostgreSQL DB server at primary site where database is
initialized at deployment time
[postgresql]
psql_01p ansible_ssh_private_key_file=psql_01p.pem
```

```
# Primary PostgreSQL DB server at standby site where postgresql service is
installed but disabled at deployment
# Standby DB server at primary site, to setup this server comment out
other servers in [dr_postgresql]
# Standby DB server at standby site, to setup this server comment out
other servers in [dr_postgresql]
[dr_postgresql] --
psql_01s ansible_ssh_private_key_file=psql_01s.pem
#psql_01ps ansible_ssh_private_key_file=psql_01ps.pem
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

配置host_vars文件夹中的host_name.yml文件

```
# Add your AWS EC2 instance IP address for the respective PostgreSQL
server host
ansible_host: "10.61.180.15"

# "{{groups.postgresql[0]}}" represents first PostgreSQL DB server as
defined in PostgreSQL hosts group [postgresql]. For concurrent multiple
PostgreSQL DB servers deployment, [0] will be incremented for each
additional DB server. For example, "{{groups.postgresql[1]}}" represents
DB server 2, "{{groups.postgresql[2]}}" represents DB server 3 ... As a
good practice and the default, two volumes are allocated to a PostgreSQL
DB server with corresponding /pgdata, /pglogs mount points, which store
PostgreSQL data, and PostgreSQL log files respectively. The number and
naming of DB volumes allocated to a DB server must match with what is
defined in global fsx_vars.yml file by src_db_vols, src_archivelog_vols
parameters, which dictates how many volumes are to be created for each DB
server. aggr_name is aggr1 by default. Do not change. lif address is the
NFS IP address for the SVM where PostgreSQL server is expected to mount
its database volumes. Primary site servers from primary SVM and standby
servers from standby SVM.
host_datastores_nfs:
  - {vol_name: "{{groups.postgresql[0]}}_pgdata", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}
  - {vol_name: "{{groups.postgresql[0]}}_pglogs", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

# Add swap space to EC2 instance, that is equal to size of RAM up to 16G
max. Determine the number of blocks by dividing swap size in MB by 128.
swap_blocks: "128"

# Postgresql user configurable parameters
psql_port: "5432"
buffer_cache: "8192MB"
archive_mode: "on"
max_wal_size: "5GB"
client_address: "172.30.15.0/24"
```

配置vars文件夹中的全局fsx_vars.yml文件

```
#####
##### PostgreSQL HADR global user configuration variables #####
##### Consolidate all variables from FSx, Linux, and postgresql #####
#####

#####
```

```

### Ontap env specific config variables ###
#####

#####
#####
# Variables for SnapMirror Peering
#####
#####

#Passphrase for cluster peering authentication
passphrase: "xxxxxxx"

#Please enter destination or standby FSx cluster name
dst_cluster_name: "FsxId0cf8e0bccb14805e8"

#Please enter destination or standby FSx cluster management IP
dst_cluster_ip: "172.30.15.90"

#Please enter destination or standby FSx cluster inter-cluster IP
dst_inter_ip: "172.30.15.13"

#Please enter destination or standby SVM name to create mirror
relationship
dst_vserver: "dr"

#Please enter destination or standby SVM management IP
dst_vserver_mgmt_lif: "172.30.15.88"

#Please enter destination or standby SVM NFS lif
dst_nfs_lif: "172.30.15.88"

#Please enter source or primary FSx cluster name
src_cluster_name: "FsxId0cf8e0bccb14805e8"

#Please enter source or primary FSx cluster management IP
src_cluster_ip: "172.30.15.20"

#Please enter source or primary FSx cluster inter-cluster IP
src_inter_ip: "172.30.15.5"

#Please enter source or primary SVM name to create mirror relationship
src_vserver: "prod"

#Please enter source or primary SVM management IP
src_vserver_mgmt_lif: "172.30.15.115"

#####

```

```
#####
# Variable for PostgreSQL Volumes, lif - source or primary FSx NFS lif
address
#####
#####

src_db_vols:
  - {vol_name: "{{groups.postgresql[0]}}_pgdata", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

src_archivelog_vols:
  - {vol_name: "{{groups.postgresql[0]}}_pglogs", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

#Names of the Nodes in the ONTAP Cluster
nfs_export_policy: "default"

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

#NFS Mount points for PostgreSQL DB volumes
mount_points:
  - "/pgdata"
  - "/pglogs"

#RedHat subscription username and password
redhat_sub_username: "xxxxx"
redhat_sub_password: "xxxxx"

#####
### DB env specific install and config variables ###
#####
#The latest version of PostgreSQL RPM is pulled/installed and config file
is deployed from a preconfigured template
#Recovery type and point: default as all logs and promote and leave all
PITR parameters blank
```

PostgreSQL 部署和 HA/DR 设置

以下任务在主 EC2 DB 服务器主机上部署 PostgreSQL DB 服务器服务并在主站点初始化数据库。然后在备用站点设置备用主 EC2 DB 服务器主机。最后，从主站点 FSx 集群到备用站点 FSx 集群建立 DB 卷复制，以实现灾难恢复。

1. 在主 FSx 集群上创建 DB 卷，并在主 EC2 实例主机上设置 postgresql。

```
ansible-playbook -i hosts postgresql_deploy.yml -u ec2-user --private-key psql_01p.pem -e @vars/fsx_vars.yml
```

2. 设置备用 DR EC2 实例主机。

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

3. 设置 FSx ONTAP 集群对等和数据库卷复制。

```
ansible-playbook -i hosts fsx_replication_setup.yml -e @vars/fsx_vars.yml
```

4. 将前面的步骤合并为单步 PostgreSQL 部署和 HA/DR 设置。

```
ansible-playbook -i hosts postgresql_hadr_setup.yml -u ec2-user -e @vars/fsx_vars.yml
```

5. 要在主站点或备用站点设置备用 PostgreSQL DB 主机，请注释掉主机文件 [dr_postgresql] 部分中的所有其他服务器，然后使用相应的目标主机（例如主站点上的 psql_01ps 或备用 EC2 计算实例）执行 postgresql_standby_setup.yml 剧本。确保主机参数文件如 `psql\_01ps.yml` 配置在 `host\_vars` 目录。

```
[dr_postgresql] --
#psql_01s ansible_ssh_private_key_file=psql_01s.pem
psql_01ps ansible_ssh_private_key_file=psql_01ps.pem
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01ps.pem -e @vars/fsx_vars.yml
```

PostgreSQL 数据库快照备份并复制到备用站点

PostgreSQL 数据库快照备份和复制到备用站点可以在 Ansible 控制器上以用户定义的间隔进行控制和执行。我们已经验证间隔可以低至 5 分钟。因此，当主站点发生故障时，如果故障发生在下一次计划快照备份之前，则可能会有 5 分钟的数据丢失。

```
*/15 * * * * /home/admin/na_postgresql_aws_deploy_hadr/data_log_snap.sh
```

故障转移到灾难恢复备用站点

为了将 PostgreSQL HA/DR 系统作为 DR 练习进行测试，请通过执行以下脚本在备用站点上的主备用 EC2 DB 实例上执行故障转移和 PostgreSQL 数据库恢复。在实际的 DR 场景中，对实际故障转移到 DR 站点执行相同的操作。

```
ansible-playbook -i hosts postgresql_failover.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

故障转移测试后重新同步复制的数据库卷

故障转移测试后运行重新同步以重新建立数据库卷 SnapMirror 复制。

```
ansible-playbook -i hosts postgresql_standby_resync.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

由于 **EC2** 计算实例故障，从主 **EC2** 数据库服务器故障转移到备用 **EC2** 数据库服务器

NetApp 建议运行手动故障转移或使用可能需要许可证的成熟 OS 集群软件。

在哪里可以找到更多信息

要了解有关本文档中描述的信息的更多信息，请查看以下文档和/或网站：

- Amazon FSx ONTAP

["https://aws.amazon.com/fsx/netapp-ontap/"](https://aws.amazon.com/fsx/netapp-ontap/)

- Amazon EC2

https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6l71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2

- NetApp 解决方案自动化

["简介"](#)

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。