



## 오픈 소스 데이터베이스 NetApp Solutions

NetApp  
April 20, 2024

This PDF was generated from [https://docs.netapp.com/ko-kr/netapp-solutions/databases/aws\\_postgres\\_fsx\\_ec2\\_hadr.html](https://docs.netapp.com/ko-kr/netapp-solutions/databases/aws_postgres_fsx_ec2_hadr.html) on April 20, 2024. Always check docs.netapp.com for the latest.

# 목차

- 오픈 소스 데이터베이스 ..... 1
  - TR-4956: AWS FSx/EC2에서 자동화된 PostgreSQL 고가용성 구축 및 재해 복구..... 1
  - TR-4722: NetApp ONTAP 모범 사례 기반 MySQL 데이터베이스 ..... 10

# 오픈 소스 데이터베이스

## TR-4956: AWS FSx/EC2에서 자동화된 PostgreSQL 고가용성 구축 및 재해 복구

Allen Cao, Niyaz Mohamed, NetApp

### 목적

PostgreSQL은 에서 가장 많이 사용되는 데이터베이스 엔진 10개 중 4위에 해당하는 널리 사용되는 오픈 소스 데이터베이스입니다 "DB-엔진". 반면 PostgreSQL은 라이선스 없는 오픈 소스 모델에서 고급 기능을 계속 사용하면서 인기를 얻고 있습니다. 반면, 오픈 소스가 되어 있기 때문에 고가용성 및 재해 복구(HA/DR) 영역, 특히 퍼블릭 클라우드에서 운영 수준의 데이터베이스 구축에 대한 자세한 지침은 부족한 실정입니다. 일반적으로 핫 스탠바이 및 웜 스탠바이, 스트리밍 복제 등을 사용하여 일반적인 PostgreSQL HA/DR 시스템을 설정하기 어려울 수 있습니다. 대기 사이트를 프로모션한 다음 운영 사이트로 다시 전환하여 HA/DR 환경을 테스트하는 경우 운영 중단이 발생할 수 있습니다. 읽기 워크로드가 스트리밍 핫 스탠바이에 배포되면 운영 환경에 잘 문서화된 성능 문제가 있습니다.

이 문서에서는 애플리케이션 수준의 PostgreSQL 스트리밍 HA/DR 솔루션을 사용하지 않고 스토리지 레벨 복제를 사용하여 AWS FSx ONTAP 스토리지 및 EC2 컴퓨팅 인스턴스를 기반으로 PostgreSQL HA/DR 솔루션을 구축하는 방법을 설명합니다. 이 솔루션은 기존의 PostgreSQL 애플리케이션 레벨 스트리밍 복제(HA/DR)와 비교할 수 있는 보다 간편하고 유사한 시스템을 생성하고 그에 상응하는 결과를 제공합니다.

이 솔루션은 PostgreSQL HA/DR을 위한 AWS 네이티브 FSX ONTAP 클라우드 스토리지에서 사용할 수 있는 검증된 고급 NetApp SnapMirror 스토리지 레벨 복제 기술을 기반으로 구축되었습니다. NetApp 솔루션 팀이 제공하는 자동화 툴킷을 사용하면 간단하게 구축할 수 있습니다. 또한 유사한 기능을 제공하는 동시에 애플리케이션 레벨의 스트리밍 기반 HA/DR 솔루션을 통해 운영 사이트의 복잡성과 성능 저하를 방지합니다. 액티브 운영 사이트에 영향을 주지 않고 솔루션을 쉽게 구축 및 테스트할 수 있습니다.

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- 퍼블릭 AWS 클라우드에서 PostgreSQL을 위한 운영 등급 HA/DR 구축
- 퍼블릭 AWS 클라우드에서 PostgreSQL 워크로드 테스트 및 검증
- NetApp SnapMirror 복제 기술을 기반으로 PostgreSQL HA/DR 전략 테스트 및 검증

### 대상

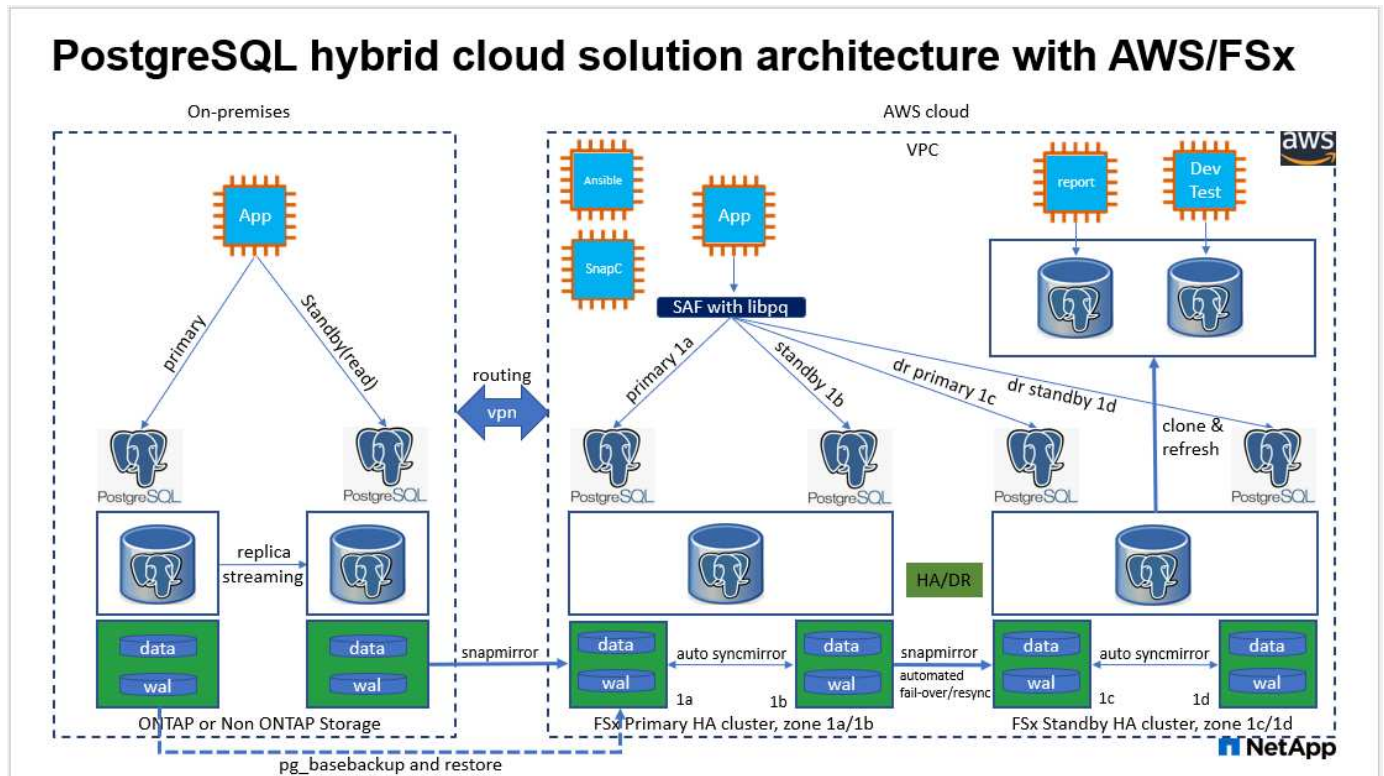
이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

- 퍼블릭 AWS 클라우드에 HA/DR로 PostgreSQL을 구축하려는 DBA
- 퍼블릭 AWS 클라우드에서 PostgreSQL 워크로드를 테스트하는 데 관심이 있는 데이터베이스 솔루션 설계자
- AWS FSx 스토리지에 배포된 PostgreSQL 인스턴스를 배포 및 관리하는 데 관심이 있는 스토리지 관리자
- AWS FSx/EC2에서 PostgreSQL 환경을 구축하는 데 관심이 있는 애플리케이션 소유자입니다.

### 솔루션 테스트 및 검증 환경

이 솔루션의 테스트 및 검증은 최종 구축 환경과 일치하지 않을 수 있는 AWS FSx 및 EC2 환경에서 수행되었습니다. 자세한 내용은 섹션을 참조하십시오 [\[Key Factors for Deployment Consideration\]](#).

있습니다



### 하드웨어 및 소프트웨어 구성 요소

* 하드웨어 *		
FSX ONTAP 저장소	현재 버전	동일한 VPC에 2개의 FSx HA 쌍, 운영 및 대기 HA 클러스터와 가용성 존
컴퓨팅용 EC2 인스턴스	T2.xLarge/4vCPU/16G	2개의 EC2 T2 xLarge를 운영 및 대기 컴퓨팅 인스턴스로 설정합니다
Ansible 컨트롤러	온프레미스 CentOS VM/4vCPU/8G	Ansible 자동화 컨트롤러를 온프레미스 또는 클라우드에서 호스팅하기 위한 VM
* 소프트웨어 *		
RedHat Linux	RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2	테스트를 위해 RedHat 서브스크립션을 배포했습니다
CentOS Linux	CentOS Linux 릴리스 8.2.2004(코어)	온프레미스 랩에 구축된 Ansible 컨트롤러를 호스팅
PostgreSQL	버전 14.5	Automation은 PostgreSQL.ora yum repo에서 사용 가능한 최신 버전의 PostgreSQL을 가져옵니다
Ansible	버전 2.10.3	요구 사항 플레이북과 함께 설치된 필수 컬렉션 및 라이브러리에 대한 사전 요구 사항

## 구축 시 고려해야 할 주요 요소

- \* PostgreSQL 데이터베이스 백업, 복원 및 복구. \* PostgreSQL 데이터베이스는 pg\_dump를 사용한 논리적 백업, pg\_basebackup 또는 하위 수준의 OS 백업 명령을 사용한 물리적 온라인 백업, 스토리지 레벨 정합성 보장 스냅샷과 같은 다양한 백업 방법을 지원합니다. 이 솔루션은 PostgreSQL 데이터베이스 데이터에 NetApp 정합성 보장 그룹 스냅샷을 사용하고 대기 사이트에서 Wal 볼륨 백업, 복원 및 복구를 수행합니다. NetApp 정합성 보장 그룹 볼륨 스냅샷은 스토리지에 쓰일 때 입출력을 순차적으로 생성하고 데이터베이스 데이터 파일의 무결성을 보호합니다.
- \* EC2 컴퓨팅 인스턴스 \* 이러한 테스트 및 검증에서는 PostgreSQL 데이터베이스 컴퓨팅 인스턴스에 대해 AWS EC2 T2.xLarge 인스턴스 유형을 사용했습니다. 데이터베이스 워크로드에 최적화되어 있으므로 배치 시 PostgreSQL용 컴퓨팅 인스턴스로 M5 유형 EC2 인스턴스를 사용하는 것이 좋습니다. 대기 컴퓨팅 인스턴스는 항상 FSx HA 클러스터용으로 구축된 패시브(대기) 파일 시스템과 동일한 존에 배포되어야 합니다.
- \* FSx 스토리지 HA 클러스터 단일 또는 다중 영역 배포. \* 이러한 테스트 및 검증에서는 단일 AWS 가용성 영역에 FSx HA 클러스터를 구축했습니다. 프로덕션 배포를 위해 FSx HA 쌍을 두 가지 가용성 영역에 배포하는 것이 좋습니다. 운영 및 대기 간에 특정 거리가 필요한 경우 다른 지역에서 비즈니스 연속성을 위한 재해 복구 대기 HA 쌍을 설정할 수 있습니다. FSx HA 클러스터는 스토리지 레벨 이중화를 제공하기 위해 액티브-패시브 파일 시스템 쌍으로 미러링되는 HA 쌍으로 프로비저닝됩니다.
- \* PostgreSQL 데이터 및 로그 배치 \* 일반적인 PostgreSQL 배포는 데이터 및 로그 파일에 대해 동일한 루트 디렉토리 또는 볼륨을 공유합니다. 테스트 및 검증에서 PostgreSQL 데이터를 분리하고 성능을 위해 두 개의 개별 볼륨에 로그인합니다. 데이터 디렉토리에서 PostgreSQL Wal 로그 및 보관된 Wal 로그를 호스팅하는 로그 디렉토리 또는 볼륨을 가리키는 소프트 링크가 사용됩니다.
- \* PostgreSQL 서비스 시작 지연 타이머 \* 이 솔루션은 NFS 마운트 볼륨을 사용하여 PostgreSQL 데이터베이스 파일과 Wal 로그 파일을 저장합니다. 데이터베이스 호스트를 재부팅하는 동안 볼륨이 마운트되지 않은 상태에서 PostgreSQL 서비스가 시작하려고 할 수 있습니다. 이로 인해 데이터베이스 서비스 시작 오류가 발생합니다. PostgreSQL 데이터베이스를 올바르게 시작하려면 10 ~ 15초의 타이머 지연이 필요합니다.
- \* 비즈니스 연속성을 위한 \* RPO/RTO. \* DR을 위해 기본에서 대기로의 FSx 데이터 복제는 비동기를 기반으로 합니다. 즉, RPO는 Snapshot 백업 및 SnapMirror 복제 빈도에 따라 달라집니다. 스냅샷 복사본 및 SnapMirror 복제 빈도가 높아지면 RPO가 감소합니다. 따라서 재해 발생 시 잠재적인 데이터 손실과 스토리지 비용 증가 간에 균형을 유지할 수 있습니다. RPO에 대해 스냅샷 복사본과 SnapMirror 복제를 최소 5분 간격으로 구현할 수 있으며 PostgreSQL은 일반적으로 RTO에 대한 1분 이내에 DR 대기 사이트에서 복구할 수 있습니다.
- \* 데이터베이스 백업. \* PostgreSQL 데이터베이스가 구축되거나 On-Premrs 데이터 센터에서 AWS FSx 스토리지로 마이그레이션된 후 데이터를 보호하기 위해 FSx HA 쌍에서 자동 동기화됩니다. 재해 발생 시 복제된 대기 사이트로 데이터를 더욱 안전하게 보호할 수 있습니다. 장기 백업 보존 또는 데이터 보호를 위해 내장된 PostgreSQL pg\_basebackup 유틸리티를 사용하여 S3 Blob 스토리지로 포팅할 수 있는 전체 데이터베이스 백업을 실행하는 것이 좋습니다.

## 솔루션 구축

이 솔루션 배포는 아래에 설명된 자세한 지침을 따라 NetApp Ansible 기반 자동화 툴킷을 사용하여 자동으로 완료할 수 있습니다.

1. 자동화 도구 키트 readme.md의 지침을 읽으십시오 "[NA\\_PostgreSQL\\_AWS\\_Deploy\\_HADR](#)".
2. 다음 비디오 단계별 안내 를 시청하십시오.

### 자동화된 PostgreSQL 배포 및 보호

1. 필요한 매개 변수 파일을 구성합니다 (hosts, host\_vars/host\_name.yml, fsx\_vars.yml)를 클릭합니다. 그런 다음 복사 버튼을 사용하여 Ansible 컨트롤러 호스트에 파일을 복사합니다.

## 자동 배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

1. AWS 계정이 설정되었으며 AWS 계정 내에 필요한 VPC 및 네트워크 세그먼트가 생성되었습니다.
2. AWS EC2 콘솔에서 2개의 EC2 Linux 인스턴스를 구축해야 합니다. 하나는 운영 사이트에, 다른 하나는 대기 DR 사이트에 운영 PostgreSQL DB 서버로 배포해야 합니다. 운영 및 대기 DR 사이트에서 컴퓨팅 이중화를 위해 2개의 추가 EC2 Linux 인스턴스를 대기 PostgreSQL DB 서버로 구축합니다. 환경 설정에 대한 자세한 내용은 이전 섹션의 아키텍처 다이어그램을 참조하십시오. 또한 를 검토합니다 "[Linux 인스턴스에 대한 사용자 가이드](#)" 를 참조하십시오.
3. AWS EC2 콘솔에서 PostgreSQL 데이터베이스 볼륨을 호스팅하기 위해 두 개의 FSx ONTAP 스토리지 HA 클러스터를 구축합니다. FSx 저장소 배포에 익숙하지 않은 경우 설명서를 참조하십시오 "[ONTAP 파일 시스템용 FSx 생성](#)" 을 참조하십시오.
4. Ansible 컨트롤러를 호스팅할 CentOS Linux VM을 구축합니다. Ansible 컨트롤러는 사내 또는 AWS 클라우드에 위치할 수 있습니다. 사내에 있는 경우 VPC, EC2 Linux 인스턴스 및 FSx 스토리지 클러스터에 SSH를 연결해야 합니다.
5. 리소스의 "RHEL/CentOS에서 CLI 배포를 위한 Ansible Control Node 설정" 섹션에 설명된 대로 Ansible 컨트롤러를 설정합니다 "[NetApp 솔루션 자동화 시작하기](#)".
6. 공용 NetApp GitHub 사이트에서 자동화 툴킷 복사본을 복제합니다.

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

1. 툴킷 루트 디렉토리에서 필수 구성 요소 플레이북을 실행하여 Ansible 컨트롤러용 필수 컬렉션 및 라이브러리를 설치합니다.

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

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

1. DB 호스트 변수 파일에 필요한 EC2 FSx 인스턴스 매개 변수를 검색합니다 `host_vars/*` 글로벌 변수 파일을 엽니다 `fsx_vars.yml` 구성.

호스트 파일을 구성합니다

운영 FSx ONTAP 클러스터 관리 IP 및 EC2 인스턴스 호스트 이름을 HOSTS 파일에 입력합니다.

```
# 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
```

**host\_name.yml** 파일을 **host\_vars** 폴더에 구성합니다

```
# 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"
```

**VAR** 폴더에서 글로벌 **FSX\_VAR.yml** 파일을 구성합니다

```
#####
##### 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 배포 및 HA/DR 설정

다음 작업은 PostgreSQL DB 서버 서비스를 구축하고 운영 EC2 DB 서버 호스트의 운영 사이트에서 데이터베이스를 초기화합니다. 그런 다음 대기 운영 EC2 DB 서버 호스트가 대기 사이트에 설정됩니다. 마지막으로, 재해 복구를 위해 운영 사이트 FSx 클러스터에서 대기 사이트 FSx 클러스터로 DB 볼륨 복제가 설정됩니다.

1. 운영 FSx 클러스터에 DB 볼륨을 생성하고 운영 EC2 인스턴스 호스트에서 PostgreSQL을 설정합니다.

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

2. 대기 DR EC2 인스턴스 호스트를 설정합니다.

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

3. FSx ONTAP 클러스터 피어링 및 데이터베이스 볼륨 복제를 설정합니다.

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

4. 이전 단계를 단일 단계의 PostgreSQL 배포 및 HA/DR 설정에 통합합니다.

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

5. 기본 또는 대기 사이트에서 대기 PostgreSQL DB 호스트를 설정하려면 hosts file [DR\_PostgreSQL] 섹션의 다른 모든 서버를 주석으로 추가한 다음 각 타겟 호스트(예: psql\_01ps 또는 운영 사이트의 대기 EC2 컴퓨팅 인스턴스)와 함께 PostgreSQL\_standby\_setup.yml 플레이북을 실행합니다. 과 같은 호스트 매개 변수 파일이 있는지 확인합니다 psql\_01ps.yml 가 에 구성되어 있습니다 host\_vars 디렉토리.

```
[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 데이터베이스 스냅샷 백업 및 대기 사이트로의 복제

대기 사이트에 대한 PostgreSQL 데이터베이스 스냅샷 백업 및 복제는 사용자 정의 간격으로 Ansible 컨트롤러에서 제어 및 실행될 수 있습니다. 간격이 5분 정도일 수 있다는 것이 검증되었습니다. 따라서 운영 사이트에서 장애가 발생할 경우 예약된 다음 스냅샷 백업 바로 전에 장애가 발생할 경우 5분 내에 데이터가 손실될 수 있습니다.

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

## DR을 위해 대기 사이트로 페일오버

PostgreSQL HA/DR 시스템을 DR 실습으로 테스트하려면 다음 플레이북을 실행하여 대기 사이트의 운영 스탠바이 EC2 DB 인스턴스에서 페일오버 및 PostgreSQL 데이터베이스 복구를 실행합니다. 실제 DR 시나리오에서는 실제로 DR 사이트로 페일오버하는 경우에도 동일하게 실행합니다.

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

장애 조치 테스트 후 복제된 **DB** 볼륨을 다시 동기화합니다

페일오버 테스트 후 재동기화를 실행하여 데이터베이스 볼륨 SnapMirror 복제를 다시 설정합니다.

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

## EC2 컴퓨팅 인스턴스 장애로 인해 운영 **EC2 DB** 서버에서 대기 **EC2 DB** 서버로 페일오버

NetApp은 수동 페일오버 실행 또는 라이선스가 필요할 수 있는 잘 구축된 OS 클러스터 소프트웨어를 사용할 것을 권장합니다.

## 추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹 사이트를 검토하십시오.

- NetApp ONTAP용 Amazon FSx

["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-8225-4784bced9843&sc\\_channel=ps&s\\_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef\\_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw\\_wcB:G:s&s\\_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2](https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=ps&s_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2)

- NetApp 솔루션 자동화

["https://docs.netapp.com/us-en/netapp-solutions/automation/automation\\_introduction.html"](https://docs.netapp.com/us-en/netapp-solutions/automation/automation_introduction.html)

## TR-4722: NetApp ONTAP 모범 사례 기반 MySQL 데이터베이스

Anup Bharti, Manohar Kulkarni, Jeffrey Steiner NetApp

MySQL 및 MariaDB 및 Percona 등의 변종은 많은 엔터프라이즈 애플리케이션에 널리 사용됩니다. 이러한 애플리케이션은 글로벌 소셜 네트워킹 사이트, 대규모 전자 상거래 시스템부터 수천 개의 데이터베이스 인스턴스가 포함된 SMB 호스팅 시스템에 이르기까지 다양합니다. 이 문서는 NetApp® ONTAP® 데이터 관리 소프트웨어에서 MySQL을 구축하기 위한 구성 요구 사항을 설명하고 조정 및 스토리지 구성을 조정하는 방법에 대한 지침을 제공합니다. 본 보고서에 명시된 환경, 구성, 버전이 현재 사용하는 환경을 지원하는지 확인하려면 상호 운용성 매트릭스 툴(IMT)을 참조하십시오.

"TR-4722: NetApp ONTAP 모범 사례 기반 MySQL 데이터베이스"

## 저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

## 상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.