



DB Automation 툴킷

NetApp Solutions

NetApp
April 20, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions/databases/automation_ora_clone_lifecycle.html on April 20, 2024. Always check docs.netapp.com for the latest.

목차

- DB Automation 툴킷 1
 - SnapCenter Oracle 클론 라이프사이클 자동화..... 1
 - 자동화된 Oracle 마이그레이션 5
 - AWS FSx ONTAP에서 자동화된 Oracle HA/DR 9
 - AWS FSx ONTAP 클러스터와 EC2 인스턴스 프로비저닝 14

DB Automation 툴킷

SnapCenter Oracle 클론 라이프사이클 자동화

Allen Cao, Niyaz Mohamed, NetApp

목적

고객은 데이터베이스를 위해 NetApp ONTAP 스토리지의 FlexClone 기능을 적극 활용하여 스토리지 비용을 대폭 절감할 수 있었습니다. 이 Ansible 기반 툴킷은 간소화된 라이프사이클 관리를 위한 NetApp SnapCenter 명령줄 유틸리티를 사용하여 클론 복제된 Oracle 데이터베이스의 설정, 클론 복제 및 업데이트를 일정에 따라 자동화합니다. 이 툴킷은 ONTAP 스토리지에 프리세일즈스나 퍼블릭 클라우드에 구축되고 NetApp SnapCenter UI 툴을 통해 관리되는 Oracle 데이터베이스에 적용됩니다.

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- Oracle 데이터베이스 클론 사양 구성 파일을 설정합니다.
- 사용자 정의 일정에 따라 클론 Oracle 데이터베이스 생성 및 업데이트

대상

이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

- SnapCenter를 통해 Oracle 데이터베이스를 관리하는 DBA입니다.
- SnapCenter를 통해 ONTAP 스토리지를 관리하는 스토리지 관리자
- SnapCenter UI에 액세스할 수 있는 애플리케이션 소유자

라이선스

이 GitHub 리포지토리의 콘텐츠를 액세스, 다운로드, 설치 또는 사용하면 에 설명된 사용권 조항에 동의하는 것입니다 "[라이선스 파일](#)".



이 GitHub 저장소의 콘텐츠와 파생 저작물을 제작 및/또는 공유하는 데는 일정한 제한이 있습니다. 콘텐츠를 사용하기 전에 라이선스 약관을 읽었는지 확인하십시오. 모든 약관에 동의하지 않는 경우 이 리포지토리에 있는 콘텐츠에 액세스하거나 다운로드하거나 사용하지 마십시오.

솔루션 구축

배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

```
Ansible controller:
  Ansible v.2.10 and higher
  ONTAP collection 21.19.1
  Python 3
  Python libraries:
    netapp-lib
    xmltodict
    jmespath
```

```
SnapCenter server:
  version 5.0
  backup policy configured
  Source database protected with a backup policy
```

```
Oracle servers:
  Source server managed by SnapCenter
  Target server managed by SnapCenter
  Target server with identical Oracle software stack as source server
  installed and configured
```

툴킷을 다운로드합니다

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

Ansible 타겟 호스트 파일 구성

이 톨킷에는 Ansible 플레이북이 실행되는 타겟을 정의하는 HOSTS 파일이 포함되어 있습니다. 일반적으로 대상 Oracle 클론 호스트입니다. 다음은 예제 파일입니다. 호스트 항목에는 타겟 호스트 IP 주소와 관리자 사용자가 호스트에 액세스하여 클론 또는 새로 고침 명령을 실행하기 위한 ssh 키가 포함됩니다.

oracle 클론 호스트 수

```
[clone_1]
ora_04.cie.netapp.com ansible_host=10.61.180.29
ansible_ssh_private_key_file=ora_04.pem
```

```
[clone_2]
[clone_3]
```

글로벌 변수 설정

Ansible 플레이북은 여러 가변 파일에서 다양한 입력을 사용합니다. 다음은 전역 변수 파일 vars.yml의 예입니다.

```
# ONTAP specific config variables
# SnapCtr specific config variables
```

```
snapctr_usr: xxxxxxxx
snapctr_pwd: 'xxxxxxx'
```

```
backup_policy: 'Oracle Full offline Backup'
# Linux specific config variables
# Oracle specific config variables
```

호스트 변수 구성

호스트 변수는 이름이 {{host_name}}.yml인 host_vars 디렉토리에 정의됩니다. 다음은 일반적인 구성을 보여주는 대상 Oracle 호스트 변수 파일 ora_04.cie.netapp.com.yml 예입니다.

```
# User configurable Oracle clone db host specific parameters
```

```
# Source database to clone from
source_db_sid: NTAP1
source_db_host: ora_03.cie.netapp.com
```

```
# Clone database
clone_db_sid: NTAP1DEV
```

```
snapctr_obj_id: '{{ source_db_host }}\{{ source_db_sid }}'
```

추가 클론 대상 **Oracle** 서버 구성

클론 대상 Oracle 서버는 소스 Oracle 서버와 동일한 Oracle 소프트웨어 스택을 설치 및 패치해야 합니다. Oracle user.bash_profile에 \$ORACLE_BASE 및 \$ORACLE_HOME이 구성되어 있습니다. 또한 \$ORACLE_HOME 변수는 소스 Oracle 서버 설정과 일치해야 합니다. 다음은 예제입니다.

```
# .bash_profile
```

```
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

```
# User specific environment and startup programs
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/NTAP1
```

플레이북 실행

SnapCenter CLI 유틸리티를 사용하여 Oracle 데이터베이스 클론 라이프사이클을 실행하기 위한 플레이북은 총 3개입니다.

1. Ansible 컨트롤러 사전 요구사항 설치 - 한 번만.

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

2. 클론 설정 지정 파일 - 한 번만.

```
ansible-playbook -i hosts clone_1_setup.yml -u admin -e  
@vars/vars.yml
```

3. crontab에서 정기적으로 클론 데이터베이스를 생성하고 업데이트하여 셸 스크립트를 사용하여 업데이트 플레이북을 호출합니다.

```
0 */4 * * * /home/admin/na_oracle_clone_lifecycle/clone_1_refresh.sh
```

추가 클론 데이터베이스의 경우 별도의 clone_n_setup.yml 및 clone_n_refresh.yml 및 clone_n_refresh.sh를 생성합니다. 이에 따라 host_vars 디렉토리에 Ansible 타겟 호스트와 hostname.yml 파일을 구성합니다.

추가 정보를 찾을 수 있는 위치

NetApp 솔루션 자동화에 대한 자세한 내용은 다음 웹 사이트를 참조하십시오 ["NetApp 솔루션 자동화"](#)

자동화된 Oracle 마이그레이션

NetApp 솔루션 엔지니어링 팀

목적

이 툴킷은 FSx ONTAP 스토리지 및 EC2 컴퓨팅 인스턴스를 타겟 인프라로 사용하여 온프레미스에서 AWS 클라우드로 Oracle 데이터베이스를 자동으로 마이그레이션합니다. 이 고객은 이미 CDB/PDB 모델에 온프레미스 Oracle 데이터베이스를 구축한 것으로 가정합니다. 이 툴킷을 통해 고객은 Oracle PDB 재배포 절차를 사용하여 Oracle 호스트의 컨테이너 데이터베이스에서 이름이 지정된 PDB를 재배포할 수 있으며 최대 가용성 옵션을 사용할 수 있습니다. 즉, 모든 온프레미스 스토리지 어레이의 소스 PDB는 서비스 중단을 최소화하면서 새 컨테이너 데이터베이스로 재배포됩니다. Oracle 재배포 절차는 데이터베이스가 온라인 상태일 때 Oracle 데이터 파일을 이동합니다. 이후 모든 데이터 파일이 AWS 클라우드로 이동할 때 전환 시점에 사용자 세션을 사내에서 재배포된 데이터베이스 서비스로 다시 라우팅합니다. 밑줄이 그어진 기술은 검증된 Oracle PDB 핫 클론 방법론입니다.



마이그레이션 툴킷은 AWS 클라우드 인프라에서 개발 및 검증되었지만 Oracle 애플리케이션 레벨 솔루션을 기반으로 합니다. 따라서 이 툴킷은 Azure, GCP 등의 다른 퍼블릭 클라우드 플랫폼에 적용할 수 있습니다

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- 온프레미스 소스 DB 서버에서 마이그레이션 사용자를 생성하고 필요한 권한을 부여합니다.
- 소스 PDB가 전환될 때까지 온라인 상태인 동안 온-프레미스 CDB에서 클라우드의 대상 CDB로 PDB를 재배포합니다.

대상

이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

- Oracle 데이터베이스를 온라인으로 이전하는 DBA입니다.
- Oracle 데이터베이스를 AWS 클라우드로 마이그레이션하는 데 관심이 있는 데이터베이스 솔루션 설계자
- Oracle 데이터베이스를 지원하는 AWS FSx ONTAP 스토리지를 관리하는 스토리지 관리자
- Oracle 데이터베이스를 현재 베타 버전에서 AWS 클라우드로 마이그레이션하고자 하는 애플리케이션 소유자

라이선스

이 GitHub 리포지토리의 콘텐츠를 액세스, 다운로드, 설치 또는 사용하면 에 설명된 사용권 조항에 동의하는 것입니다 "[라이선스 파일](#)".



이 GitHub 저장소의 콘텐츠와 파생 저작물을 제작 및/또는 공유하는 데는 일정한 제한이 있습니다. 콘텐츠를 사용하기 전에 라이선스 약관을 읽었는지 확인하십시오. 모든 약관에 동의하지 않는 경우 이 리포지토리에 있는 콘텐츠에 액세스하거나 다운로드하거나 사용하지 마십시오.

솔루션 구축

배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

```
Ansible v.2.10 and higher
ONTAP collection 21.19.1
Python 3
Python libraries:
  netapp-lib
  xmltodict
  jmespath
```

```
Source Oracle CDB with PDBs on-premises
Target Oracle CDB in AWS hosted on FSx and EC2 instance
Source and target CDB on same version and with same options installed
```

```
Network connectivity
  Ansible controller to source CDB
  Ansible controller to target CDB
  Source CDB to target CDB on Oracle listener port (typical 1521)
```

툴킷을 다운로드합니다

```
git clone https://github.com/NetApp/na_ora_aws_migration.git
```

호스트 변수 구성

호스트 변수는 이름이 {{host_name}}.yml인 host_vars 디렉토리에 정의됩니다. 일반적인 구성을 보여 주는 호스트 변수 파일 HOST_NAME.yml 예가 포함되어 있습니다. 주요 고려 사항은 다음과 같습니다.

```
Source Oracle CDB - define host specific variables for the on-prem CDB
ansible_host: IP address of source database server host
source_oracle_sid: source Oracle CDB instance ID
source_pdb_name: source PDB name to migrate to cloud
source_file_directory: file directory of source PDB data files
target_file_directory: file directory of migrated PDB data files
```

```
Target Oracle CDB - define host specific variables for the target CDB
including some variables for on-prem CDB
ansible_host: IP address of target database server host
target_oracle_sid: target Oracle CDB instance ID
target_pdb_name: target PDB name to be migrated to cloud (for max
availability option, the source and target PDB name must be the same)
source_oracle_sid: source Oracle CDB instance ID
source_pdb_name: source PDB name to be migrated to cloud
source_port: source Oracle CDB listener port
source_oracle_domain: source Oracle database domain name
source_file_directory: file directory of source PDB data files
target_file_directory: file directory of migrated PDB data files
```

DB 서버 호스트 파일 구성

AWS EC2 인스턴스는 기본적으로 호스트 이름 지정을 위해 IP 주소를 사용합니다. Ansible용 hosts 파일에서 다른 이름을 사용하는 경우 소스 및 대상 서버에 대해 /etc/hosts 파일에서 호스트 이름 확인을 설정합니다. 다음은 예제입니다.

```
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 source_db_server
172.30.15.107 target_db_server
```

Playbook 실행 - 순서대로 실행

1. Ansible 컨트롤러 사전 요구사항을 설치합니다.

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

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

2. 온프레미스 서버에 대해 사전 마이그레이션 작업 실행 - 관리자가 sudo 권한이 있는 온프레미스 Oracle 호스트에 연결하기 위한 ssh 사용자라고 가정합니다.

```
ansible-playbook -i hosts ora_pdb_relocate.yml -u admin -k -K -t  
ora_pdb_relo_onprem
```

3. 온프레미스 CDB에서 AWS EC2 인스턴스의 타겟 CDB로의 Oracle PDB 재배포 실행 - EC2 DB 인스턴스 연결의 경우 EC2 사용자 가정, EC2 사용자 ssh 키 쌍의 db1.pem

```
ansible-playbook -i hosts ora_pdb_relocate.yml -u ec2-user --private  
-key db1.pem -t ora_pdb_relo_primary
```

추가 정보를 찾을 수 있는 위치

NetApp 솔루션 자동화에 대한 자세한 내용은 다음 웹 사이트를 참조하십시오 ["NetApp 솔루션 자동화"](#)

AWS FSx ONTAP에서 자동화된 Oracle HA/DR

NetApp 솔루션 엔지니어링 팀

목적

이 툴킷은 FSx for ONTAP 스토리지 및 EC2 컴퓨팅 인스턴스를 통해 AWS 클라우드에 구축된 Oracle 데이터베이스용고가용성 및 재해 복구(HA/DR) 환경을 설정하고 관리하는 작업을 자동화합니다.

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- HA/DR 타겟 호스트 커널 구성, 소스 서버 호스트와 일치하도록 Oracle 구성 설정
- FSx ONTAP - 클러스터 피어링, SVM 피어링, Oracle 볼륨 SnapMirror 관계 설정을 소스에서 타겟으로 하여 설정합니다.
- crontab에서 스냅샷 실행을 통해 Oracle 데이터베이스 데이터를 백업합니다
- 스냅샷을 통해 Oracle 데이터베이스 아카이브 로그 백업 - crontab에서 실행
- HA/DR 호스트에서 페일오버 및 복구를 실행하고 HA/DR 환경을 검증합니다

- 파일오버 테스트 후 재동기화를 실행 - HA/DR 모드에서 데이터베이스 볼륨 SnapMirror 관계를 다시 설정합니다

대상

이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

- 고가용성, 데이터 보호 및 재해 복구를 위해 AWS에서 Oracle 데이터베이스를 설정한 DBA
- AWS 클라우드의 스토리지 수준 Oracle HA/DR 솔루션에 관심이 있는 데이터베이스 솔루션 설계자
- Oracle 데이터베이스를 지원하는 AWS FSx ONTAP 스토리지를 관리하는 스토리지 관리자
- AWS FSx/EC2 환경에서 HA/DR을 위해 Oracle 데이터베이스를 가동하려는 애플리케이션 소유자.

라이선스

이 GitHub 리포지토리의 콘텐츠를 액세스, 다운로드, 설치 또는 사용하면 에 설명된 사용권 조항에 동의하는 것입니다 ["라이선스 파일"](#).



이 GitHub 저장소의 콘텐츠와 파생 저작물을 제작 및/또는 공유하는 데는 일정한 제한이 있습니다. 콘텐츠를 사용하기 전에 라이선스 약관을 읽었는지 확인하십시오. 모든 약관에 동의하지 않는 경우 이 리포지토리에 있는 콘텐츠에 액세스하거나 다운로드하거나 사용하지 마십시오.

솔루션 구축

배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

```
Ansible v.2.10 and higher
ONTAP collection 21.19.1
Python 3
Python libraries:
  netapp-lib
  xmltodict
  jmespath
```

```
AWS FSx storage as is available
```

```
AWS EC2 Instance
  RHEL 7/8, Oracle Linux 7/8
  Network interfaces for NFS, public (internet) and optional management
  Existing Oracle environment on source, and the equivalent Linux
  operating system at the target
```

툴킷을 다운로드합니다

```
git clone https://github.com/NetApp/na_ora_hadr_failover_resync.git
```

글로벌 변수 설정

Ansible 플레이북은 상황에 따라 다릅니다. 예제 글로벌 변수 파일 `FSX_VARS_EXAMPLE.yml`이 포함되어 일반적인 구성을 보여 줍니다. 주요 고려 사항은 다음과 같습니다.

ONTAP - retrieve FSx storage parameters using AWS FSx console for both source and target FSx clusters.

cluster name: source/destination

cluster management IP: source/destination

inter-cluster IP: source/destination

vserver name: source/destination

vserver management IP: source/destination

NFS lifs: source/destination

cluster credentials: fsxadmin and vsadmin pwd to be updated in roles/ontap_setup/defaults/main.yml file

Oracle database volumes - they should have been created from AWS FSx console, volume naming should follow strictly with following standard:

Oracle binary: {{ host_name }}_bin, generally one lun/volume

Oracle data: {{ host_name }}_data, can be multiple luns/volume, add additional line for each additional lun/volume in variable such as {{ host_name }}_data_01, {{ host_name }}_data_02 ...

Oracle log: {{ host_name }}_log, can be multiple luns/volume, add additional line for each additional lun/volume in variable such as {{ host_name }}_log_01, {{ host_name }}_log_02 ...

host_name: as defined in hosts file in root directory, the code is written to be specifically matched up with host name defined in host file.

Linux and DB specific global variables - keep it as is.

Enter redhat subscription if you have one, otherwise leave it black.

호스트 변수 구성

호스트 변수는 이름이 {{host_name}}.yml인 host_vars 디렉토리에 정의됩니다. 일반적인 구성을 보여 주는 호스트 변수 파일 HOST_NAME.yml 예가 포함되어 있습니다. 주요 고려 사항은 다음과 같습니다.

```
Oracle - define host specific variables when deploying Oracle in
multiple hosts concurrently
  ansible_host: IP address of database server host
  log_archive_mode: enable archive log archiving (true) or not (false)
  oracle_sid: Oracle instance identifier
  pdb: Oracle in a container configuration, name pdb_name string and
number of pdbs (Oracle allows 3 pdbs free of multitenant license fee)
  listener_port: Oracle listener port, default 1521
  memory_limit: set Oracle SGA size, normally up to 75% RAM
  host_datastores_nfs: combining of all Oracle volumes (binary, data,
and log) as defined in global vars file. If multi luns/volumes, keep
exactly the same number of luns/volumes in host_var file
```

```
Linux - define host specific variables at Linux level
  hugepages_nr: set hugepage for large DB with large SGA for
performance
  swap_blocks: add swap space to EC2 instance. If swap exist, it will
be ignored.
```

DB 서버 호스트 파일 구성

AWS EC2 인스턴스는 기본적으로 호스트 이름 지정을 위해 IP 주소를 사용합니다. Ansible용 hosts 파일에서 다른 이름을 사용하는 경우 소스 및 대상 서버에 대해 /etc/hosts 파일에서 호스트 이름 확인을 설정합니다. 다음은 예제입니다.

```
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhost4
::1         localhost localhost.localdomain localhost6
localhost6.localhost6
172.30.15.96 db1
172.30.15.107 db2
```

Playbook 실행 - 순서대로 실행

1. Ansible 컨트롤러 사전 설치를 참조하십시오.

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

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

2. 타겟 EC2 DB 인스턴스를 설정합니다.

```
ansible-playbook -i hosts ora_dr_setup.yml -u ec2-user --private-key  
db2.pem -e @vars/fsx_vars.yml
```

3. 소스 및 타겟 데이터베이스 볼륨 간에 FSx ONTAP SnapMirror 관계 설정

```
ansible-playbook -i hosts ontap_setup.yml -u ec2-user --private-key  
db2.pem -e @vars/fsx_vars.yml
```

4. crontab의 스냅샷을 통해 Oracle 데이터베이스 데이터 볼륨을 백업합니다.

```
10 * * * * cd /home/admin/na_ora_hadr_failover_resync &&  
/usr/bin/ansible-playbook -i hosts ora_replication_cg.yml -u ec2-  
user --private-key db1.pem -e @vars/fsx_vars.yml >>  
logs/snap_data_`date +%Y-%m%d-%H%M%S`.log 2>&1
```

5. crontab의 스냅샷을 통해 Oracle 데이터베이스 아카이브 로그 볼륨을 백업합니다.

```
0,20,30,40,50 * * * * cd /home/admin/na_ora_hadr_failover_resync &&  
/usr/bin/ansible-playbook -i hosts ora_replication_logs.yml -u ec2-  
user --private-key db1.pem -e @vars/fsx_vars.yml >>  
logs/snap_log_`date +%Y-%m%d-%H%M%S`.log 2>&1
```

6. 타겟 EC2 DB 인스턴스에서 페일오버를 실행하고 Oracle 데이터베이스를 복구합니다. - HA/DR 구성을 테스트하고 검증합니다.

```
ansible-playbook -i hosts ora_recovery.yml -u ec2-user --private-key  
db2.pem -e @vars/fsx_vars.yml
```

7. 페일오버 테스트 후 재동기화를 실행하고 복제 모드에서 데이터베이스 볼륨 SnapMirror 관계를 다시 설정합니다.

```
ansible-playbook -i hosts ontap_ora_resync.yml -u ec2-user --private-key db2.pem -e @vars/fsx_vars.yml
```

추가 정보를 찾을 수 있는 위치

NetApp 솔루션 자동화에 대한 자세한 내용은 다음 웹 사이트를 참조하십시오 ["NetApp 솔루션 자동화"](#)

AWS FSx ONTAP 클러스터와 EC2 인스턴스 프로비저닝

NetApp 솔루션 엔지니어링 팀

목적

이 툴킷은 다음에 데이터베이스 구축에 사용할 수 있는 AWS FSx ONTAP 스토리지 클러스터와 EC2 컴퓨팅 인스턴스의 프로비저닝 작업을 자동화합니다.

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- 사전 정의된 VPC 서브넷에서 AWS 클라우드에서 EC2 컴퓨팅 인스턴스를 프로비저닝하고 EC2 인스턴스 액세스용 ssh 키를 EC2-user로 설정합니다.
- 원하는 가용성 영역에서 AWS FSx ONTAP 스토리지 클러스터를 프로비저닝하고 스토리지 SVM을 구성하고 클러스터 관리자 사용자 fsxadmin 암호를 설정합니다.

대상

이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

- AWS EC2 환경에서 데이터베이스를 관리하는 DBA
- AWS EC2 에코시스템의 데이터베이스 구축에 관심이 있는 데이터베이스 솔루션 설계자
- 데이터베이스를 지원하는 AWS FSx ONTAP 스토리지를 관리하는 스토리지 관리자
- AWS EC2 에코시스템에 데이터베이스를 스탠드업하려는 애플리케이션 소유자.

라이선스

이 GitHub 리포지토리의 콘텐츠를 액세스, 다운로드, 설치 또는 사용하면 에 설명된 사용권 조항에 동의하는 것입니다 ["라이선스 파일"](#).



이 GitHub 저장소의 콘텐츠와 파생 저작물을 제작 및/또는 공유하는 데는 일정한 제한이 있습니다. 콘텐츠를 사용하기 전에 라이선스 약관을 읽었는지 확인하십시오. 모든 약관에 동의하지 않는 경우 이 리포지토리에 있는 콘텐츠에 액세스하거나 다운로드하거나 사용하지 마십시오.

솔루션 구축

배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

```
An Organization and AWS account has been setup in AWS public cloud
An user to run the deployment has been created
IAM roles has been configured
IAM roles granted to user to permit provisioning the resources
```

VPC and security configuration

```
A VPC has been created to host the resources to be provisioned
A security group has been configured for the VPC
A ssh key pair has been created for EC2 instance access
```

Network configuration

```
Subnets has been created for VPC with network segments assigned
Route tables and network ACL configured
NAT gateways or internet gateways configured for internet access
```

툴킷을 다운로드합니다

```
git clone https://github.com/NetApp/na_aws_fsx_ec2_deploy.git
```

연결 및 인증

이 툴킷은 AWS 클라우드 셸에서 실행되어야 합니다. AWS 클라우드 셸은 AWS 리소스를 손쉽게 관리, 탐색, 상호 작용할 수 있는 브라우저 기반 셸입니다. CloudShell은 콘솔 자격 증명으로 사전 인증됩니다. 공통 개발 및 운영 도구가 사전 설치되어 있으므로 로컬 설치 또는 구성이 필요하지 않습니다.

Terraform 공급자 .tf 및 기본 .tf 파일 구성

provider.tf는 Terraform이 API 호출을 통해 리소스를 프로비저닝하는 공급자를 정의합니다. main.tf는 프로비저닝할 리소스의 리소스와 속성을 정의합니다. 몇 가지 세부 정보는 다음과 같습니다.

```
provider.tf:
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.54.0"
    }
  }
}
```

```
main.tf:
resource "aws_instance" "ora_01" {
  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_tag
  }
}
....
```

Terraform 변수 .tf 및 Terraform.tfvars 구성

variables.tf 는 main.tf 에서 사용할 변수를 선언합니다. Terraform.tfvars에는 변수의 실제 값이 들어 있습니다. 다음은 몇 가지 예입니다.

```
variables.tf:
### EC2 instance variables ###
```

```
variable "ami" {
  type      = string
  description = "EC2 AMI image to be deployed"
}
```

```
variable "instance_type" {
  type      = string
  description = "EC2 instance type"
}
```

```
terraform.tfvars:
# EC2 instance variables
```

```
ami = "ami-06640050dc3f556bb" //RedHat 8.6 AMI
instance_type = "t2.micro"
ec2_tag = "ora_01"
subnet_id = "subnet-04f5fe7073ff514fb"
ssh_key_name = "sufi_new"
root_volume_size = 30
```

단계별 절차 - 순서대로 실행

1. AWS 클라우드 셸에 Terraform을 설치합니다.

```
git clone https://github.com/tfutils/tfenv.git ~/.tfenv
```

```
mkdir ~/bin
```

```
ln -s ~/.tfenv/bin/* ~/bin/
```

```
tfenv install
```

```
tfenv use 1.3.9
```

2. NetApp GitHub 공개 사이트에서 툴킷을 다운로드합니다

```
git clone https://github.com/NetApp-  
Automation/na_aws_fsx_ec2_deploy.git
```

3. init를 실행하여 Terraform을 초기화합니다

```
terraform init
```

4. 실행계획을 출력한다

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

5. 실행 계획을 적용합니다

```
terraform apply "main.plan"
```

6. 완료 시 리소스를 제거하려면 destroy를 실행합니다

```
terraform destroy
```

추가 정보를 찾을 수 있는 위치

NetApp 솔루션 자동화에 대한 자세한 내용은 다음 웹 사이트를 참조하십시오 "[NetApp 솔루션 자동화](#)"

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.