



Kits de herramientas de automatización DE BASES DE DATOS

NetApp Solutions

NetApp
April 26, 2024

This PDF was generated from https://docs.netapp.com/es-es/netapp-solutions/databases/automation_ora_clone_lifecycle.html on April 26, 2024. Always check docs.netapp.com for the latest.

Tabla de contenidos

- Kits de herramientas de automatización DE BASES DE DATOS 1
 - Automatización del ciclo de vida de clones de Oracle de SnapCenter 1
 - Migración Oracle automatizada 5
 - Oracle HA/recuperación ante desastres automatizada en AWS FSx ONTAP 9
 - Clúster de AWS FSx ONTAP y aprovisionamiento de instancias de EC2 15

Kits de herramientas de automatización DE BASES DE DATOS

Automatización del ciclo de vida de clones de Oracle de SnapCenter

Allen Cao, Niyaz Mohamed, NetApp

Específico

A los clientes les encanta la función FlexClone del almacenamiento de NetApp ONTAP para bases de datos con un ahorro considerable en costes de almacenamiento. Este kit de herramientas basado en Ansible automatiza la configuración, el clonado y la actualización de bases de datos de Oracle clonadas según lo programado mediante las utilidades de línea de comandos de NetApp SnapCenter para una gestión optimizada del ciclo de vida. Este kit de herramientas es aplicable a las bases de datos de Oracle puestas en marcha en sistemas de almacenamiento de ONTAP, ya sea en los recursos previos o en el cloud público y gestionadas mediante la herramienta de interfaz de usuario de NetApp SnapCenter.

Esta solución aborda los siguientes casos prácticos:

- Configurar el archivo de configuración de especificación del clon de la base de datos de Oracle.
- Cree y actualice la base de datos clonada de Oracle en la programación definida por el usuario.

Destinatarios

Esta solución está dirigida a las siguientes personas:

- Un administrador de bases de datos que gestiona bases de datos Oracle con SnapCenter.
- Un administrador de almacenamiento que gestiona el almacenamiento de ONTAP con SnapCenter.
- Propietario de una aplicación que tiene acceso a la interfaz de usuario de SnapCenter.

Licencia

Al acceder, descargar, instalar o utilizar el contenido de este repositorio de GitHub, usted acepta los términos de la Licencia establecidos en "[Archivo de licencia](#)".



Hay ciertas restricciones en cuanto a la producción y/o el intercambio de cualquier obra derivada con el contenido de este repositorio de GitHub. Asegúrese de leer los términos de la licencia antes de utilizar el contenido. Si NO ACEPTA TODOS LOS TÉRMINOS, NO ACCEDA, DESCARGA NI UTILICE EL CONTENIDO DE ESTE REPOSITORIO.

Puesta en marcha de la solución

Requisitos previos para la implementación

La implementación requiere los siguientes requisitos previos.

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

Descargue el kit de herramientas

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

Configuración de archivos de hosts de destino de Ansible

El kit de herramientas incluye un archivo de hosts que define los destinos con los que se ejecuta un libro de estrategia de Ansible. Normalmente, son los hosts de clonado de Oracle de destino. A continuación se muestra un archivo de ejemplo. Una entrada de host incluye la dirección IP del host de destino, así como la clave ssh para que un usuario administrador acceda al host para ejecutar el comando de clonación o actualización.

#Hosts del clon de Oracle

```
[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]
```

Configuración de variables globales

Los libros de estrategia de Ansible toman entradas variables de varios archivos variables. A continuación se muestra un ejemplo de archivo de variables globales vars.yml.

```
# 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
```

Configuración de variables de host

Las variables de host se definen en el directorio HOST_vars denominado {{ HOST_NAME }}.yml. A continuación se muestra un ejemplo del archivo de variables de host de Oracle de destino ora_04.cie.netapp.com.yml que muestra la configuración típica.

```
# 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 }}'
```

Configuración adicional del servidor de Oracle de destino de clonación

El servidor de Oracle de destino de clonación debe tener la misma pila de software de Oracle que el servidor de Oracle de origen instalado y con parches. El usuario de Oracle .bash_profile tiene \$ORACLE_BASE y \$ORACLE_HOME configurados. Además, la variable \$ORACLE_HOME debe coincidir con el valor del servidor de Oracle de origen. A continuación se muestra un ejemplo.

```
# .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
```

Ejecución de PlayBook

Existen tres libros de estrategia para ejecutar el ciclo de vida de los clones de base de datos de Oracle con las utilidades de la CLI de SnapCenter.

1. Instale los requisitos previos de la controladora de Ansible: Solo una vez.

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

2. Configurar archivo de especificación de clonación: Sólo una vez.

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

3. Cree y actualice la base de datos clonada regularmente desde crontab con un script de shell para llamar a un libro de estrategia de actualización.

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

Para una base de datos de clon adicional, cree clone_n_setup.yml y clone_n_refresh.yml independientes, y clone_n_refresh.sh. Configure los hosts de destino de Ansible y el archivo hostname.yml en el directorio host_vars según corresponda.

Dónde encontrar información adicional

Para obtener más información sobre la automatización de soluciones de NetApp, revise el siguiente sitio web ["Automatización de soluciones de NetApp"](#)

Migración Oracle automatizada

Equipo de ingeniería de soluciones de NetApp

Específico

Este kit de herramientas automatiza la migración de las bases de datos de Oracle de on-premises a la nube de AWS con el almacenamiento FSx ONTAP y la instancia de computación EC2 como infraestructura de destino. Asume que el cliente ya tiene una base de datos de Oracle en las instalaciones implementada en el modelo CDB/PDB. El kit de herramientas permitirá al cliente reubicar una PDB con nombre desde una base de datos de contenedor en un host de Oracle mediante el procedimiento de reubicación de Oracle PDB con una opción de disponibilidad máxima. Esto significa que la PDB de origen en cualquier cabina de almacenamiento local se reubica en una nueva base de datos de contenedores con una interrupción del servicio mínima. El procedimiento de reubicación de Oracle moverá los archivos de datos de Oracle mientras la base de datos está en línea. Posteriormente, redirige las sesiones de usuario de on-premises a los servicios de base de datos reubicados en el momento de cambiar cuando todos los archivos de datos se mueven a la nube de AWS. La tecnología subrayada es la metodología probada de clonado en caliente de PDB de Oracle.



Aunque el kit de herramientas de migración ha sido desarrollado y validado en la infraestructura de nube de AWS, se basa en soluciones de aplicaciones de Oracle. Por lo tanto, el kit de herramientas es aplicable a otras plataformas de cloud público, como Azure, GCP, etc.

Esta solución aborda los siguientes casos prácticos:

- Cree el usuario de migración y otorgue los privilegios necesarios en el servidor de base de datos de origen local.
- Reubique una PDB desde la CDB local a una CDB de destino en la nube mientras la PDB de origen está en línea hasta que se conmute.

Destinatarios

Esta solución está dirigida a las siguientes personas:

- Un administrador de bases de datos que migra las bases de datos de Oracle de los recursos locales a la nube de AWS.
- Un arquitecto de soluciones de bases de datos que esté interesado en la migración de bases de datos de Oracle de premisses al cloud de AWS.
- Un administrador de almacenamiento que gestiona el almacenamiento de ONTAP de AWS FSx compatible con las bases de datos de Oracle.
- Un propietario de aplicación al que le gusta migrar sus bases de datos de Oracle desde los premisas a la nube de AWS.

Licencia

Al acceder, descargar, instalar o utilizar el contenido de este repositorio de GitHub, usted acepta los términos de la Licencia establecidos en "[Archivo de licencia](#)".



Hay ciertas restricciones en cuanto a la producción y/o el intercambio de cualquier obra derivada con el contenido de este repositorio de GitHub. Asegúrese de leer los términos de la licencia antes de utilizar el contenido. Si NO ACEPTA TODOS LOS TÉRMINOS, NO ACCEDA, DESCARGA NI UTILICE EL CONTENIDO DE ESTE REPOSITORIO.

Puesta en marcha de la solución

Requisitos previos para la implementación

La implementación requiere los siguientes requisitos previos.

```
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)
```

Descargue el kit de herramientas

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

Configuración de variables de host

Las variables de host se definen en el directorio HOST_vars denominado {{ HOST_NAME }}.yml. Se incluye un ejemplo de archivo de variables host host host host host_name.yml para demostrar la configuración típica. A continuación se presentan algunas consideraciones clave:

```
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
```

Configuración del archivo de host del servidor de BASE DE DATOS

La instancia de AWS EC2 utiliza la dirección IP para la nomenclatura de hosts de forma predeterminada. Si utiliza un nombre diferente en el archivo de hosts para Ansible, configure la resolución de nomenclatura de host en el archivo /etc/hosts para el servidor de origen y el de destino. A continuación se muestra un ejemplo.

```
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
```

Ejecución de PlayBook: Se ejecuta en secuencia

1. Instale los requisitos previos de la controladora Ansible.

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

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

2. Ejecute tareas previas a la migración en el servidor local - asumiendo que admin es usuario ssh para la conexión al host de Oracle local con permiso sudo.

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

3. Ejecute la reubicación de Oracle PDB desde la CDB local hasta la CDB de destino en la instancia de AWS EC2, asumiendo EC2-user para la conexión a la instancia de la base de datos EC2 y db1.pem con pares de claves ssh de EC2 usuarios.

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

Dónde encontrar información adicional

Para obtener más información sobre la automatización de soluciones de NetApp, revise el siguiente sitio web ["Automatización de soluciones de NetApp"](#)

Oracle HA/recuperación ante desastres automatizada en AWS FSx ONTAP

Equipo de ingeniería de soluciones de NetApp

Específico

Este kit de herramientas automatiza las tareas de configuración y gestión de un entorno de alta disponibilidad y recuperación ante desastres para la base de datos de Oracle implementada en la nube de AWS con el almacenamiento de FSx para ONTAP e instancias de computación EC2.

Esta solución aborda los siguientes casos prácticos:

- Configurar host de destino HA/DR: Configuración del núcleo, configuración de Oracle para que coincida con el host del servidor de origen.
- Configuración de FSx ONTAP: Paridad de clústeres, paridad de Vserver y configuración de las relaciones de snapmirror de Oracle Volumes, desde el origen hasta el destino.

- Realizar una copia de seguridad de los datos de la base de datos de Oracle a través de la instantánea
- Realizar una copia de seguridad de archive log de la base de datos Oracle mediante snapshot - ejecutar fuera de crontab
- Ejecutar la recuperación tras fallos y la recuperación en host HA/DR: Probar y validar el entorno HA/DR
- Ejecutar resincronización después de la prueba de conmutación al nodo de respaldo y restablecer la relación de snapmirror para volúmenes de base de datos en modo HA/recuperación ante desastres

Destinatarios

Esta solución está dirigida a las siguientes personas:

- Un administrador de bases de datos que configura la base de datos de Oracle en AWS para alta disponibilidad, protección de datos y recuperación ante desastres.
- Un arquitecto de soluciones de bases de datos que esté interesado en la solución de recuperación ante desastres/alta disponibilidad de Oracle de nivel de almacenamiento en el cloud de AWS.
- Un administrador de almacenamiento que gestiona el almacenamiento de ONTAP de AWS FSx compatible con las bases de datos de Oracle.
- Propietario de una aplicación al que le gusta poner en marcha la base de datos de Oracle para alta disponibilidad/recuperación ante desastres en un entorno AWS FSx/EC2.

Licencia

Al acceder, descargar, instalar o utilizar el contenido de este repositorio de GitHub, usted acepta los términos de la Licencia establecidos en ["Archivo de licencia"](#).



Hay ciertas restricciones en cuanto a la producción y/o el intercambio de cualquier obra derivada con el contenido de este repositorio de GitHub. Asegúrese de leer los términos de la licencia antes de utilizar el contenido. Si NO ACEPTA TODOS LOS TÉRMINOS, NO ACCEDA, DESCARGA NI UTILICE EL CONTENIDO DE ESTE REPOSITORIO.

Puesta en marcha de la solución

Requisitos previos para la implementación

La implementación requiere los siguientes requisitos previos.

```
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
```

Descargue el kit de herramientas

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

Configuración de variables globales

Los libros de estrategia de Ansible se basan en variables. Se incluye un ejemplo de archivo de variable global `fsx_vars_EXAMPLE.yml` para demostrar la configuración típica. A continuación se presentan algunas consideraciones clave:

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.

Configuración de variables de host

Las variables de host se definen en el directorio HOST_vars denominado {{ HOST_NAME }}.yml. Se incluye un ejemplo de archivo de variables host host host host host_name.yml para demostrar la configuración típica. A continuación se presentan algunas consideraciones clave:

```
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.
```

Configuración del archivo de host del servidor de BASE DE DATOS

La instancia de AWS EC2 utiliza la dirección IP para la nomenclatura de hosts de forma predeterminada. Si utiliza un nombre diferente en el archivo de hosts para Ansible, configure la resolución de nomenclatura de host en el archivo /etc/hosts para los servidores de origen y de destino. A continuación se muestra un ejemplo.

```
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 db1
172.30.15.107 db2
```

Ejecución de PlayBook: Se ejecuta en secuencia

1. Instale los requisitos previos de la controladora Ansible.

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

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

2. Configurar la instancia de base de datos EC2 de destino.

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

3. Configura la relación de snapmirror de FSx ONTAP entre los volúmenes de las bases de datos de origen y destino.

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

4. Realizar backups de volúmenes de datos de la base de datos de Oracle mediante la instantánea desde crontab.

```
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. Realizar un backup de volúmenes de archive log de la base de datos de Oracle a través de la instantánea desde crontab.

```
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. Ejecutar failover y recuperar la base de datos Oracle en la instancia de la base de datos EC2 de destino: Probar y validar la configuración HA/DR.

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


7. Ejecutar resincronización después de la prueba de conmutación al nodo de respaldo y restablecer la relación de snapmirror de volúmenes de base de datos en modo de replicación.

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

Dónde encontrar información adicional

Para obtener más información sobre la automatización de soluciones de NetApp, revise el siguiente sitio web ["Automatización de soluciones de NetApp"](#)

Clúster de AWS FSx ONTAP y aprovisionamiento de instancias de EC2

Equipo de ingeniería de soluciones de NetApp

Específico

Este kit de herramientas automatiza las tareas de aprovisionamiento de un clúster de almacenamiento de AWS FSx ONTAP y una instancia de computación de EC2, que posteriormente se puede utilizar para la puesta en marcha de bases de datos.

Esta solución aborda los siguientes casos prácticos:

- Aprovechone una instancia de computación de EC2 en la nube de AWS en una subred de VPC predefinida y establezca la clave ssh para el acceso de instancia de EC2 como usuario EC2.
- Aprovechona un clúster de almacenamiento de AWS FSx ONTAP en las zonas de disponibilidad deseadas y configura una SVM de almacenamiento y establece la contraseña fsxadmin del usuario administrador del clúster.

Destinatarios

Esta solución está dirigida a las siguientes personas:

- Un administrador de bases de datos que gestiona bases de datos en un entorno AWS EC2.
- Un arquitecto de la solución de bases de datos interesado en la puesta en marcha de bases de datos en el ecosistema AWS EC2.
- Un administrador de almacenamiento que gestiona el almacenamiento de AWS FSx ONTAP compatible con bases de datos.
- Propietario de aplicaciones al que le gusta mantener su base de datos en el ecosistema EC2 de AWS.

Licencia

Al acceder, descargar, instalar o utilizar el contenido de este repositorio de GitHub, usted acepta los términos de la Licencia establecidos en ["Archivo de licencia"](#).



Hay ciertas restricciones en cuanto a la producción y/o el intercambio de cualquier obra derivada con el contenido de este repositorio de GitHub. Asegúrese de leer los términos de la licencia antes de utilizar el contenido. Si NO ACEPTA TODOS LOS TÉRMINOS, NO ACCEDA, DESCARGA NI UTILICE EL CONTENIDO DE ESTE REPOSITORIO.

Puesta en marcha de la solución

Requisitos previos para la implementación

La implementación requiere los siguientes requisitos previos.

```
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
```

Descargue el kit de herramientas

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

Conectividad y autenticación

Se supone que el kit de herramientas se ejecuta desde una shell de cloud de AWS. AWS Cloud Shell es un shell basado en navegador que facilita la administración, exploración e interacción segura con sus recursos de AWS. CloudShell se autentica previamente con sus credenciales de consola. Las herramientas comunes de desarrollo y operaciones están preinstaladas, por lo que no se requiere instalación o configuración local.

Configuración de archivos Terraform provider.tf y main.tf

El proveedor.tf define el proveedor que Terraform está aprovisionando recursos a través de llamadas API. El main.tf define los recursos y atributos de los recursos que se van a aprovisionar. A continuación se presentan algunos detalles:

```
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
  }
}
....
```

Configuración de Terraform Variables.tf y terraform.tfvars

El archivo Variables.tf declara las variables que se van a utilizar en main.tf. El terraform.tfvars contiene los valores reales para las variables. A continuación se muestran algunos ejemplos:

```
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
```

Procedimientos paso a paso: Ejecutados en secuencia

1. Instale Terraform en el shell de cloud de AWS.

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

```
mkdir ~/bin
```

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

```
tfenv install
```

```
tfenv use 1.3.9
```

2. Descargue el kit de herramientas del sitio público de GitHub de NetApp

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

3. Ejecute init para inicializar Terraform

```
terraform init
```

4. Genere la salida del plan de ejecución

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

5. Aplique el plan de ejecución

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

6. Ejecute destroy para eliminar los recursos cuando haya terminado

```
terraform destroy
```

Dónde encontrar información adicional

Para obtener más información sobre la automatización de soluciones de NetApp, revise el siguiente sitio web ["Automatización de soluciones de NetApp"](#)

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.