TR-4998: Oracle HA in AWS EC2 with Pacemaker Clustering and FSx ONTAP

NetApp database solutions

NetApp
August 04, 2025

Table of Contents

TR-4998: Oracle HA in AWS EC2 with Pacemaker Clustering and FSx ONTAP

Allen Cao, Niyaz Mohamed, NetApp

This solution provides an overview and details for enabling Oracle high availability (HA) in AWS EC2 with Pacemaker clustering on Redhat Enterprise Linux (RHEL) and Amazon FSx ONTAP for the database storage HA via NFS protocol.

Purpose

Many customers who strive to self-manage and run Oracle in the public cloud need to overcome a few challenges. One of those challenges is enabling high availability for the Oracle database. Traditionally, Oracle customers rely on an Oracle database feature called "Real Application Cluster" or RAC for active-active transaction support on multiple cluster nodes. One failed node would not stall application processing. Unfortunately, Oracle RAC implementation is not readily available or supported in many popular public clouds such as AWS EC2. By leveraging built-in Pacemaker clustering (PCS) in RHEL and Amazon FSx ONTAP, customers can achieve a viable alternative without Oracle RAC license cost for active-passive clustering on both compute and storage to support mission-critical Oracle database workload in the AWS cloud.

This documentation demonstrates the details of setting up Pacemaker clustering on RHEL, deploying Oracle database on EC2 and Amazon FSx ONTAP with NFS protocol, configuring Oracle resources in Pacemaker for HA, and wrapping up the demo with validation under most often encountered HA scenarios. The solution also provides information on fast Oracle database backup, restore, and clone with the NetApp SnapCenter UI tool.

This solution addresses the following use cases:

- Pacemaker HA clustering setup and configuration in RHEL.
- Oracle database HA deployment in AWS EC2 and Amazon FSx ONTAP.

Audience

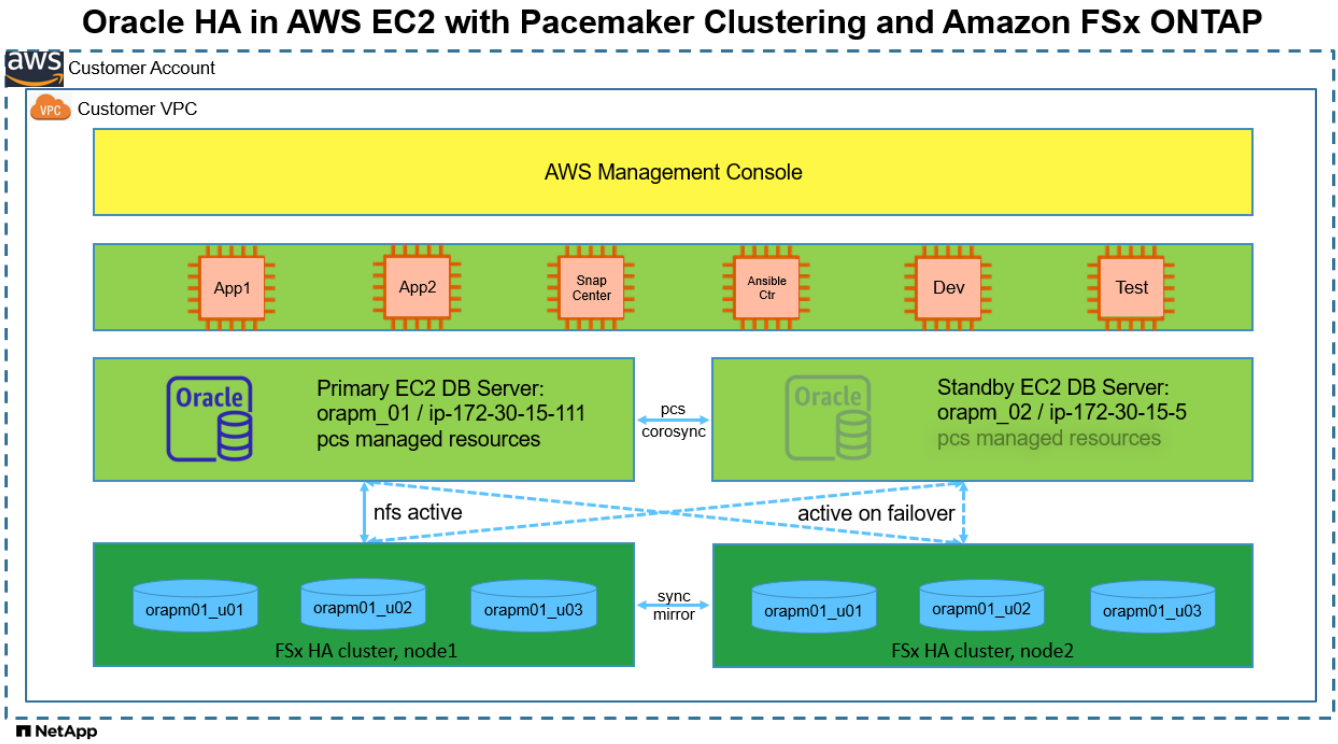
This solution is intended for the following people:

- A DBA who would like to deploy Oracle in AWS EC2 and Amazon FSx ONTAP.
- A database solution architect who would like to test Oracle workloads in AWS EC2 and Amazon FSx ONTAP.
- A storage administrator who would like to deploy and manage an Oracle database in AWS EC2 and Amazon FSx ONTAP.
- An application owner who would like to stand up an Oracle database in AWS EC2 and Amazon FSx ONTAP.

Solution test and validation environment

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section Key factors for deployment consideration for more information.

Architecture



Hardware and software components

Hardware		
Amazon FSx ONTAP storage	Current version offered by AWS	Single-AZ in us-east-1, 1024 GiB capacity, 128 MB/s throughput
EC2 instances for DB server	t2.xlarge/4vCPU/16G	Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a standby DB server
VM for Ansible controller	4 vCPUs, 16GiB RAM	One Linux VM to run automated AWS EC2/FSx provisioning and Oracle deployment on NFS
Software		
RedHat Linux	RHEL Linux 8.6 (LVM) - x64 Gen2	Deployed RedHat subscription for testing
Oracle Database	Version 19.18	Applied RU patch p34765931_190000_Linux-x86-64.zip
Oracle OPatch	Version 12.2.0.1.36	Latest patch p6880880_190000_Linux-x86-64.zip
Pacemaker	Version 0.10.18	High Availability Add-On for RHEL 8.0 by RedHat
NFS	Version 3.0	Oracle dNFS enabled

Ansible	core 2.16.2	Python 3.6.8

Oracle database active/passive configuration in the AWS EC2/FSx lab environment

Server	Database	DB Storage
primary node: orapm01/ip-172.30.15.111	NTAP(NTAP_PDB1,NTAP_PDB2, NTAP_PDB3)	/u01, /u02, /u03 NFS mounts on Amazon FSx ONTAP volumes
standby node: orapm02/ip-172.30.15.5	NTAP(NTAP_PDB1,NTAP_PDB2, NTAP_PDB3) when failover	/u01, /u02, /u03 NFS mounts when failover

Key factors for deployment consideration

- **Amazon FSx ONTAP HA.** Amazon FSx ONTAP is provisioned in an HA pair of storage controllers in single or multiple availability zones by default. It provides storage redundancy in an active/passive fashion for mission-critical database workloads. The storage failover is transparent to the end user. User intervention is not required in the event of a storage failover.
- **PCS resources group and resources ordering.** A resources group allows multiple resources with dependency to run on the same cluster node. The resource order enforces the resources startup order and the shutdown order in reverse.
- **Preferred node.** The Pacemaker cluster is purposely deployed in active/passive clustering (not a requirement by Pacemaker) and is in sync with FSx ONTAP clustering. The active EC2 instance is configured as a preferred node for Oracle resources when available with a location constraint.
- **Fence delay on standby node.** In a two-node PCS cluster, a quorum is artificially set as 1. In the event of a communication issue between the cluster nodes, either node could try to fence the other node, which can potentially cause data corruption. Setting up a delay on the standby node mitigates the issue and allows the primary node to continue providing services while the standby node is fenced.
- **Multi az deployment consideration.** The solution is deployed and validated in a single availability zone. For multi-az deployment, additional AWS networking resources are needed to move the PCS floating IP between the availability zones.
- **Oracle database storage layout.** In this solution demonstration, we provision three database volumes for the test database NTAP to host Oracle binary, data, and log. The volumes are mounted on the Oracle DB server as /u01 - binary, /u02 - data, and /u03 - log via NFS. Dual control files are configured on /u02 and /u03 mount points for redundancy.
- **dNFS configuration.** By using dNFS (available since Oracle 11g), an Oracle database running on a DB VM can drive significantly more I/O than the native NFS client. Automated Oracle deployment configures dNFS on NFSv3 by default.
- **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to achieve fast (under a minute) snapshot backup, quick (minutes) database restore, and database clone.

Solution deployment

The following sections provide step-by-step procedures for deployment and configuration of Oracle database HA in AWS EC2 with Pacemaker clustering and Amazon FSx ONTAP for database storage protection.

Prerequisites for deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.
2. Provision a Linux VM as the Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: Getting Started with NetApp solution automation^ in section -
   `Setup the Ansible Control Node for CLI deployments on RHEL / CentOS` or
   `Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian`.

   Enable ssh public/private key authentication between Ansible controller and EC2 instance DB VMs.

Provision EC2 instances and Amazon FSx ONTAP storage cluster

Although EC2 instance and Amazon FSx ONTAP can be provisioned from AWS console manually, it is recommended to use NetApp Terraform based automation toolkit to automate the provisioning of EC2 instances and FSx ONTAP storage cluster. Following are the detailed procedures.

1. From AWS CloudShell or Ansible controller VM, clone a copy of automation toolkit for EC2 and FSx ONTAP.

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

If the toolkit is not executed from AWS CloudShell, AWS CLI authentication is required with your AWS account using AWS user account access/secret key pair.

2. Review READme.md file included in the toolkit. Revise main.tf and associated parameter files as necessary for the required AWS resources.

```
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
}
```

3. Validate and execute the Terraform plan. A successful execution would create two EC2 instances and A FSx ONTAP storage cluster in target AWS account. The automation output displays the EC2 instance IP address and FSx ONTAP cluster end points.

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

```
terraform apply main.plan
```

This completes the EC2 instances and FSx ONTAP provisioning for Oracle.

Pacemaker cluster setup

The High Availability Add-On for RHEL is a clustered system that provides reliability, scalability, and availability to critical production services such as Oracle database services. In this use case demonstration, a two-node Pacemaker cluster is set up and configured to support the high availability of an Oracle database in an active/passive clustering scenario.

Login to EC2 instances, as ec2-user, complete following tasks on `both` EC2 instances:

1. Remove the AWS Red Hat Update Infrastructure (RHUI) client.

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

2. Register the EC2 instance VMs with Red Hat.

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

3. Enable RHEL high availability rpms.

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

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

4. Install pacemaker and fence agant.

```
sudo yum update -y
```

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

5. Create a password for hacluster user on all cluster nodes. Use the same password for all nodes.

```
sudo passwd hacluster
```

6. Start the pcs service and enable it to start on boot.

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

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

7. Validate pcsd service.

```
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. Add cluster nodes to host files.

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

```
[ec2-user@ip-172-30-15-5 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6

# 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. Install and configure awscli for connectivity to AWS account.

```
sudo yum install awscli
```

```
sudo aws configure
```

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

10. Install the resource-agents package if not installed already.

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

On `only one` of the cluster node, complete following tasks to create pcs cluster.

1. Authenticate the pcs user 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. Create the pcs 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. Enable the 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. Start and validate the 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
```

This completes the Pacemaker cluster setup and initial configuration.

Pacemaker cluster fencing configuration

Pacemaker fencing configuration is mandatory for a production cluster. It ensures that a malfunctioning node on your AWS EC2 cluster is automatically isolated, thus preventing the node from consuming the cluster's resources, compromising the cluster's functionality, or corrupting shared data. This section demonstrates the configuration of cluster fencing using the fence_aws fencing agent.

1. As root user, enter the following AWS metadata query to get the Instance ID for each EC2 instance node.

```
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. Enter the following command to configure the fence device. Use the pcmk_host_map command to map the RHEL host name to the Instance ID. Use the AWS Access Key and the AWS Secret Access Key of the AWS user account that you previously used for AWS authentication.

```
sudo pcs stonith \
create clusterfence fence_aws access_key=XXXXXXXXXXXXXXXXX
secret_key=XXXXXXXXXXXXXXXXXX \
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. Validate the fencing configuration.

```
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. Set stonith-action to off instead of reboot at the cluster level.

```
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
```

With stonith-action set to off, the fenced cluster node will initially be shutdown. After the period defined in stonith power_timeout (240 seconds), the fenced node will be rebooted and rejoins the cluster.

5. Set fence delay to 10 seconds for standby node.

```
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=XXXXXXXXXXXXXXXX
    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=XXXXXXXXXXXXXXXX
  Operations:
    monitor: clusterfence-monitor-interval-60s
      interval=60s
```

Execute `pcs stonith refresh` command to refresh stopped stonith fence agent or clear failed stonith resource actions.

Deploy Oracle database in PCS cluster

We recommend leveraging the NetApp-provided Ansible playbook to execute database installation and configuration tasks with predefined parameters on the PCS cluster. For this automated Oracle deployment, three user-defined parameter files need user input before playbook execution.

- hosts - define targets that the automation playbook is running against.
- vars/vars.yml - the global variable file that defines variables that apply to all targets.
- host_vars/host_name.yml - the local variable file that defines variables that apply only to a named target. In our use case, these are the Oracle DB servers.

In addition to these user-defined variable files, there are several default variable files that contain default parameters that do not require change unless necessary. The following shows the details of automated Oracle deployment in AWS EC2 and FSx ONTAP in a PCS clustering configuration.

1. From Ansible controller admin user home directory, clone a copy of the NetApp Oracle deployment automation toolkit for NFS.

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

The Ansible controller can be located in the same VPC as the database EC2 instance or on-premises as long as there is network connectivity between them.

2. Fill in the user defined parameters in hosts parameter files. Following are example of typical host file configuration.

```
[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. Fill in the user defined parameters in vars/vars.yml parameter files. Following are example of typical vars.yml file configuration.

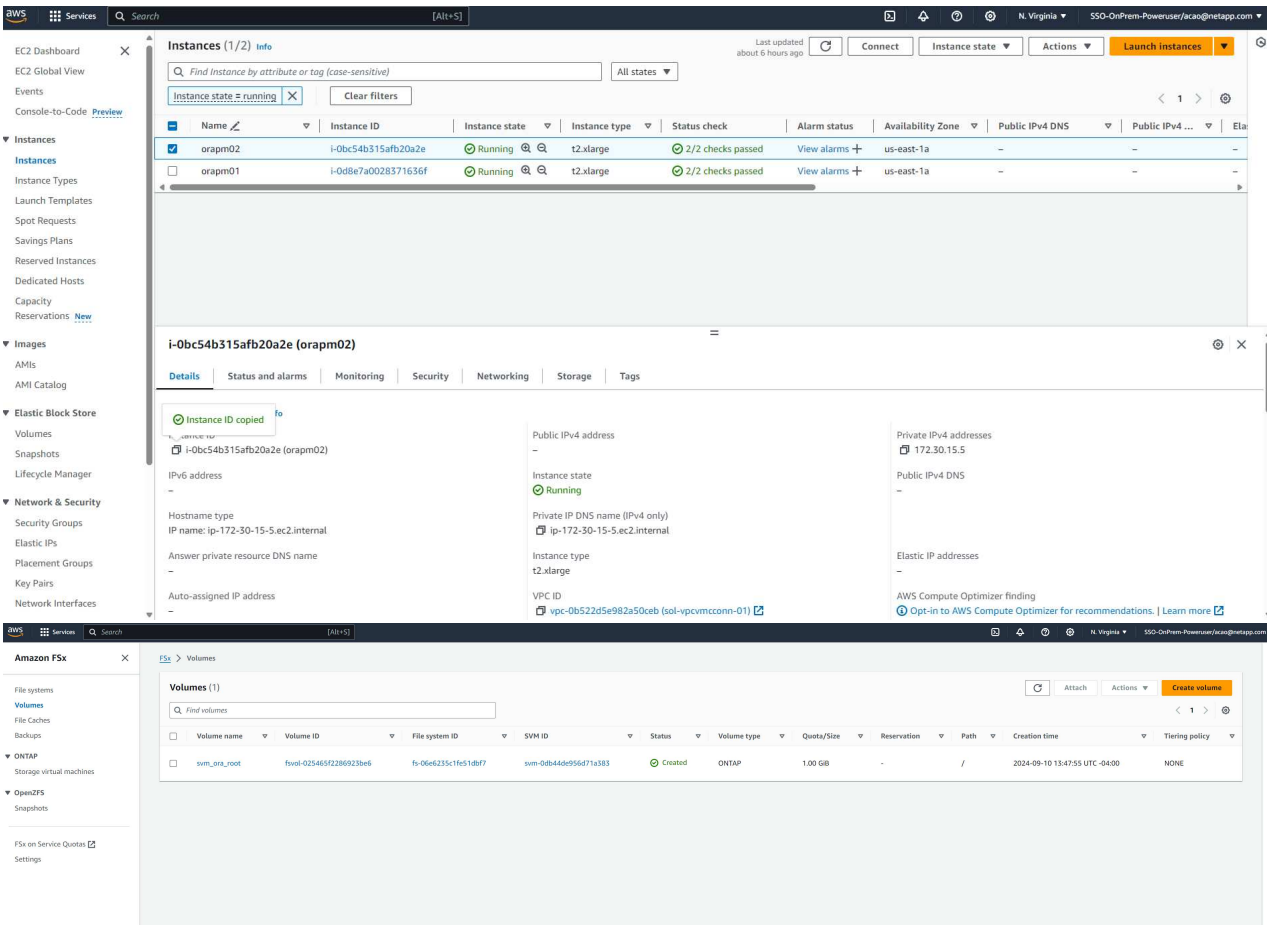
```
[admin@ansiblectl na_oracle_deploy_nfs]$ cat vars/vars.yml
#################################################################################
##
###### Oracle 19c deployment user configuration variables
######
###### Consolidate all variables from ONTAP, linux and oracle
######
#################################################################################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

############################################
### 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: "xxxxxxxx"


#####################################################
### 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: "xxxxxxxx"
```

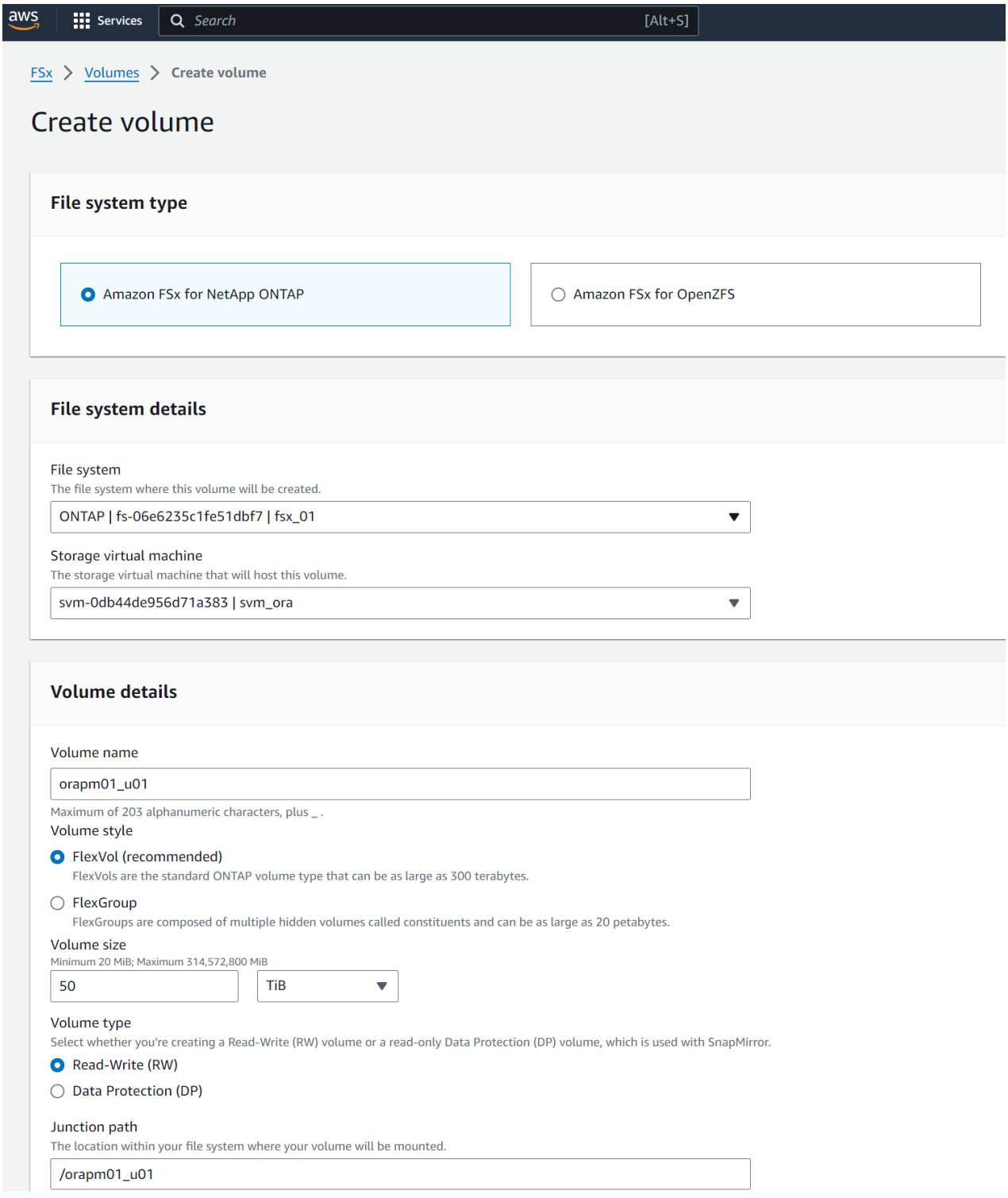
4. Fill in the user defined parameters in host_vars/host_name.yml parameter files. Following are example of typical host_vars/host_name.yml file configuration.

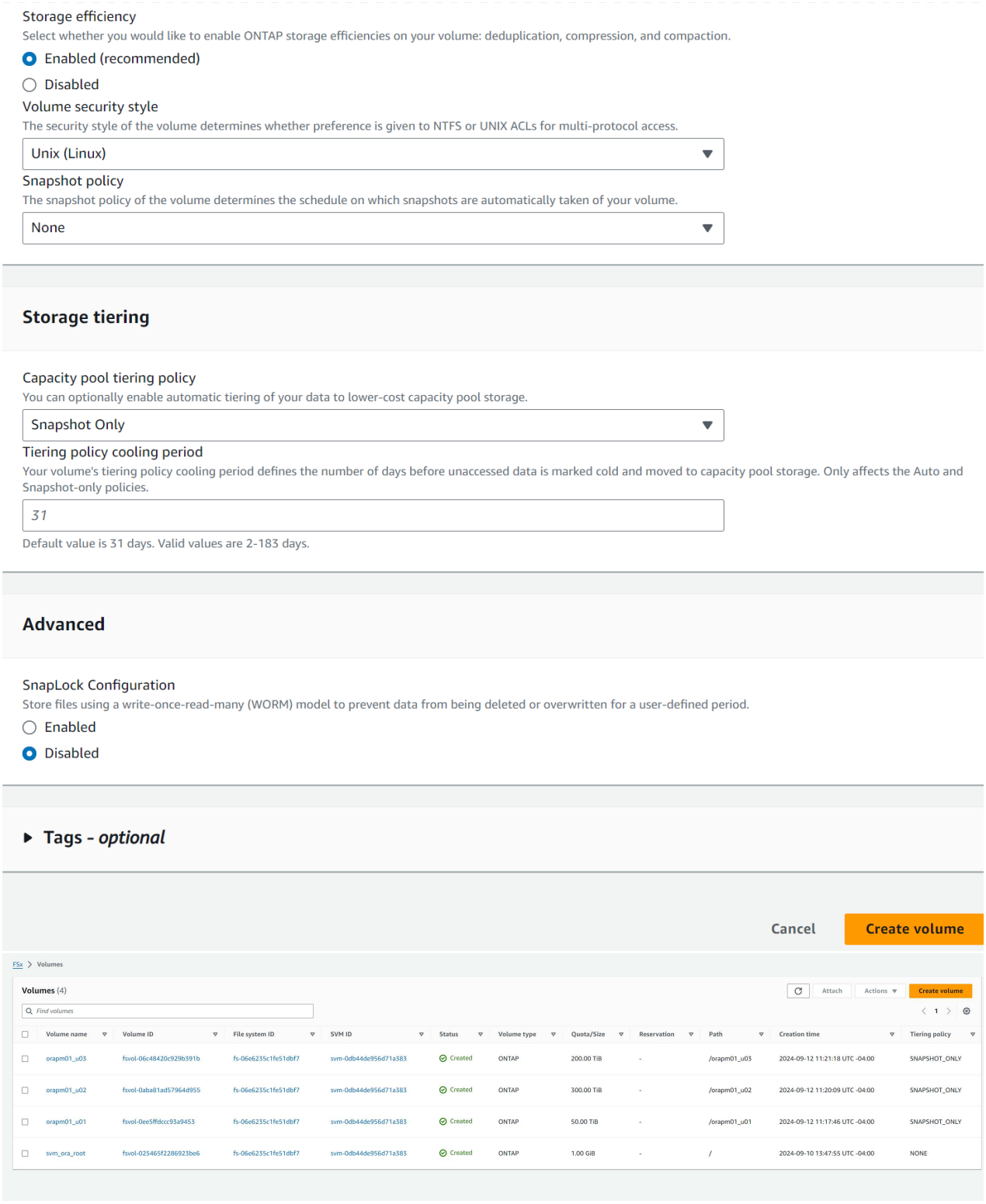
```
[admin@ansiblectl na_oracle_deploy_nfs]$ cat host_vars/orapm01.yml
# User configurable Oracle host specific parameters

# 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
```

nfs_lif address can be retrieved from FSx ONTAP cluster end points output from automated EC2 and FSx ONTAP deployment in previous section.

5. Create database volumes from AWS FSx console. Ensure to use PCS primary node host name (orapm01) as prefix for the volumes as demonstrated below.




aws
Services
Search
[Alt+S]
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



6. Stage following Oracle 19c installation files on PCS primary node EC2 instance ip-172-30-15-111.ec2.internal /tmp/archive directory with 777 permission.

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

7. Execute playbook for Linux config for `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. Execute playbook for oracle config `only on primary node` (comment out standby node in hosts file).

```
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. After database is deployed, comment out /u01, /u02, /u03 mounts in /etc/fstab on primary node since the mount points will be managed by PCS only.

```
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,rsize=65536,wsize=65536 0 0
#172.30.15.95:/orapm01_u02 /u02 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536 0 0
#172.30.15.95:/orapm01_u03 /u03 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536 0 0
```

10. Copy /etc/oratab /etc/oraInst.loc, /home/oracle/.bash_profile to standby node. Ensure to maintain proper file ownership and permissions.
11. Shutdown database, listener, and umount /u01, /u02, /u03 on primary node.

```
[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. Create mount points on standby node ip-172-30-15-5.

```
mkdir /u01
mkdir /u02
mkdir /u03
```

13. Mount the FSx ONTAP database volumes on standby node 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,rsize=65536,wsize=65536
```

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

```
mount -t nfs 172.30.15.95:/orapm01_u03 /u03 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=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. Changed to oracle user, relink binary.

```
[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. Copy dnfs lib back to odm folder. Relink could lose the dfns library file.

```
[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. Start database to validate on standby node 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. Shutdown db and failback database to primary node ip-172-30-15-111.

```
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,rsize=65536,wsize=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,rsize=65536,wsize=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,rsize=65536,wsize=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/tnslsnr/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
```

Configure Oracle resources for PCS management

The goal of configuring Pacemaker clustering is to set up an active/passive high-availability solution for running Oracle in AWS EC2 and FSx ONTAP environment with minimal user intervention in the event of a failure. The following demonstrates Oracle resources configuration for PCS management.

1. As root user on primary EC2 instance ip-172-30-15-111, create a secondary private IP address with an unused private IP address in the VPC CIDR block as floating IP. In the process, create an oracle resource group that the secondary private IP address will belong to.

```
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
```

If the privip happens to be created on standby cluster node, move it to primary node as shown below.

2. Move a resource between cluster nodes.

```
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. Create a virtual IP (vip) for Oracle. Virtual IP will float between primary and standby node as needed.

```
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. As oracle user, update listener.ora and tnsnames.ora file to point to vip address. Restart the listener. Bounce database if needed for DB to register with listener.

```
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/xdb_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. Add /u01, /u02, /u03 mount points to oracle resource group.

```
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,rsize=65536,wsize=655
36' --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,rsize=65536,wsize=655
36' --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,rsize=65536,wsize=655
36' --group oracle
```

6. Create a PCS monitor user 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 "XXXXXXXX";

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. Add database to oracle resource group.

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

8. Add database listener to oracle resource group.

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

9. Update all resources location constraints in oracle resource group to primary node as preferred node.

```
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. Validate Oracle resources configuration.

```
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
```

Post deployment HA validation

After the deployment, it is vital to run some testing and validation to ensure that the PCS Oracle database failover cluster is configured correctly and functions as expected. The test validation includes managed failover and simulated unexpected resource failure and recovery by the cluster protection mechanism.

1. Validate node fencing by manually triggering the fencing of standby node and observe that standby node was brought offline and rebooted after a timeout.

```
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. Simulate an database listener failure by killing listener process and observe that PCS monitored the

listener failure and restarted it in a few seconds.

```
[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/xdb_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. Simulate an database failure by killing the pmon process and observe that PCS monitored the databadse failure and restarted it in a few seconds.

```
**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. Validate a managed database failover from primary to standby by putting primary node on standby-mode to failover Oracle resources to standby node.

```
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 sequencial 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. Validate a managed database failback from standby to primary by unstandby primary node and observe that Oracle resources failback automatically due to prefered node setting.

```
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>
```

This completes the Oracle HA validation and solution demonstration in AWS EC2 with Pacemaker clustering and Amazon FSx ONTAP as database storage backend.

Oracle backup, restore, and clone with SnapCenter

NetApp recommends SnapCenter UI tool to manage Oracle database deployed in AWS EC2 and Amazon FSx ONTAP. Refer to TR-4979 Simplified, Self-managed Oracle in VMware Cloud on AWS with guest-mounted FSx ONTAP section `Oracle backup, restore, and clone with SnapCenter` for details on setting up SnapCenter and executing the database backup, restore, and clone workflows.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- Configuring and managing high availability clusters
- Amazon FSx ONTAP
- Deploying Oracle Direct NFS

Copyright information

Trademark information