NetApp

Oracle Database

NetApp Solutions

NetApp
August 25, 2022

Table of Contents

Oracle Database

Deploying Oracle Database

Solution Overview

Automated Deployment of Oracle19c for ONTAP on NFS

Organizations are automating their environments to gain efficiencies, accelerate deployments, and reduce manual effort. Configuration management tools like Ansible are being used to streamline enterprise database operations. In this solution, we demonstrate how you can use Ansible to automate the provisioning and configuration of Oracle 19c with NetApp ONTAP. By enabling storage administrators, systems administrators, and DBAs to consistently and rapidly deploy new storage, configure database servers, and install Oracle 19c software, you achieve the following benefits:

- Eliminate design complexities and human errors, and implement a repeatable consistent deployment and best practices

- Decrease time for provisioning of storage, configuration of DB hosts, and Oracle installation

- Increase database administrators, systems and storage administrators productivity

- Enable scaling of storage and databases with ease

NetApp provides customers with validated Ansible modules and roles to accelerate deployment, configuration, and lifecycle management of your Oracle database environment. This solution provides instruction and Ansible playbook code, to help you:

- Create and configure ONTAP NFS storage for Oracle Database

- Install Oracle 19c on RedHat Enterprise Linux 7/8 or Oracle Linux 7/8

- Configure Oracle 19c on ONTAP NFS storage

For more details or to begin, please see the overview videos below.

AWX/Tower Deployments

- Part 1: Getting Started, Requirements, Automation Details and Initial AWX/Tower Configuration

▶ https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v1.mp4 *(video)*

- Part 2: Variables and Running the Playbook

▶ https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v2.mp4 *(video)*

CLI Deployment

- Part 1: Getting Started, Requirements, Automation Details and Ansible Control Host Setup

▶ https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v4.mp4 *(video)*

- Part 2: Variables and Running the Playbook

▶ https://docs.netapp.com/us-en/netapp-solutions/media/oracle3.mp4 *(video)*

Getting started

This solution has been designed to be run in an AWX/Tower environment or by CLI on an Ansible control host.

AWX/Tower

For AWX/Tower environments, you are guided through creating an inventory of your ONTAP cluster management and Oracle server (IPs and hostnames), creating credentials, configuring a project that pulls the Ansible code from NetApp Automation Github, and the Job Template that launches the automation.

1. Fill out the variables specific to your environment, and copy and paste them into the Extra Vars fields in your job template.

2. After the extra vars have been added to your job template, you can launch the automation.

3. The job template is run in three phases by specifying tags for ontap_config, linux_config, and oracle_config.

CLI via the Ansible control host

1. To configure the Linux host so that is can be used as an Ansible control host
   click here for RHEL 7/8 or CentOS 7/8, or
   here for Ubuntu/Debian

2. After the Ansible control host is configured, you can git clone the Ansible Automation repository.

3. Edit the hosts file with the IPs and/or hostnames of your ONTAP cluster management and Oracle server's management IPs.

4. Fill out the variables specific to your environment, and copy and paste them into the `vars.yml` file.

5. Each Oracle host has a variable file identified by its hostname that contains host-specific variables.

6. After all variable files have been completed, you can run the playbook in three phases by specifying tags for `ontap_config`, `linux_config`, and `oracle_config`.

Requirements

Environment	Requirements
Ansible environment	AWX/Tower or Linux host to be the Ansible control host
	Ansible v.2.10 and higher
	Python 3
	Python libraries - netapp-lib - xmltodict - jmespath
ONTAP	ONTAP version 9.3 - 9.7
	Two data aggregates
	NFS vlan and ifgrp created

Environment	Requirements
Oracle server(s)	RHEL 7/8
	Oracle Linux 7/8
	Network interfaces for NFS, public, and optional mgmt
	Oracle installation files on Oracle servers

Automation Details

This automated deployment is designed with a single Ansible playbook that consists of three separate roles.
The roles are for ONTAP, Linux, and Oracle configurations.
The following table describes which tasks are being automated.

Role	Tasks
ontap_config	Pre-check of the ONTAP environment
	Creation of NFS based SVM for Oracle
	Creation of export policy
	Creation of volumes for Oracle
	Creation of NFS LIFs
linux_config	Create mount points and mount NFS volumes
	Verify NFS mounts
	OS specific configuration
	Create Oracle directories
	Configure hugepages
	Disable SELinux and firewall daemon
	Enable and start chronyd service
	increase file descriptor hard limit
	Create pam.d session file
oracle_config	Oracle software installation
	Create Oracle listener
	Create Oracle databases
	Oracle environment configuration
	Save PDB state
	Enable instance archive mode
	Enable DNFS client
	Enable database auto startup and shutdown between OS reboots

Default parameters

To simplify automation, we have preset many required Oracle deployment parameters with default values. It is generally not necessary to change the default parameters for most deployments. A more advanced user can make changes to the default parameters with caution. The default parameters are located in each role folder under defaults directory.

Deployment instructions

Before starting, download the following Oracle installation and patch files and place them in the `/tmp/archive` directory with read, write, and execute access for all users on each DB server to be deployed. The automation tasks look for the named installation files in that particular directory for Oracle installation and configuration.

```
LINUX.X64_193000_db_home.zip -- 19.3 base installer
p31281355_190000_Linux-x86-64.zip -- 19.8 RU patch
p6880880_190000_Linux-x86-64.zip -- opatch version 12.2.0.1.23
```

License

You should read license information as stated in the Github repository. By accessing, downloading, installing, or using the content in this repository, you agree the terms of the license laid out here.

Note that there are certain restrictions around producing and/or sharing any derivative works with the content in this repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download, or use the content in this repository.

After you are ready, click here for detailed AWX/Tower deployment procedures or here for CLI deployment.

Step-by-step deployment procedure

AWX/Tower deployment Oracle 19c Database

1. Create the inventory, group, hosts, and credentials for your environment

This section describes the setup of inventory, groups, hosts, and access credentials in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.

    a. Navigate to Resources → Inventories → Add, and click Add Inventory.

    b. Provide the name and organization details, and click Save.

    c. On the Inventories page, click the inventory created.

    d. If there are any inventory variables, paste them in the variables field.

    e. Navigate to the Groups sub-menu and click Add.

    f. Provide the name of the group for ONTAP, paste the group variables (if any) and click Save.

    g. Repeat the process for another group for Oracle.

    h. Select the ONTAP group created, go to the Hosts sub-menu and click Add New Host.

    i. Provide the IP address of the ONTAP cluster management IP, paste the host variables (if any), and click Save.

      j. This process must be repeated for the Oracle group and Oracle host(s) management IP/hostname.

2. Create credential types. For solutions involving ONTAP, you must configure the credential type to match username and password entries.

    a. Navigate to Administration → Credential Types, and click Add.

    b. Provide the name and description.

    c. Paste the following content in Input Configuration:

```
fields:
  - id: username
    type: string
    label: Username
  - id: password
    type: string
    label: Password
    secret: true
  - id: vsadmin_password
    type: string
    label: vsadmin_password
    secret: true
```

    d. Paste the following content into Injector Configuration:

```
extra_vars:
  password: '{{ password }}'
  username: '{{ username }}'
  vsadmin_password: '{{ vsadmin_password }}'
```

3. Configure the credentials.

    a. Navigate to Resources → Credentials, and click Add.

    b. Enter the name and organization details for ONTAP.

    c. Select the custom Credential Type you created for ONTAP.

    d. Under Type Details, enter the username, password, and vsadmin_password.

    e. Click Back to Credential and click Add.

    f. Enter the name and organization details for Oracle.

    g. Select the Machine credential type.

    h. Under Type Details, enter the Username and Password for the Oracle hosts.

    i. Select the correct Privilege Escalation Method, and enter the username and password.

2. Create a project

1. Go to Resources → Projects, and click Add.

    a. Enter the name and organization details.

b. Select Git in the Source Control Credential Type field.

c. enter https://github.com/NetApp-Automation/na_oracle19c_deploy.git as the source control URL.

d. Click Save.

e. The project might need to sync occasionally when the source code changes.

3. Configure Oracle host_vars

The variables defined in this section are applied to each individual Oracle server and database.

1. Input your environment-specific parameters in the following embedded Oracle hosts variables or host_vars form.

(i) The items in blue must be changed to match your environment.

Host VARS Config

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_datastores_nfs {
  display: block;
}
#more_datastores_nfs_button {
  display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button1" onclick="CopyClassText1()">Copy</button></div><pre><code><div
class="CopyMeClass1" id="CopyMeID1">
###############################################################
```

```
#############      Host Variables Configuration      #############
################################################################

# Add your Oracle Host
ansible_host: 10.61.180.15

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: true

# Number of pluggable databases per container instance identified by sid.
Pdb_name specifies the prefix for container database naming in this case
cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: cdb2
pdb_num: 3
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: 1523

# 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 SGA should not exceed 75% available RAM on node.
memory_limit: 5464

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them black if em express is not installed.
em_configuration: DBEXPRESS
em_express_port:
```

```
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>5501</i></span>

# "{{groups.oracle[0]}}" represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example,  "{{groups.oracle[1]}}" represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u01</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u02</i></span>&quot;,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u03</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
<a id="more_datastores_nfs" href="javascript:datastoredropdown();">More
NFS volumes</a><div id="select_more_datastores_nfs"></div><a
id="more_datastores_nfs_button"
```

```
href="javascript:adddatastorevolumes();">Enter NFS volumes'
details</a><div id="extra_datastores_nfs"></div>
</div></code></pre></div></div>
<script>
function CopyClassText1(){
    var textToCopy = document.getElementById("CopyMeID1");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_datastores_nfs").style.display = "none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button1").innerHTML = "Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
      document.getElementById("copy-button1").innerHTML = "Copy";
      document.getElementById("more_datastores_nfs").style.display =
"block";
}

function datastoredropdown() {
    document.getElementById("more_datastores_nfs").style.display = "none";
    document.getElementById("more_datastores_nfs_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_datastores_nfs");
```

```
    while (x < 100) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_datastores_nfs">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_datastores_nfs"
id="number_of_extra_datastores_nfs">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatastorevolumes() {
    var y =
document.getElementById("number_of_extra_datastores_nfs").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_datastores_nfs");
    while (j < y) {
        j++;
        myHTML += '  - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>"{{groups.oracle[0]}}_u01"</i></span>,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node02</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.201</i></span>,
size: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_datastores_nfs").style.display =
"none";
    document.getElementById("more_datastores_nfs_button").style.display =
"none";
}

</script>
```

a. Fill in all variables in the blue fields.

b. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower.

c. Navigate back to AWX or Tower and go to Resources → Hosts, and select and open the Oracle server configuration page.

d. Under the Details tab, click edit and paste the copied variables from step 1 to the Variables field under the YAML tab.

e. Click Save.

f. Repeat this process for any additional Oracle servers in the system.

4. Configure global variables

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.

> ℹ️ The items in blue must be changed to match your environment.

VARS

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
   transition-duration: 0.4s;
   background-color: white;
   color: #1563a3;
   border: 2px solid #1563a3;
}
button:hover {
   background-color: #1563a3;
   color: white;
}
#more_storage_vlans {
   display: block;
}
#more_storage_vlans_button {
   display: none;
}
#more_nfs_volumes {
   display: block;
}
#more_nfs_volumes_button {
   display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
```

```
button" onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyMeID">
####################################################################
###### Oracle 19c deployment global user configuration variables ######
######  Consolidate all variables from ontap, linux and oracle   ######
####################################################################


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


#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in inventory/hosts
file or in inventory groups in case of AWX/Tower
hosts_group: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>ontap</i></span>


#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>false</i></span>


#Names of the Nodes in the ONTAP Cluster
nodes:
 - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
01</i></span>
 - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
02</i></span>



#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
   - {vlan_id: &quot;<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>203</i></span>&quot;, name: &quot;<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>infra_NFS</i></span>&quot;,
protocol: &quot;<span <div
contenteditable="true"/><i>NFS</i></span>&quot;}
<a id="more_storage_vlans" href="javascript:storagevlandropdown();">More
```

Storage VLANs<div id="select_more_storage_vlans"></div><a
id="more_storage_vlans_button" href="javascript:addstoragevlans();">Enter
Storage VLANs details<div id="extra_storage_vlans"></div>

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i>}
  - {aggr_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node02</i>}

#SVM name
svm_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>ora_svm</i>

SVM Management LIF Details
svm_mgmt_details:
  - {address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.91.100</i>, netmask: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>255.255.255.0</i>,
home_port: <span <div contenteditable="true"/><i>e0M</i>}

NFS storage parameters when data_protocol set to NFS. Volume named after
Oracle hosts name identified by mount point as follow for oracle DB server
1. Each mount point dedicates to a particular Oracle files: u01 - Oracle
binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by
click on "More NFS volumes" and also add the volumes list to corresponding
host_vars as host_datastores_nfs variable. For multiple DB server
deployment, additional volumes sets needs to be added for additional DB
server. Input variable "{{groups.oracle[1]}}_u01",
"{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for
second DB server. Place volumes for multiple DB servers alternatingly
between controllers for balanced IO performance, e.g. DB server 1 on
controller node1, DB server 2 on controller node2 etc. Make sure match lif
address with controller node.
volumes_nfs:

```
    - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u01</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
    - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u02</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
    - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u03</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
<a id="more_nfs_volumes" href="javascript:nfsvolumesdropdown();">More NFS
volumes</a><div id="select_more_nfs_volumes"></div><a
id="more_nfs_volumes_button" href="javascript:addnfsvolumes();">Enter NFS
volumes' details</a><div id="extra_nfs_volumes"></div>

#NFS LIFs IP address and netmask
nfs_lifs_details:
  - address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.200</i></span> #for node-1
    netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>
  - address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.201</i></span> #for node-2
    netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>
```

```
#NFS client match
client_match: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.0/24</i></span>


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

#NFS Mount points for Oracle DB volumes
mount_points:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03</i></span>

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>

#################################################
### DB env specific install and config variables ###
#################################################

db_domain: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>your.domain.com</i></span>
```

```
# Set initial password for all required Oracle passwords. Change them
after installation.
initial_pwd_all: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>netapp123</i></span>

</div></code></pre></div></div>
<script>
function CopyClassText(){
    var textToCopy = document.getElementById("CopyMeID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_nfs_volumes").style.display = "none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button").innerHTML = "Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
      document.getElementById("copy-button").innerHTML = "Copy";
      document.getElementById("more_storage_vlans").style.display =
"block";
      document.getElementById("more_nfs_volumes").style.display = "block";
}
function storagevlandropdown() {
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_storage_vlans_button").style.display =
```

```
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_storage_vlans");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_storage_vlans_info">How many extra VLANs do you
wish to add?</a><select name="number_of_extra_storage_vlans"
id="number_of_extra_storage_vlans">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addstoragevlans() {
    var y =
document.getElementById("number_of_extra_storage_vlans").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_storage_vlans");
    while (j < y) {
        j++;
        myHTML += '   - {vlan_id: &quot;<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>203</i></span>&quot;, name: &quot;<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>infra_NFS</i></span>&quot;,
protocol: &quot;<span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>NFS</i></span>&quot;}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_storage_vlans").style.display =
"none";
    document.getElementById("more_storage_vlans_button").style.display =
"none";
}
function nfsvolumesdropdown() {
    document.getElementById("more_nfs_volumes").style.display = "none";
    document.getElementById("more_nfs_volumes_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_nfs_volumes");
    while (x < 100) {
```

```
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_nfs_volumes_info">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_nfs_volumes"
id="number_of_extra_nfs_volumes">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addnfsvolumes() {
    var y = document.getElementById("number_of_extra_nfs_volumes").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_nfs_volumes");
    while (j < y) {
        j++;
        myHTML += '  - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>rtpora04_u01</i></span>, aggr_name: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>aggr01_node02</i></span>,
lif: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.201</i></span>, size: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_nfs_volumes").style.display =
"none";
    document.getElementById("more_nfs_volumes_button").style.display =
"none";
}

</script>
```

1. Fill in all variables in blue fields.

2. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower into the following job template.

5. Configure and launch the job template.

1. Create the job template.

    a. Navigate to Resources → Templates → Add and click Add Job Template.

    b. Enter the name and description

    c. Select the Job type; Run configures the system based on a playbook, and Check performs a dry run of a playbook without actually configuring the system.

d. Select the corresponding inventory, project, playbook, and credentials for the playbook.

e. Select the all_playbook.yml as the default playbook to be executed.

f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.

g. Check the box Prompt on Launch in the Job Tags field.

h. Click Save.

2. Launch the job template.

a. Navigate to Resources → Templates.

b. Click the desired template and then click Launch.

c. When prompted on launch for Job Tags, type in requirements_config. You might need to click the Create Job Tag line below requirements_config to enter the job tag.

> (i)　requirements_config ensures that you have the correct libraries to run the other roles.

d. Click Next and then Launch to start the job.

e. Click View → Jobs to monitor the job output and progress.

f. When prompted on launch for Job Tags, type in ontap_config. You might need to click the Create "Job Tag" line right below ontap_config to enter the job tag.

g. Click Next and then Launch to start the job.

h. Click View → Jobs to monitor the job output and progress

i. After the ontap_config role has completed, run the process again for linux_config.

j. Navigate to Resources → Templates.

k. Select the desired template and then click Launch.

l. When prompted on launch for the Job Tags type in linux_config, you might need to select the Create "job tag" line right below linux_config to enter the job tag.

m. Click Next and then Launch to start the job.

n. Select View → Jobs to monitor the job output and progress.

o. After the linux_config role has completed, run the process again for oracle_config.

p. Go to Resources → Templates.

q. Select the desired template and then click Launch.

r. When prompted on launch for Job Tags, type oracle_config. You might need to select the Create "Job Tag" line right below oracle_config to enter the job tag.

s. Click Next and then Launch to start the job.

t. Select View → Jobs to monitor the job output and progress.

6. Deploy additional database on same Oracle host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container databases on the same server, complete the following steps.

1. Revise host_vars variables.

a. Go back to step 2 - Configure Oracle host_vars.

b. Change the Oracle SID to a different naming string.

    c. Change the listener port to different number.

    d. Change the EM Express port to a different number if you are installing EM Express.

    e. Copy and paste the revised host variables to the Oracle Host Variables field in the Host Configuration Detail tab.

2. Launch the deployment job template with only the oracle_config tag.

Validate Oracle installation

```
1. Log in to Oracle server as oracle user and execute the following
commands:
```

```
ps -ef | grep ora
```

ⓘ   This will list oracle processes if installation completed as expected and oracle DB started

```
2. Log in to the database to check the db configuration settings and the
PDBs created with the following command sets.
```

```
sqlplus / as sysdba
```

[oracle@localhost ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

```
select name, log_mode from v$database;
```

SQL> select name, log_mode from v$database;
NAME LOG_MODE
--------- ------------
CDB2 ARCHIVELOG

```
show pdbs;
```

SQL> show pdbs

```
    CON_ID CON_NAME                         OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                         READ ONLY  NO
         3 CDB2_PDB1                        READ WRITE NO
         4 CDB2_PDB2                        READ WRITE NO
         5 CDB2_PDB3                        READ WRITE NO
```

```
col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;
```

SQL> col svrname form a30
SQL> col dirname form a30
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME DIRNAME NFSVERSION
---------------------------- ---------------------------- ---------------
172.21.126.200 /rhelora03_u02 NFSv3.0
172.21.126.200 /rhelora03_u03 NFSv3.0
172.21.126.200 /rhelora03_u01 NFSv3.0

```
This confirms that dNFS is working properly.
```

```
3. Connect to database via listener to check hte Oracle listener
configuration with the following command. Change to the appropriate
listener port and database service name.
```

```
sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com
```

[oracle@localhost ~]$ sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 13:19:57 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed May 05 2021 17:11:11 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

```
SQL> show user
USER is "SYSTEM"
SQL> show con_name
CON_NAME
CDB2_PDB1
```

```
  This confirms that Oracle listener is working properly.
```

Where to go for help?

If you need help with the toolkit, please join the NetApp Solution Automation community support slack channel and look for the solution-automation channel to post your questions or inquires.

Step-by-step deployment procedure

CLI deployment Oracle 19c Database

This section covers the steps required to prepare and deploy Oracle19c Database with the CLI. Make sure that you have reviewed the Getting Started and Requirements section and prepared your environment accordingly.

Download Oracle19c repo

```
  1. From your ansible controller, run the following command:
```

```
  git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

```
  2. After downloading the repository, change directories to
  na_oracle19c_deploy <cd na_oracle19c_deploy>.
```

Edit the hosts file

Complete the following before deployment:

1. Edit your hosts file na_oracle19c_deploy directory.
2. Under [ontap], change the IP address to your cluster management IP.
3. Under the [oracle] group, add the oracle hosts names. The host name must be resolved to its IP address either through DNS or the hosts file, or it must be specified in the host.
4. After you have completed these steps, save any changes.

The following example depicts a host file:

```
#ONTAP Host<div>
[ontap]
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>10.61.184.183<i></i></span>
</div>
#Oracle hosts<div>
<div>
[oracle]<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora01<i></i></span>
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora02<i></i></span>
</div>
```

This example executes the playbook and deploys oracle 19c on two oracle DB servers concurrently. You can also test with just one DB server. In that case, you only need to configure one host variable file.

ⓘ The playbook executes the same way regardless of how many Oracle hosts and databases you deploy.

Edit the host_name.yml file under host_vars

Each Oracle host has its host variable file identified by its host name that contains host-specific variables. You can specify any name for your host. Edit and copy the `host_vars` from the Host VARS Config section and paste it into your desired `host_name.yml` file.

ⓘ The items in blue must be changed to match your environment.

Host VARS Config

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
```

```css
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_datastores_nfs {
  display: block;
}
#more_datastores_nfs_button {
  display: none;
}
</style>
```

<div class="listingblock"><div class="content"><div><button id="copy-button1" onclick="CopyClassText1()">Copy</button></div><pre><code><div class="CopyMeClass1" id="CopyMeID1">

```
######################################################################
##############      Host Variables Configuration      #############
######################################################################

# Add your Oracle Host
ansible_host: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>10.61.180.15</i></span>

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>true</i></span>

# Number of pluggable databases per container instance identified by sid.
Pdb_name specifies the prefix for container database naming in this case
cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>cdb2</i></span>
pdb_num: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>3</i></span>
pdb_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>"{{ oracle_sid }}_pdb"</i></span>
```

```
# CDB listener port, use different listener port for additional CDB on
same host
listener_port: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>1523</i></span>

# 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 SGA should not exceed 75% available RAM on node.
memory_limit: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>5464</i></span>

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them black if em express is not installed.
em_configuration: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>DBEXPRESS</i></span>
em_express_port: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>5501</i></span>

# "{{groups.oracle[0]}}" represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example,  "{{groups.oracle[1]}}" represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u01</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true"
```

```
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u02</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>{{groups.oracle[0]}}_u03</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
<a id="more_datastores_nfs" href="javascript:datastoredropdown();">More
NFS volumes</a><div id="select_more_datastores_nfs"></div><a
id="more_datastores_nfs_button"
href="javascript:adddatastorevolumes();">Enter NFS volumes'
details</a><div id="extra_datastores_nfs"></div>
</div></code></pre></div></div>
<script>
function CopyClassText1(){
    var textToCopy = document.getElementById("CopyMeID1");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_datastores_nfs").style.display = "none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button1").innerHTML = "Copied!";
          setTimeout(revert_copy, 3000);
      }
```

```
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
     document.getElementById("copy-button1").innerHTML = "Copy";
     document.getElementById("more_datastores_nfs").style.display =
"block";
}

function datastoredropdown() {
    document.getElementById("more_datastores_nfs").style.display = "none";
    document.getElementById("more_datastores_nfs_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_datastores_nfs");
    while (x < 100) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_datastores_nfs">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_datastores_nfs"
id="number_of_extra_datastores_nfs">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatastorevolumes() {
    var y =
document.getElementById("number_of_extra_datastores_nfs").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_datastores_nfs");
    while (j < y) {
        j++;
        myHTML += '  - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>"{{groups.oracle[0]}}_u01"</i></span>,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node02</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>172.21.94.201</i></span>,
size: <span <div contenteditable="true" style="color:#004EFF; font-
```

```
weight:bold; font-style:italic; text-
decoration:underline;"/><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_datastores_nfs").style.display =
"none";
    document.getElementById("more_datastores_nfs_button").style.display =
"none";
}


</script>
```

Edit the vars.yml file

The `vars.yml` file consolidates all environment-specific variables (ONTAP, Linux, or Oracle) for Oracle deployment.

- Edit and copy the variables from the VARS section and paste these variables into your `vars.yml` file.

VARS

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_storage_vlans {
  display: block;
}
#more_storage_vlans_button {
  display: none;
}
```

```
#more_nfs_volumes {
  display: block;
}
#more_nfs_volumes_button {
  display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button" onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyMeID">
######################################################################
###### Oracle 19c deployment global user configuration variables ######
######  Consolidate all variables from ontap, linux and oracle   ######
######################################################################


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


#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in inventory/hosts
file or in inventory groups in case of AWX/Tower
hosts_group: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>ontap</i></span>


#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>false</i></span>


#Names of the Nodes in the ONTAP Cluster
nodes:
 - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
01</i></span>
 - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
02</i></span>


#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
```

```
  - {vlan_id: &quot;<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>203</i></span>&quot;, name: &quot;<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>infra_NFS</i></span>&quot;,
protocol: &quot;<span <div
contenteditable="true"/><i>NFS</i></span>&quot;}
<a id="more_storage_vlans" href="javascript:storagevlandropdown();">More
Storage VLANs</a><div id="select_more_storage_vlans"></div><a
id="more_storage_vlans_button" href="javascript:addstoragevlans();">Enter
Storage VLANs details</a><div id="extra_storage_vlans"></div>


#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>}
  - {aggr_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node02</i></span>}


#SVM name
svm_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>ora_svm</i></span>



# SVM Management LIF Details
svm_mgmt_details:
  - {address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.91.100</i></span>, netmask: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>255.255.255.0</i></span>,
home_port: <span <div contenteditable="true"/><i>e0M</i></span>}


# NFS storage parameters when data_protocol set to NFS. Volume named after
Oracle hosts name identified by mount point as follow for oracle DB server
1. Each mount point dedicates to a particular Oracle files: u01 - Oracle
binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by
click on "More NFS volumes" and also add the volumes list to corresponding
host_vars as host_datastores_nfs variable. For multiple DB server
```

deployment, additional volumes sets needs to be added for additional DB server. Input variable "{{groups.oracle[1]}}_u01", "{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for second DB server. Place volumes for multiple DB servers alternatingly between controllers for balanced IO performance, e.g. DB server 1 on controller node1, DB server 2 on controller node2 etc. Make sure match lif address with controller node.

```
volumes_nfs:
  - {vol_name: &quot<span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>{{groups.oracle[0]}}_u01</i></span>&quot, aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>, size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>{{groups.oracle[0]}}_u02</i></span>&quot, aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>, size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: &quot<span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>{{groups.oracle[0]}}_u03</i></span>&quot, aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>, size: <span <div contenteditable="true"/><i>25</i></span>}
<a id="more_nfs_volumes" href="javascript:nfsvolumesdropdown();">More NFS volumes</a><div id="select_more_nfs_volumes"></div><a id="more_nfs_volumes_button" href="javascript:addnfsvolumes();">Enter NFS volumes' details</a><div id="extra_nfs_volumes"></div>

#NFS LIFs IP address and netmask
nfs_lifs_details:
  - address: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span> #for node-1
    netmask: <span <div contenteditable="true" style="color:#004EFF; font-
```

```
weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>
  - address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.201</i></span> #for node-2
    netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>

#NFS client match
client_match: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.0/24</i></span>

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

#NFS Mount points for Oracle DB volumes
mount_points:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03</i></span>

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
```

```
######################################################
### DB env specific install and config variables ###
######################################################

db_domain: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>your.domain.com</i></span>

# Set initial password for all required Oracle passwords. Change them
after installation.
initial_pwd_all: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>netapp123</i></span>

</div></code></pre></div></div>
<script>
function CopyClassText(){
    var textToCopy = document.getElementById("CopyMeID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_nfs_volumes").style.display = "none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button").innerHTML = "Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
```

```
        document.getElementById("copy-button").innerHTML = "Copy";
        document.getElementById("more_storage_vlans").style.display =
"block";
        document.getElementById("more_nfs_volumes").style.display = "block";
}
function storagevlandropdown() {
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_storage_vlans_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_storage_vlans");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_storage_vlans_info">How many extra VLANs do you
wish to add?</a><select name="number_of_extra_storage_vlans"
id="number_of_extra_storage_vlans">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addstoragevlans() {
    var y =
document.getElementById("number_of_extra_storage_vlans").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_storage_vlans");
    while (j < y) {
        j++;
        myHTML += '   - {vlan_id: &quot;<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>203</i></span>&quot;, name: &quot;<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>infra_NFS</i></span>&quot;,
protocol: &quot;<span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>NFS</i></span>&quot;}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_storage_vlans").style.display =
"none";
    document.getElementById("more_storage_vlans_button").style.display =
"none";
}
function nfsvolumesdropdown() {
```

```
    document.getElementById("more_nfs_volumes").style.display = "none";
    document.getElementById("more_nfs_volumes_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_nfs_volumes");
    while (x < 100) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_nfs_volumes_info">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_nfs_volumes"
id="number_of_extra_nfs_volumes">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addnfsvolumes() {
    var y = document.getElementById("number_of_extra_nfs_volumes").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_nfs_volumes");
    while (j < y) {
        j++;
        myHTML += '  - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>rtpora04_u01</i></span>, aggr_name: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>aggr01_node02</i></span>,
lif: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.201</i></span>, size: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_nfs_volumes").style.display =
"none";
    document.getElementById("more_nfs_volumes_button").style.display =
"none";
}

</script>
```

Run the playbook

After completing the required environment prerequisites and copying the variables into `vars.yml` and

`your_host.yml`, you are now ready to deploy the playbooks.

ⓘ | <username> must be changed to match your environment.

1. Run the ONTAP playbook by passing the correct tags and ONTAP cluster username. Fill the password for ONTAP cluster, and vsadmin when prompted.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t
ontap_config -e @vars/vars.yml
```

2. Run the Linux playbook to execute Linux portion of deployment. Input for admin ssh password as well as sudo password.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t
linux_config -e @vars/vars.yml
```

3. Run the Oracle playbook to execute Oracle portion of deployment. Input for admin ssh password as well as sudo password.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t
oracle_config -e @vars/vars.yml
```

Deploy Additional Database on Same Oracle Host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container database on the same server, complete the following steps:

1. Revise the host_vars variables.

   a. Go back to step 3 - Edit the `host_name.yml` file under `host_vars`.

   b. Change the Oracle SID to a different naming string.

   c. Change the listener port to different number.

   d. Change the EM Express port to a different number if you have installed EM Express.

   e. Copy and paste the revised host variables to the Oracle host variable file under `host_vars`.

2. Execute the playbook with the `oracle_config` tag as shown above in Run the playbook.

Validate Oracle installation

1. Log in to Oracle server as oracle user and execute the following commands:

```
ps -ef | grep ora
```

> (i) This will list oracle processes if installation completed as expected and oracle DB started

2. Log in to the database to check the db configuration settings and the PDBs created with the following command sets.

```
sqlplus / as sysdba
```

[oracle@localhost ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

```
select name, log_mode from v$database;
```

SQL> select name, log_mode from v$database;
NAME LOG_MODE
--------- ------------
CDB2 ARCHIVELOG

```
show pdbs;
```

SQL> show pdbs

```
     CON_ID CON_NAME                          OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                        READ ONLY  NO
         3 CDB2_PDB1                       READ WRITE NO
         4 CDB2_PDB2                       READ WRITE NO
         5 CDB2_PDB3                       READ WRITE NO
```

```
col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;
```

SQL> col svrname form a30
SQL> col dirname form a30
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME DIRNAME NFSVERSION
------------------------------ ------------------------------ ---------------
172.21.126.200 /rhelora03_u02 NFSv3.0
172.21.126.200 /rhelora03_u03 NFSv3.0
172.21.126.200 /rhelora03_u01 NFSv3.0

```
This confirms that dNFS is working properly.
```

```
3. Connect to database via listener to check hte Oracle listener
configuration with the following command. Change to the appropriate
listener port and database service name.
```

```
sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com
```

[oracle@localhost ~]$ sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 13:19:57 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed May 05 2021 17:11:11 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> show user

```
USER is "SYSTEM"
SQL> show con_name
CON_NAME
CDB2_PDB1
```

> This confirms that Oracle listener is working properly.

Where to go for help?

If you need help with the toolkit, please join the NetApp Solution Automation community support slack channel and look for the solution-automation channel to post your questions or inquires.

Oracle Database Data Protection

Solution Overview

Automated Data Protection for Oracle Databases

Organizations are automating their environments to gain efficiencies, accelerate deployments, and reduce manual effort. Configuration management tools like Ansible are being used to streamline enterprise database operations. In this solution, we demonstrate how you can use Ansible to automate the data protection of Oracle with NetApp ONTAP. By enabling storage administrators, systems administrators, and DBAs to consistently and rapidly setup data replication to an offsite data center or to public cloud, you achieve the following benefits:

- Eliminate design complexities and human errors, and implement a repeatable consistent deployment and best practices
- Decrease time for configuration of Intercluster replication, CVO instantiation, and recovery of Oracle databases
- Increase database administrators, systems and storage administrators productivity
- Provides database recovery workflow for ease of testing a DR scenario.

NetApp provides customers with validated Ansible modules and roles to accelerate deployment, configuration, and lifecycle management of your Oracle database environment. This solution provides instruction and Ansible playbook code, to help you:

On Prem to on prem replication

- Create intercluster lifs on source and destination
- Establish cluster and vserver peering
- Create and initialize SnapMirror of Oracle volumes
- Create a replication schedule through AWX/Tower for Oracle binaries, databases, and logs
- Restore Oracle DB on the destination, and bring database online

On Prem to CVO in AWS

- Create AWS connector
- Create CVO instance in AWS

- Add On-Prem cluster to Cloud Manager

- Create intercluster lifs on source

- Establish cluster and vserver peering

- Create and initialize SnapMirror of Oracle volumes

- Create a replication schedule through AWX/Tower for Oracle binaries, databases, and logs

- Restore Oracle DB on the destination, and bring database online

For more details or to begin, please see the overview videos below.

AWX/Tower Deployments

- Part 1: TBD

  video
- Part 2: TBD

  video

   After you are ready, click here for getting started with the solution.

Getting started

This solution has been designed to be run in an AWX/Tower environment.

AWX/Tower

For AWX/Tower environments, you are guided through creating an inventory of your ONTAP cluster management and Oracle server (IPs and hostnames), creating credentials, configuring a project that pulls the Ansible code from NetApp Automation Github, and the Job Template that launches the automation.

1. The solution has been designed to run in a private cloud scenario (on-premise to on-premise), and hybrid cloud (on-premise to public cloud Cloud Volumes ONTAP [CVO])

2. Fill out the variables specific to your environment, and copy and paste them into the Extra Vars fields in your job template.

3. After the extra vars have been added to your job template, you can launch the automation.

4. The automation is set to be ran three phases (Setup, Replication Schedule for Oracle Binaries, Database, Logs, and Replication Schedule just for Logs), and a forth phase to recovering the database at a DR site.

5. For detailed instructions for obtaining the keys and tokens necessary for the CVO Data Protection visit Gather Pre-requisites For CVO and Connector Deployments

Requirements

On-Prem |

Environment	Requirements
Ansible environment	AWX/Tower
	Ansible v.2.10 and higher
	Python 3
	Python libraries - netapp-lib - xmltodict - jmespath
ONTAP	ONTAP version 9.8 +
	Two data aggregates
	NFS vlan and ifgrp created
Oracle server(s)	RHEL 7/8
	Oracle Linux 7/8
	Network interfaces for NFS, public, and optional mgmt
	Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud)

CVO

Environment	Requirements
Ansible environment	AWX/Tower
	Ansible v.2.10 and higher
	Python 3
	Python libraries - netapp-lib - xmltodict - jmespath
ONTAP	ONTAP version 9.8 +
	Two data aggregates
	NFS vlan and ifgrp created
Oracle server(s)	
	RHEL 7/8
	Oracle Linux 7/8
	Network interfaces for NFS, public, and optional mgmt
	Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud)
	Set appropriate swap space on the Oracle EC2 instance, by default some EC2 instances are deployed with 0 swap

Environment	Requirements
Cloud Manager/AWS	AWS Access/Secret Key
	NetApp Cloud Manager Account
	NetApp Cloud Manager Refresh Token

Automation Details

On-Prem |

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations.
The following table describes which tasks are being automated.

Playbook	Tasks
ontap_setup	Pre-check of the ONTAP environment
	Creation of Intercluster LIFs on source cluster (OPTIONAL)
	Creation of Intercluster LIFs on destination cluster (OPTIONAL)
	Creation of Cluster and SVM Peering
	Creation of destination SnapMirror and Initialization of designated Oracle volumes
ora_replication_cg	Enable backup mode for each database in /etc/oratab
	Snapshot taken of Oracle Binary and Database volumes
	Snapmirror Updated
	Turn off backup mode for each database in /etc/oratab
ora_replication_log	Switch current log for each database in /etc/oratab
	Snapshot taken of Oracle Log volume
	Snapmirror Updated
ora_recovery	Break SnapMirror
	Enable NFS and create junction path for Oracle volumes on the destination
	Configure DR Oracle Host
	Mount and verify Oracle volumes
	Recover and start Oracle database

CVO

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations.
The following table describes which tasks are being automated.

Playbook	Tasks
cvo_setup	Pre-check of the environment
	AWS Configure/AWS Access Key ID/Secret Key/Default Region
	Creation of AWS Role
	Creation of NetApp Cloud Manager Connector instance in AWS
	Creation of Cloud Volumes ONTAP (CVO) instance in AWS
	Add On-Prem Source ONTAP Cluster to NetApp Cloud Manager
	Creation of destination SnapMirror and Initialization of designated Oracle volumes
ora_replication_cg	Enable backup mode for each database in /etc/oratab
	Snapshot taken of Oracle Binary and Database volumes
	Snapmirror Updated
	Turn off backup mode for each database in /etc/oratab
ora_replication_log	Switch current log for each database in /etc/oratab
	Snapshot taken of Oracle Log volume
	Snapmirror Updated
ora_recovery	Break SnapMirror
	Enable NFS and create junction path for Oracle volumes on the destination CVO
	Configure DR Oracle Host
	Mount and verify Oracle volumes
	Recover and start Oracle database

Default parameters

To simplify automation, we have preset many required Oracle parameters with default values. It is generally not necessary to change the default parameters for most deployments. A more advanced user can make changes to the default parameters with caution. The default parameters are located in each role folder under defaults directory.

License

You should read license information as stated in the Github repository. By accessing, downloading, installing, or using the content in this repository, you agree the terms of the license laid out here.

Note that there are certain restrictions around producing and/or sharing any derivative works with the content in this repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download, or use the content in this repository.

After you are ready, click here for detailed AWX/Tower procedures.

Step-by-step deployment procedure

1. Create the inventory, group, hosts, and credentials for your environment

This section describes the setup of inventory, groups, hosts, and access credentials in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.

   a. Navigate to Resources → Inventories → Add, and click Add Inventory.

   b. Provide the name and organization details, and click Save.

   c. On the Inventories page, click the inventory created.

   d. Navigate to the Groups sub-menu and click Add.

   e. Provide the name oracle for your first group and click Save.

   f. Repeat the process for a second group called dr_oracle.

   g. Select the oracle group created, go to the Hosts sub-menu and click Add New Host.

   h. Provide the IP address of the Source Oracle host's management IP, and click Save.

   i. This process must be repeated for the dr_oracle group and add the the DR/Destination Oracle host's management IP/hostname.

   > ⓘ Below are instructions for creating the credential types and credentials for either On-Prem with ONTAP, or CVO on AWS.

On-Prem

1. Configure the credentials.

2. Create Credential Types. For solutions involving ONTAP, you must configure the credential type to match username and password entries.

   a. Navigate to Administration → Credential Types, and click Add.

   b. Provide the name and description.

   c. Paste the following content in Input Configuration:

   ```
   fields:
     - id: dst_cluster_username
       type: string
       label: Destination Cluster Username
     - id: dst_cluster_password
       type: string
       label: Destination Cluster Password
       secret: true
     - id: src_cluster_username
       type: string
       label: Source Cluster Username
     - id: src_cluster_password
       type: string
       label: Source Cluster Password
       secret: true
   ```

   d. Paste the following content into Injector Configuration and then click Save:

   ```
   extra_vars:
     dst_cluster_username: '{{ dst_cluster_username }}'
     dst_cluster_password: '{{ dst_cluster_password }}'
     src_cluster_username: '{{ src_cluster_username }}'
     src_cluster_password: '{{ src_cluster_password }}'
   ```

3. Create Credential for ONTAP

   a. Navigate to Resources → Credentials, and click Add.

   b. Enter the name and organization details for the ONTAP Credentials

   c. Select the credential type that was created in the previous step.

   d. Under Type Details, enter the Username and Password for your Source and Destination Clusters.

   e. Click Save

4. Create Credential for Oracle

   a. Navigate to Resources → Credentials, and click Add.

   b. Enter the name and organization details for Oracle

   c. Select the Machine credential type.

   d. Under Type Details, enter the Username and Password for the Oracle hosts.

   e. Select the correct Privilege Escalation Method, and enter the username and password.

   f. Click Save

   g. Repeat process if needed for a different credential for the dr_oracle host.

CVO

1. Configure the credentials.

2. Create credential types. For solutions involving ONTAP, you must configure the credential type to match username and password entries, we will also add entries for Cloud Central and AWS.

   a. Navigate to Administration → Credential Types, and click Add.

   b. Provide the name and description.

   c. Paste the following content in Input Configuration:

```
fields:
  - id: dst_cluster_username
    type: string
    label: CVO Username
  - id: dst_cluster_password
    type: string
    label: CVO Password
    secret: true
  - id: cvo_svm_password
    type: string
    label: CVO SVM Password
    secret: true
  - id: src_cluster_username
    type: string
    label: Source Cluster Username
  - id: src_cluster_password
    type: string
    label: Source Cluster Password
    secret: true
  - id: regular_id
    type: string
    label: Cloud Central ID
    secret: true
  - id: email_id
    type: string
    label: Cloud Manager Email
    secret: true
  - id: cm_password
    type: string
    label: Cloud Manager Password
    secret: true
  - id: access_key
    type: string
    label: AWS Access Key
    secret: true
  - id: secret_key
    type: string
    label: AWS Secret Key
    secret: true
  - id: token
    type: string
    label: Cloud Central Refresh Token
    secret: true
```

d. Paste the following content into Injector Configuration and click Save:

```
extra_vars:
  dst_cluster_username: '{{ dst_cluster_username }}'
  dst_cluster_password: '{{ dst_cluster_password }}'
  cvo_svm_password: '{{ cvo_svm_password }}'
  src_cluster_username: '{{ src_cluster_username }}'
  src_cluster_password: '{{ src_cluster_password }}'
  regular_id: '{{ regular_id }}'
  email_id: '{{ email_id }}'
  cm_password: '{{ cm_password }}'
  access_key: '{{ access_key }}'
  secret_key: '{{ secret_key }}'
  token: '{{ token }}'
```

3. Create Credential for ONTAP/CVO/AWS

   a. Navigate to Resources → Credentials, and click Add.

   b. Enter the name and organization details for the ONTAP Credentials

   c. Select the credential type that was created in the previous step.

   d. Under Type Details, enter the Username and Password for your Source and CVO Clusters, Cloud Central/Manager, AWS Access/Secret Key and Cloud Central Refresh Token.

   e. Click Save

4. Create Credential for Oracle (Source)

   a. Navigate to Resources → Credentials, and click Add.

   b. Enter the name and organization details for Oracle host

   c. Select the Machine credential type.

   d. Under Type Details, enter the Username and Password for the Oracle hosts.

   e. Select the correct Privilege Escalation Method, and enter the username and password.

   f. Click Save

5. Create Credential for Oracle Destination

   a. Navigate to Resources → Credentials, and click Add.

   b. Enter the name and organization details for the DR Oracle host

   c. Select the Machine credential type.

   d. Under Type Details, enter the Username (ec2-user or if you have changed it from default enter that), and the SSH Private Key

   e. Select the correct Privilege Escalation Method (sudo), and enter the username and password if needed.

   f. Click Save

2. Create a project

1. Go to Resources → Projects, and click Add.

a. Enter the name and organization details.

b. Select Git in the Source Control Credential Type field.

c. enter https://github.com/NetApp-Automation/na_oracle19c_data_protection.git as the source control URL.

d. Click Save.

e. The project might need to sync occasionally when the source code changes.

3. Configure global variables

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.

> ℹ️ The items in blue must be changed to match your environment.

On-Prem

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_binary_vols {
  display: block;
}
#more_binary_vols_button {
  display: none;
}
#more_database_vols {
  display: block;
}
#more_database_vols_button {
  display: none;
}
#more_log_vols {
  display: block;
}
#more_log_vols_button {
  display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button
id="copy-button-onprem"
onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyOnPrem">
################################################################
###
```

```
###### Oracle Data Protection global user configuration variables
######
######  Consolidate all variables from ontap, aws, and oracle
######
#################################################################
###

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

#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in
inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>false</i></span>

#######################################################
# Inter-cluster LIF details
#######################################################
#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
02</i></span>

 #Names of the Nodes in the Destination ONTAP Cluster
dst_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>DR-
AFF-01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>DR-
AFF-02</i></span>
```

```
#Define whether or not to create intercluster lifs on source cluster
(ONLY CHANGE to 'No' IF YOU HAVE ALREADY CREATED THE INTERCLUSTER
LIFS)
create_source_intercluster_lifs:  <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>

source_intercluster_network_port_details:
  using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_ifgrp: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_vlans: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;"/><i>yes</i></span>
  ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a</i></span>
  vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10</i></span>
  ports:
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0b</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0g</i></span>
  broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>NFS</i></span>
  ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>Default</i></span>
  failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>iclifs</i></span>

source_intercluster_lif_details:
```

```
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_1</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.1</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>AFF-01</i></span>
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_2</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.2</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>AFF-02</i></span>

#Define whether or not to create intercluster lifs on destination
cluster (ONLY CHANGE to 'No' IF YOU HAVE ALREADY CREATED THE
INTERCLUSTER LIFS)
create_destination_intercluster_lifs:  <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;"/><i>yes</i></span>

destination_intercluster_network_port_details:
  using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_ifgrp: <span <div contenteditable="true"
```

```
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_vlans: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;"/><i>yes</i></span>
  ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a</i></span>
  vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10</i></span>
  ports:
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0b</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0g</i></span>
  broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>NFS</i></span>
  ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>Default</i></span>
  failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>iclifs</i></span>

destination_intercluster_lif_details:
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_1</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.3</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
```

```
10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>DR-AFF-01</i></span>
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_2</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.4</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>DR-AFF-02</i></span>

#####################################################################
###############################
# Variables for SnapMirror Peering
#####################################################################
###############################
#src_lif: #Will be retrieve through Ansible Task
#dst_lif: #Will be retrieve through Ansible Task
passphrase: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>your-passphrase</i></span>

#####################################################################
###############################
# Source & Destination List
#####################################################################
###############################
#Please Enter Destination Cluster Name
dst_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>dst-cluster-
name</i></span>

#Please Enter Destination Cluster
dst_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
```

```
decoration:underline; text-decoration:underline;"/><i>dst-cluster-
ip</i></span>


#Please Enter Destination SVM to create mirror relationship
dst_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>dst-vserver</i></span>


#Please Enter NFS Lif for dst vserver
dst_nfs_lif: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>dst-nfs-lif</i></span>


#Please Enter Source Cluster Name
src_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
name</i></span>


#Please Enter Source Cluster
src_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
ip</i></span>


#Please Enter Source SVM
src_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>src-vserver</i></span>


####################################################################
################################
# Variable for Oracle Volumes and SnapMirror Details
####################################################################
################################
#Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>oracle</i></span>


#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>binary_vol</i></span>
```

```
<a id="more_binary_vols"
href="javascript:binaryvolsdropdown();">More Binary Vols</a><div
id="select_more_binary_vols"></div><a id="more_binary_vols_button"
href="javascript:addbinaryvols();">Enter Volume details</a><div
id="extra_binary_vols"></div>
#Please Enter Source Database Volume(s)
src_db_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>db_vol</i></span>
<a id="more_database_vols"
href="javascript:databasevolsdropdown();">More Database Vols</a><div
id="select_more_database_vols"></div><a
id="more_database_vols_button"
href="javascript:adddatabasevols();">Enter Volume details</a><div
id="extra_database_vols"></div>
#Please Enter Source Archive Volume(s)
src_archivelog_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>log_vol</i></span>
<a id="more_log_vols" href="javascript:logvolsdropdown();">More Log
Vols</a><div id="select_more_log_vols"></div><a
id="more_log_vols_button" href="javascript:addlogvols();">Enter
Volume details</a><div id="extra_log_vols"></div>
#Please Enter Destination Snapmirror Policy
snapmirror_policy: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>async_policy_oracle</i></span>

####################################################################
################################
# Export Policy Details
####################################################################
################################
#Enter the destination export policy details
export_policy_details:
  name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>nfs_export_policy</i></span>
  client_match: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>0.0.0.0/0</i></span>
  ro_rule: sys
```

```
   rw_rule: sys


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

#NFS Mount points for Oracle DB volumes
mount_points:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03</i></span>

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated
to each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>

####################################################
### DB env specific install and config variables ###
####################################################
#Recovery Type (leave as scn)
recovery_type: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>scn</i></span>

#Oracle Control Files
control_files:
  - <span <div contenteditable="true" style="color:#004EFF; font-
```

```
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02/oradata/CDB2/control01.ctl</i></span
>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03/orareco/CDB2/control02.ctl</i></span
>

</div></code></pre></div></div>
<script>
function CopyClassText(){
    var textToCopy = document.getElementById("CopyOnPrem");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_binary_vols").style.display =
"none";
    document.getElementById("more_database_vols").style.display =
"none";
    document.getElementById("more_log_vols").style.display = "none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button-onprem").innerHTML =
"Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
      document.getElementById("copy-button-onprem").innerHTML =
"Copy";
```

```
        document.getElementById("more_binary_vols").style.display =
"block";
        document.getElementById("more_database_vols").style.display =
"block";
        document.getElementById("more_log_vols").style.display =
"block";
}
function binaryvolsdropdown() {
    document.getElementById("more_binary_vols").style.display =
"none";
    document.getElementById("more_binary_vols_button").style.display
= "block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_binary_vols");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_binary_vols_info">How many extra volumes
do you wish to add?</a><select name="number_of_extra_binary_vols"
id="number_of_extra_binary_vols">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addbinaryvols() {
    var y =
document.getElementById("number_of_extra_binary_vols").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_binary_vols");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>binary_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_binary_vols").style.display
= "none";
    document.getElementById("more_binary_vols_button").style.display
= "none";
}
function databasevolsdropdown() {
```

```
    document.getElementById("more_database_vols").style.display =
"none";

document.getElementById("more_database_vols_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_database_vols");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_database_vols"
id="number_of_extra_database_vols">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatabasevols() {
    var y =
document.getElementById("number_of_extra_database_vols").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_database_vols");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>db_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_database_vols").style.display =
"none";

document.getElementById("more_database_vols_button").style.display =
"none";
}
function logvolsdropdown() {
    document.getElementById("more_log_vols").style.display = "none";
    document.getElementById("more_log_vols_button").style.display =
"block";
    var x=1;
```

```
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_log_vols");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_log_vols" id="number_of_extra_log_vols">' +
buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addlogvols() {
    var y =
document.getElementById("number_of_extra_log_vols").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_log_vols");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>log_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_log_vols").style.display =
"none";
    document.getElementById("more_log_vols_button").style.display =
"none";
}

</script>
```

CVO

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
```

```
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_binary_vols1 {
  display: block;
}
#more_binary_vols1_button {
  display: none;
}
#more_database_vols1 {
  display: block;
}
#more_database_vols1_button {
  display: none;
}
#more_log_vols1 {
  display: block;
}
#more_log_vols1_button {
  display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button
id="copy-button-cvo"
onclick="CopyClassTextCVO()">Copy</button></div><pre><code><div
class="CopyMeClassCVO" id="CopyCVO">
#######################################################################
###
###### Oracle Data Protection global user configuration variables
######
######  Consolidate all variables from ontap, aws, CVO and oracle
######
#######################################################################
###


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

```
#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in
inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>false</i></span>

#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>AFF-
02</i></span>

 #Names of the Nodes in the Destination CVO Cluster
dst_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>DR-
AFF-01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>DR-
AFF-02</i></span>

#Define whether or not to create intercluster lifs on source cluster
(ONLY CHANGE to 'No' IF YOU HAVE ALREADY CREATED THE INTERCLUSTER
LIFS)
create_source_intercluster_lifs:  <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>

source_intercluster_network_port_details:
  using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_ifgrp: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
```

```
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  using_vlans: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
  failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;"/><i>yes</i></span>
  ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a</i></span>
  vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10</i></span>
  ports:
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0b</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0g</i></span>
  broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>NFS</i></span>
  ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>Default</i></span>
  failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>iclifs</i></span>

source_intercluster_lif_details:
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_1</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.1</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>
```

```
      node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>AFF-01</i></span>
  - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_2</i></span>
    address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.2</i></span>
    netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
    home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>AFF-02</i></span>


#############################################
### CVO Deployment Variables ###
#############################################


####### Access Keys Variables ######

# Region where your CVO will be deployed.
region_deploy: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>us-east-1</i></span>


########### CVO and Connector Vars ########

# AWS Managed Policy required to give permission for IAM role
creation.
aws_policy: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>arn:aws:iam::1234567:policy/OCCM</i></spa
n>


# Specify your aws role name, a new role is created if one already
does not exist.
aws_role_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>arn:aws:iam::1234567:policy/OCCM</i></spa
n>
```

```
# Name your connector.
connector_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>awx_connector</i></span>

# Name of the key pair generated in AWS.
key_pair: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>key_pair</i></span>

# Name of the Subnet that has the range of IP addresses in your VPC.
subnet: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>subnet-12345</i></span>

# ID of your AWS secuirty group that allows access to on-prem
resources.
security_group: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>sg-123123123</i></span>

# You Cloud Manager Account ID.
account: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>account-A23123A</i></span>

# Name of the your CVO instance
cvo_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>test_cvo</i></span>

# ID of the VPC in AWS.
vpc: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"/><i>vpc-
123123123</i></span>

######################################################################
##############################
# Variables for - Add on-prem ONTAP to Connector in Cloud Manager
######################################################################
##############################

# For Federated users, Client ID from API Authentication Section of
Cloud Central to generate access token.
sso_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
```

```
decoration:underline;"/><i>123123123123123123123</i></span>

# For regular access with username and password, please specify
"pass" as the connector_access. For SSO users, use "refresh_token"
as the variable.
connector_access: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>pass</i></span>

##################################################################
###############################
# Variables for SnapMirror Peering
##################################################################
###############################
passphrase: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>your-passphrase</i></span>

##################################################################
################################
# Source & Destination List
##################################################################
###############################
#Please Enter Destination Cluster Name
dst_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>dst-cluster-
name</i></span>

#Please Enter Destination Cluster (Once CVO is Created Add this
Variable to all templates)
dst_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>dst-cluster-
ip</i></span>

#Please Enter Destination SVM to create mirror relationship
dst_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>dst-vserver</i></span>

#Please Enter NFS Lif for dst vserver (Once CVO is Created Add this
Variable to all templates)
dst_nfs_lif: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
```

```
text-decoration:underline;"/><i>dst-nfs-lif</i></span>

#Please Enter Source Cluster Name
src_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
name</i></span>

#Please Enter Source Cluster
src_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
ip</i></span>

#Please Enter Source SVM
src_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>src-vserver</i></span>

###################################################################
################################
# Variable for Oracle Volumes and SnapMirror Details
###################################################################
################################
#Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>oracle</i></span>

#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>binary_vol</i></span>
<a id="more_binary_vols1"
href="javascript:binaryvols1dropdown();">More Binary Vols</a><div
id="select_more_binary_vols1"></div><a id="more_binary_vols1_button"
href="javascript:addbinaryvols1();">Enter Volume details</a><div
id="extra_binary_vols1"></div>
#Please Enter Source Database Volume(s)
src_db_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>db_vol</i></span>
<a id="more_database_vols1"
```

```
href="javascript:databasevols1dropdown();">More Database
Vols</a><div id="select_more_database_vols1"></div><a
id="more_database_vols1_button"
href="javascript:adddatabasevols1();">Enter Volume details</a><div
id="extra_database_vols1"></div>
#Please Enter Source Archive Volume(s)
src_archivelog_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>log_vol</i></span>
<a id="more_log_vols1" href="javascript:logvols1dropdown();">More
Log Vols</a><div id="select_more_log_vols1"></div><a
id="more_log_vols1_button" href="javascript:addlogvols1();">Enter
Volume details</a><div id="extra_log_vols1"></div>
#Please Enter Destination Snapmirror Policy
snapmirror_policy: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>async_policy_oracle</i></span>


##########################################################################
##################################
# Export Policy Details
##########################################################################
##################################
#Enter the destination export policy details (Once CVO is Created
Add this Variable to all templates)
export_policy_details:
  name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>nfs_export_policy</i></span>
  client_match: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>0.0.0.0/0</i></span>
  ro_rule: sys
  rw_rule: sys


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

#NFS Mount points for Oracle DB volumes
mount_points:
```

```
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03</i></span>

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated
to each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>

####################################################
### DB env specific install and config variables ###
####################################################
#Recovery Type (leave as scn)
recovery_type: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>scn</i></span>

#Oracle Control Files
control_files:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02/oradata/CDB2/control01.ctl</i></span
>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03/orareco/CDB2/control02.ctl</i></span
>
```

```
</div></code></pre></div></div>
<script>
function CopyClassTextCVO(){
    var textToCopy = document.getElementById("CopyCVO");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_binary_vols1").style.display =
"none";
    document.getElementById("more_database_vols1").style.display =
"none";
    document.getElementById("more_log_vols1").style.display =
"none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button-cvo").innerHTML =
"Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
      document.getElementById("copy-button-cvo").innerHTML = "Copy";
      document.getElementById("more_binary_vols1").style.display =
"block";
      document.getElementById("more_database_vols1").style.display =
"block";
      document.getElementById("more_log_vols1").style.display =
"block";
}
function binaryvols1dropdown() {
```

```
    document.getElementById("more_binary_vols1").style.display =
"none";

document.getElementById("more_binary_vols1_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_binary_vols1");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_binary_vols1_info">How many extra volumes
do you wish to add?</a><select name="number_of_extra_binary_vols1"
id="number_of_extra_binary_vols1">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addbinaryvols1() {
    var y =
document.getElementById("number_of_extra_binary_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_binary_vols1");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>binary_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_binary_vols1").style.display =
"none";

document.getElementById("more_binary_vols1_button").style.display =
"none";
}
function databasevols1dropdown() {
    document.getElementById("more_database_vols1").style.display =
"none";

document.getElementById("more_database_vols1_button").style.display
= "block";
```

```
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_database_vols1");
    while (x < 10) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_database_vols1_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_database_vols1"
id="number_of_extra_database_vols1">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatabasevols1() {
    var y =
document.getElementById("number_of_extra_database_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_database_vols1");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>db_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_database_vols1").style.display
= "none";

document.getElementById("more_database_vols1_button").style.display
= "none";
}
function logvols1dropdown() {
    document.getElementById("more_log_vols1").style.display =
"none";
    document.getElementById("more_log_vols1_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_log_vols1");
    while (x < 10) {
```

```
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_log_vols1" id="number_of_extra_log_vols1">' +
buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addlogvols1() {
    var y =
document.getElementById("number_of_extra_log_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_log_vols1");
    while (j < y) {
        j++;
        myHTML += '  - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>log_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_log_vols1").style.display =
"none";
    document.getElementById("more_log_vols1_button").style.display =
"none";
}

</script>
```

4. Automation Playbooks

There are four separate playbooks that need to be ran.

1. Playbook for Setting up your environment, On-Prem or CVO.

2. Playbook for replicating Oracle Binaries and Databases on a schedule

3. Playbook for replicating Oracle Logs on a schedule

4. Playbook for Recovering your database on a destination host

ONTAP/CVO Setup

ONTAP and CVO Setup

1. Configure and launch the job template.

    1. Create the job template.

        a. Navigate to Resources → Templates → Add and click Add Job Template.

        b. Enter the name ONTAP/CVO Setup

        c. Select the Job type; Run configures the system based on a playbook.

        d. Select the corresponding inventory, project, playbook, and credentials for the playbook.

        e. Select the ontap_setup.yml playbook for an On-Prem environment or select the cvo_setup.yml for replicating to a CVO instance.

        f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.

        g. Click Save.

    2. Launch the job template.

        a. Navigate to Resources → Templates.

        b. Click the desired template and then click Launch.

        > ℹ️ We will use this template and copy it out for the other playbooks.

Replication For Binary and Database Volumes

Scheduling the Binary and Database Replication Playbook

1. Configure and launch the job template.

    1. Copy the previously created job template.

        a. Navigate to Resources → Templates.

        b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template

        c. Click Edit Template on the copied template, and change the name to Binary and Database Replication Playbook.

        d. Keep the same inventory, project, credentials for the template.

        e. Select the ora_replication_cg.yml as the playbook to be executed.

        f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.

        g. Click Save.

    2. Schedule the job template.

        a. Navigate to Resources → Templates.

        b. Click the Binary and Database Replication Playbook template and then click Schedules at the top set of options.

        c. Click Add, add Name Schedule for Binary and Database Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.

> **ⓘ** A separate schedule will be created for the Log volume replication, so that it can be replicated on a more frequent cadence.

Replication for Log Volumes

Scheduling the Log Replication Playbook

1. Configure and launch the job template.

    1. Copy the previously created job template.

        a. Navigate to Resources → Templates.

        b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template

        c. Click Edit Template on the copied template, and change the name to Log Replication Playbook.

        d. Keep the same inventory, project, credentials for the template.

        e. Select the ora_replication_logs.yml as the playbook to be executed.

        f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.

        g. Click Save.

    2. Schedule the job template.

        a. Navigate to Resources → Templates.

        b. Click the Log Replication Playbook template and then click Schedules at the top set of options.

        c. Click Add, add Name Schedule for Log Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.

> **ⓘ** It is recommended to set the log schedule to update every hour to ensure the recovery to the last hourly update.

Restore and Recover Database

Scheduling the Log Replication Playbook

1. Configure and launch the job template.

    1. Copy the previously created job template.

        a. Navigate to Resources → Templates.

        b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template

        c. Click Edit Template on the copied template, and change the name to Restore and Recovery Playbook.

        d. Keep the same inventory, project, credentials for the template.

        e. Select the ora_recovery.yml as the playbook to be executed.

        f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.

        g. Click Save.

> **ⓘ** This playbook will not be ran until you are ready to restore your database at the remote site.

5. Recovering Oracle Database

1. On-premises production Oracle databases data volumes are protected via NetApp SnapMirror replication to either a redundant ONTAP cluster in secondary data center or Cloud Volume ONTAP in public cloud. In a fully configured disaster recovery environment, recovery compute instances in secondary data center or public cloud are standby and ready to recover the production database in the case of a disaster. The standby compute instances are kept in sync with on-prem instances by running paraellel updates on OS kernel patch or upgrade in a lockstep.

2. In this solution demonstrated, Oracle binary volume is replicated to target and mounted at target instance to bring up Oracle software stack. This approach to recover Oracle has advantage over a fresh installation of Oracle at last minute when a disaster occurred. It guarantees Oracle installation is fully in sync with current on-prem production software installation and patch levels etc. However, this may or may not have additional sofware licensing implication for the replicated Oracle binary volume at recovery site depending on how the software licensing is structured with Oracle. User is recommended to check with its software licensing personnel to assess the potential Oracle licensing requirement before deciding to use the same approach.

3. The standby Oracle host at the destination is configured with the Oracle prerequisite configurations.

4. The SnapMirrors are broken and the volumes are made writable and mounted to the standby Oracle host.

5. The Oracle recovery module performs following tasks to recovery and startup Oracle at recovery site after all DB volumes are mounted at standby compute instance.

   a. Sync the control file: We deployed duplicate Oracle control files on different database volume to protect critical database control file. One is on the data volume and another is on log volume. Since data and log volumes are replicated at different frequency, they will be out of sync at the time of recovery.

   b. Relink Oracle binary: Since the Oracle binary is relocated to a new host, it needs a relink.

   c. Recover Oracle database: The recovery mechanism retrieves last System Change Number in last available archived log in Oracle log volume from control file and recovers Oracle database to recoup all business transactions that was able to be replicated to DR site at the time of failure. The database is then started up in a new incarnation to carry on user connections and business transaction at recovery site.

> **ⓘ** Before running the Recovering playbook make sure you have the following:
> Make sure it copy over the /etc/oratab and /etc/oraInst.loc from the source Oracle host to the destination host

Oracle Database Deployment on AWS EC2/FSx Best Practices

WP-7357: Oracle Database Deployment on EC2/FSx Best Practices Introduction

Allen Cao, Niyaz Mohamed, Jeffrey Steiner, NetApp

Many mission-critical enterprise Oracle databases are still hosted on-premises, and many enterprises are looking to migrate these Oracle databases to a public cloud. Often, these Oracle databases are application centric and thus require user-specific configurations, a capability that is missing from many database-as-a-

service public-cloud offerings. Therefore, the current database landscape calls for a public-cloud-based Oracle database solution built from a high-performance, scalable compute and storage service that can accommodate unique requirements. AWS EC2 compute instances and the AWS FSx storage service might be the missing pieces of this puzzle that you can leverage to build and migrate your mission critical Oracle database workloads to a public cloud.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for enterprises. The simple Amazon EC2 web-service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon FSx for ONTAP is an AWS storage service that uses industry-leading NetApp ONTAP block and file storage, which exposes NFS, SMB, and iSCSI. With such a powerful storage engine, it has never been easier to relocate mission-critical Oracle database apps to AWS with sub-millisecond response times, multiple GBps of throughput, and 100,000+ IOPS per database instance. Better yet, the FSx storage service comes with native replication capability that allows you to easily migrate your on-premises Oracle database to AWS or to replicate your mission critical Oracle database to a secondary AWS availability zone for HA or DR.

The goal of this documentation is to provide step-by-step processes, procedures, and best-practice guidance on how to deploy and configure an Oracle database with FSx storage and an EC2 instance that delivers performance similar to an on-premises system. NetApp also provides an automation toolkit that automates most of the tasks that are required for the deployment, configuration, and management of your Oracle database workload in the AWS public cloud.

Solution Architecture

The following architecture diagram illustrates a highly available Oracle database deployment on an AWS EC2 instance with the FSx storage service. A similar deployment scheme but with the standby in a different region can be set up for disaster recovery.

Within the environment, the Oracle compute instance is deployed via an AWS EC2 instance console. There are multiple EC2 instance types available from the console. NetApp recommends deploying a database-oriented EC2 instance type such as an m5 Ami image with RedHat enterprise Linux 8 and up to 10Gps network bandwidth.

Oracle database storage on FSx volumes on the other hand is deployed with the AWS FSx console or CLI. The Oracle binary, data, or log volumes are subsequently presented and mounted on an EC2 instance Linux host. Each data or log volume can have multiple LUNs allocated depending on the underlying storage protocol employed.

An FSx storage cluster is designed with double redundancy, so that both the primary and standby storage clusters are deployed in two different availability zones. Database volumes are replicated from a primary FSx cluster to a standby FSx cluster at a user-configurable interval for all Oracle binary, data and log volumes.

This high availability Oracle environment is managed with an Ansible controller node and a SnapCenter backup server and UI tool. Oracle installation, configuration, and replication are automated using Ansible playbook-based toolkits. Any update to the Oracle EC2 instance kernel operating system or Oracle patching can be executed in parallel to keep the primary and standby in sync. In fact, the initial automation setup can be easily expanded to perform some repeating daily Oracle tasks if needed.

SnapCenter provides workflows for Oracle database point-in-time recovery or for database cloning at either the primary or standby zones if needed. Through the SnapCenter UI, you can configure Oracle database backup and replication to standby FSx storage for high availability or disaster recovery based on your RTO or RPO objectives.

The solution provides an alternative process that delivers capabilities similar to those available from Oracle RAC and Data Guard deployment.

Factors to consider for Oracle database deployment

A public cloud provides many choices for compute and storage, and using the correct type of compute instance and storage engine is a good place to start for database deployment. You should also select compute and storage configurations that are optimized for Oracle databases.

The following sections describe the key considerations when deploying Oracle database in an AWS public cloud on an EC2 instance with FSx storage.

VM performance

Selecting the right VM size is important for optimal performance of a relational database in a public cloud. For better performance, NetApp recommends using an EC2 M5 Series instance for Oracle deployment, which is optimized for database workloads. The same instance type is also used to power a RDS instance for Oracle by AWS.

- Choose the correct vCPU and RAM combination based on workload characteristics.
- Add swap space to a VM. The default EC2 instance deployment does not create a swap space, which is not optimal for a database.

Storage layout and settings

NetApp recommends the following storage layout:

- For NFS storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file.

- For iSCSI storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file. However, each data and log volume ideally should contain four LUNs. The LUNs are ideally balanced on the HA cluster nodes.

- For storage IOPS and throughput, you can choose the threshold for provisioned IOPS and throughput for the FSx storage cluster, and these parameters can be adjusted on the fly anytime the workload changes.
  - The auto IOPS setting is three IOPS per GiB of allocated storage capacity or user defined storage up to 80,000.

- The throughput level is incremented as follow: 128, 256, 512, 1024, 2045 MBps.

Review the Amazon FSx for NetApp ONTAP performance documentation when sizing throughput and IOPS.

NFS configuration

Linux, the most common operating system, includes native NFS capabilities. Oracle offers the direct NFS (dNFS) client natively integrated into Oracle. Oracle has supported NFSv3 for over 20 years, and NFSv4 is supported with Oracle 12.1.0.2 and later. Automated Oracle deployment using the NetApp automation toolkit automatically configures dNFS on NFSv3.

Other factors to consider:

- TCP slot tables are the NFS equivalent of host-bus-adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

  For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables to 128.

  ```
  sysctl -a | grep tcp.*.slot_table
  ```

- The following table provides recommended NFS mount options for Linux NFSv3 - single instance.

File Type	Mount Options
• Control files • Data files • Redo logs	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
• ORACLE_HOME • ORACLE_BASE	rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536

> ⓘ Before using dNFS, verify that the patches described in Oracle Doc 1495104.1 are installed. Starting with Oracle 12c, DNFS includes support for NFSv3, NFSv4, and NFSv4.1. NetApp support policies cover v3 and v4 for all clients, but, at the time of writing, NFSv4.1 is not supported for use with Oracle dNFS.

High availability

As indicated in the solution architecture, HA is built on storage-level replication. Therefore, the startup and availability of Oracle is contingent on how quickly the compute and storage can be brought up and recovered. See the following key factors:

- Have a standby compute instance ready and synced up with the primary through Ansible parallel update to both hosts.

- Replicate the binary volume from the primary for standby purposes so that you do not need to install Oracle at the last minute and figure out what needs to be installed and patched.

- Replication frequency dictates how fast the Oracle database can be recovered to make service available.

There is a trade off between the replication frequency and storage consumption.

- Leverage automation to make recovery and switch over to standby quick and free of human error. NetApp provides an automation toolkit for this purpose.

Step-by-Step Oracle Deployment Procedures on AWS EC2/FSx

Deploy an EC2 Linux instance for Oracle via EC2 console

If you are new to AWS, you first need to set up an AWS environment. The documentation tab at the AWS website landing page provides EC2 instruction links on how to deploy a Linux EC2 instance that can be used to host your Oracle database via the AWS EC2 console. The following section is a summary of these steps. For details, see the linked AWS EC2-specific documentation.

Setting up your AWS EC2 environment

You must create an AWS account to provision the necessary resources to run your Oracle environment on the EC2 and FSx service. The following AWS documentation provides the necessary details:

- Set up to use Amazon EC2

Key topics:

- Sign up for AWS.
- Create a key pair.
- Create a security group.

Enabling multiple availability zones in AWS account attributes

For an Oracle high availability configuration as demonstrated in the architecture diagram, you must enable at least four availability zones in a region. The multiple availability zones can also be situated in different regions to meet the required distances for disaster recovery.

Creating and connecting to an EC2 instance for hosting Oracle database

See the tutorial Get started with Amazon EC2 Linux instances for step-by-step deployment procedures and best practices.

Key topics:

- Overview.
- Prerequisites.
- Step 1: Launch an instance.
- Step 2: Connect to your instance.
- Step 3: Clean up your instance.

The following screen shots demonstrate the deployment of an m5-type Linux instance with the EC2 console for running Oracle.

1. From the EC2 dashboard, click the yellow Launch Instance button to start the EC2 instance deployment workflow.

2. In Step 1, select "Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)."

3. In Step 2, select an m5 instance type with the appropriate CPU and memory allocation based on your Oracle database workload. Click "Next: Configure Instance Details."

4. In Step 3, choose the VPC and subnet where the instance should be placed and enable public IP assignment. Click "Next: Add Storage."

5. In Step 4, allocate enough space for the root disk. You may need the space to add a swap. By default, EC2 instance assign zero swap space, which is not optimal for running Oracle.

6. In Step 5, add a tag for instance identification if needed.

7. In Step 6, select an existing security group or create a new one with the desired inbound and outbound policy for the instance.

8. In Step 7, review the instance configuration summary, and click Launch to start instance deployment. You are prompted to create a key pair or select a key pair for access to the instance.

9. Log into EC2 instance using an SSH key pair. Make changes to your key name and instance IP address as appropriate.

```
ssh -i ora-db1v2.pem ec2-user@54.80.114.77
```

You need to create two EC2 instances as primary and standby Oracle servers in their designated availability

zone as demonstrated in the architecture diagram.

Provision FSx for ONTAP file systems for Oracle database storage

EC2 instance deployment allocates an EBS root volume for the OS. FSx for ONTAP file systems provides Oracle database storage volumes, including the Oracle binary, data, and log volumes. The FSx storage NFS volumes can be either provisioned from the AWS FSx console or from Oracle installation, and configuration automation that allocates the volumes as the user configures in a automation parameter file.

Creating FSx for ONTAP file systems

Referred to this documentation Managing FSx for ONTAP file systems for creating FSx for ONTAP file systems.

Key considerations:

- SSD storage capacity. Minimum 1024 GiB, maximum 192 TiB.
- Provisioned SSD IOPS. Based on workload requirements, a maximum of 80,000 SSD IOPS per file system.
- Throughput capacity.
- Set administrator fsxadmin/vsadmin password. Required for FSx configuration automation.
- Backup and maintenance. Disable automatic daily backups; database storage backup is executed through SnapCenter scheduling.
- Retrieve the SVM management IP address as well as protocol-specific access addresses from SVM details page. Required for FSx configuration automation.

See the following step-by-step procedures for setting up either a primary or standby HA FSx cluster.

1. From the FSx console, click Create File System to start the FSx provision workflow.

2. Select Amazon FSx for NetApp ONTAP. Then click Next.

3. Select Standard Create and, in File System Details, name your file system, Multi-AZ HA. Based on your database workload, choose either Automatic or User-Provisioned IOPS up to 80,000 SSD IOPS. FSx storage comes with up to 2TiB NVMe caching at the backend that can deliver even higher measured IOPS.

File system details

File system name - optional   Info

aws_ora_prod

Maximum of 256 Unicode letters, whitespace, and numbers, plus + - = . _ : /

Deployment type   Info

○ Multi-AZ

○ Single-AZ

SSD storage capacity   Info

1024

Minimum 1024 GiB; Maximum 192 TiB.

Provisioned SSD IOPS

Amazon FSx provides 3 IOPS per GiB of storage capacity. You can also provision additional SSD IOPS as needed.

○ Automatic (3 IOPS per GiB of SSD storage)

○ User-provisioned

40000

Maximum 80,000 IOPS

Throughput capacity   Info

The sustained speed at which the file server hosting your file system can serve data. The file server can also burst to higher speeds for periods of time.

○ Recommended throughput capacity
   128 MB/s

○ Specify throughput capacity

Throughput capacity

512 MB/s   ▼

4. In the Network & Security section, select the VPC, security group, and subnets. These should be created before FSx deployment. Based on the role of the FSx cluster (primary or standby), place the FSx storage nodes in the appropriate zones.

Network & security

Virtual Private Cloud (VPC)   Info
Specify the VPC from which your file system is accessible.

vpc-0474064fc537e5182 ▼

VPC Security Groups   Info
Specify VPC Security Groups to associate with your file system's network interfaces.

Choose VPC security group(s) ▼

sg-08148ca915189ac87 (default)   ✕

Preferred subnet   Info
Specify the preferred subnet for your file system.

subnet-08c952541f4ab282d (us-east-1a) ▼

Standby subnet

subnet-0a84d6eeeb0f4e5c0 (us-east-1b) ▼

VPC route tables
Specify the VPC route tables associated with your file system.

◉ VPC's default route table

◯ Select one or more VPC route tables

Endpoint IP address range
Specify the IP address range in which the endpoints to access your file system will be created

◉ No preference

◯ Select an IP address range

5. In the Security & Encryption section, accept the default, and enter the fsxadmin password.

Security & encryption

Encryption key   Info
AWS Key Management Service (KMS) encryption key that protects your file system data at rest.

aws/fsx (default) ▼

Description	Account	KMS key ID
Default master key that protects my FSx resources when no other key is defined	759995470648	5b31feff-6759-4306-a852-9c99a743982a

File system administrative password
Password for this file system's "fsxadmin" user, which you can use to access the ONTAP CLI or REST API.

◯ Don't specify a password

◉ Specify a password

Password

Confirm password

6. Enter the SVM name and the vsadmin password.

Default storage virtual machine configuration

Storage virtual machine name

fsxora_prod

SVM administrative password
Password for this SVM's "vsadmin" user, which you can use to access the ONTAP CLI or REST API.

○ Don't specify a password

● Specify a password

Password

Confirm password

Active Directory
Joining an Active Directory enables access from Windows and MacOS clients over the SMB protocol.

● Do not join an Active Directory

○ Join an Active Directory

7. Leave the volume configuration blank; you do not need to create a volume at this point.

Default volume configuration

Volume name

```
vol1
```

Maximum of 203 alphanumeric characters, plus _ .

Junction path

```
/vol1
```

The location within your file system where your volume will be mounted.

Volume size

```
1024
```

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency
Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

○ Enabled (recommended)

● Disabled

Capacity pool tiering policy
You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

```
Auto                                        ▼
```

▶ Backup and maintenance - *optional*

▶ Tags - *optional*

Cancel   Back   Next

8. Review the Summary page, and click Create File System to complete FSx file system provision.

Provisioning of database volumes for Oracle database

See Managing FSx for ONTAP volumes - creating a volume for details.

Key considerations:

- Sizing the database volumes appropriately.
- Disabling capacity pool tiering policy for performance configuration.
- Enabling Oracle dNFS for NFS storage volumes.
- Setting up multipath for iSCSI storage volumes.

Create database volume from FSx console

From the AWS FSx console, you can create three volumes for Oracle database file storage: one for the Oracle binary, one for the Oracle data, and one for the Oracle log. Make sure that volume naming matches the Oracle host name (defined in the hosts file in the automation toolkit) for proper identification. In this example, we use db1 as the EC2 Oracle host name instead of a typical IP-address-based host name for an EC2 instance.

Create volume ✕

File system

ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007 ▼

Storage virtual machine

svm-005c6edf027866ca4 | fsx ▼

Volume name

db1_bin

Maximum of 203 alphanumeric characters, plus _ .

Junction path

/db1_bin

The location within your file system where your volume will be mounted.

Volume size

51200

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

⦿ Enabled (recommended)

◯ Disabled

Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

None ▼

Cancel   **Confirm**

Create volume   ✕

File system

| ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007   ▼ |

Storage virtual machine

| svm-005c6edf027866ca4 | fsx   ▼ |

Volume name

| db1_data |

Maximum of 203 alphanumeric characters, plus _ .

Junction path

| /db1_data |

The location within your file system where your volume will be mounted.

Volume size

| 512000 |

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

🔘 Enabled (recommended)

⚪ Disabled

Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

| None   ▼ |

Cancel   **Confirm**

Create volume   ✕

File system

> ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007   ▼

Storage virtual machine

> svm-005c6edf027866ca4 | fsx   ▼

Volume name

> db1_log

Maximum of 203 alphanumeric characters, plus _ .

Junction path

> /db1_log

The location within your file system where your volume will be mounted.

Volume size

> 256000

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

○ Enabled (recommended)

○ Disabled

Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

> None   ▼

Cancel   **Confirm**

ⓘ Creating iSCSI LUNs is not currently supported by the FSx console. For iSCSI LUNs deployment for Oracle, the volumes and LUNs can be created by using automation for ONTAP with the NetApp Automation Toolkit.

Install and configure Oracle on an EC2 instance with FSx database volumes

The NetApp automation team provide an automation kit to run Oracle installation and configuration on EC2 instances according to best practices. The current version of the automation kit supports Oracle 19c on NFS with the default RU patch 19.8. The automation kit can be easily adapted for other RU patches if needed.

Prepare a Ansible controller to run automation

Follow the instruction in the section "Creating and connecting to an EC2 instance for hosting Oracle database" to provision a small EC2 Linux instance to run the Ansible controller. Rather than using RedHat, Amazon Linux t2.large with 2vCPU and 8G RAM should be sufficient.

Retrieve NetApp Oracle deployment automation toolkit

Log into the EC2 Ansible controller instance provisioned from step 1 as ec2-user and from the ec2-user home directory, execute the `git clone` command to clone a copy of the automation code.

```
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

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

Execute automated Oracle 19c deployment using automation toolkit

See these detailed instruction CLI deployment Oracle 19c Database to deploy Oracle 19c with CLI automation. There is a small change in command syntax for playbook execution because you are using an SSH key pair instead of a password for host access authentication. The following list is a high level summary:

1. By default, an EC2 instance uses an SSH key pair for access authentication. From Ansible controller automation root directories `/home/ec2-user/na_oracle19c_deploy`, and `/home/ec2-user/na_rds_fsx_oranfs_config`, make a copy of the SSH key `accesststkey.pem` for the Oracle host deployed in the step "Creating and connecting to an EC2 instance for hosting Oracle database."

2. Log into the EC2 instance DB host as ec2-user, and install the python3 library.

   ```
   sudo yum install python3
   ```

3. Create a 16G swap space from the root disk drive. By default, an EC2 instance creates zero swap space. Follow this AWS documentation: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file?.

4. Return to the Ansible controller (`cd /home/ec2-user/na_rds_fsx_oranfs_config`), and execute the preclone playbook with the appropriate requirements and `linux_config` tags.

   ```
   ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private
   -key accesststkey.pem -e @vars/fsx_vars.yml -t requirements_config
   ```

   ```
   ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private
   -key accesststkey.pem -e @vars/fsx_vars.yml -t linux_config
   ```

5. Switch to the `/home/ec2-user/na_oracle19c_deploy-master` directory, read the README file, and populate the global `vars.yml` file with the relevant global parameters.

6. Populate the `host_name.yml` file with the relevant parameters in the `host_vars` directory.

7. Execute the playbook for Linux, and press Enter when prompted for the vsadmin password.

```
ansible-playbook -i hosts all_playbook.yml -u ec2-user --private-key
accesststkey.pem -t linux_config -e @vars/vars.yml
```

8. Execute the playbook for Oracle, and press enter when prompted for the vsadmin password.

```
ansible-playbook -i hosts all_playbook.yml -u ec2-user --private-key
accesststkey.pem -t oracle_config -e @vars/vars.yml
```

Change the permission bit on the SSH key file to 400 if needed. Change the Oracle host (`ansible_host` in the `host_vars` file) IP address to your EC2 instance public address.

Setting up SnapMirror between primary and standby FSx HA cluster

For high availability and disaster recovery, you can set up SnapMirror replication between the primary and standby FSx storage cluster. Unlike other cloud storage services, FSx enables a user to control and manage storage replication at a desired frequency and replication throughput. It also enables users to test HA/DR without any effect on availability.

The following steps show how to set up replication between a primary and standby FSx storage cluster.

1. Setup primary and standby cluster peering. Log into the primary cluster as the fsxadmin user and execute the following command. This reciprocal create process executes the create command on both the primary cluster and the standby cluster. Replace `standby_cluster_name` with the appropriate name for your environment.

```
cluster peer create -peer-addrs
standby_cluster_name,inter_cluster_ip_address -username fsxadmin
-initial-allowed-vserver-peers *
```

2. Set up vServer peering between the primary and standby cluster. Log into the primary cluster as the vsadmin user and execute the following command. Replace `primary_vserver_name`, `standby_vserver_name`, `standby_cluster_name` with the appropriate names for your environment.

```
vserver peer create -vserver primary_vserver_name -peer-vserver
standby_vserver_name -peer-cluster standby_cluster_name -applications
snapmirror
```

3. Verify that the cluster and vserver peerings are set up correctly.

```
FsxId00164454fac5591e6::> cluster peer show
Peer Cluster Name          Cluster Serial Number Availability   Authentication
------------------------   --------------------- -------------- --------------
FsxId0b6a95149d07aa82e     1-80-000011           Available      ok

FsxId00164454fac5591e6::> vserver peer show
            Peer        Peer                                 Peering         Remote
Vserver     Vserver     State         Peer Cluster           Applications    Vserver
----------- ----------- ------------- ------------------     --------------  ---------
svm_FSxOraSource
            svm_FSxOraTarget
                        peered        FsxId0b6a95149d07aa82e
                                                             snapmirror      svm_FSxOraTarget

FsxId00164454fac5591e6::>
```

4. Create target NFS volumes at the standby FSx cluster for each source volume at the primary FSx cluster. Replace the volume name as appropriate for your environment.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online
-policy default -type DP
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online
-policy default -type DP
```

```
vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online
-policy default -type DP
```

5. You can also create iSCSI volumes and LUNs for the Oracle binary, Oracle data, and the Oracle log if the iSCSI protocol is employed for data access. Leave approximately 10% free space in the volumes for snapshots.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online
-policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_bin/dr_db1_bin_01 -size 45G -ostype linux
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online
-policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_01 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_02 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_03 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_04 -size 100G -ostype
linux
```

vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online -policy default -unix-permissions ---rwxr-xr-x -type RW

```
lun create -path /vol/dr_db1_log/dr_db1_log_01 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_02 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_03 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_04 -size 45G -ostype linux
```

6. For iSCSI LUNs, create mapping for the Oracle host initiator for each LUN, using the binary LUN as an example. Replace the igroup with an appropriate name for your environment, and increment the lun-id for each additional LUN.

```
lun mapping create -path /vol/dr_db1_bin/dr_db1_bin_01 -igroup ip-10-0-
1-136 -lun-id 0
```

```
lun mapping create -path /vol/dr_db1_data/dr_db1_data_01 -igroup ip-10-
0-1-136 -lun-id 1
```

7. Create a SnapMirror relationship between the primary and standby database volumes. Replace the appropriate SVM name for your environment.s

```
snapmirror create -source-path svm_FSxOraSource:db1_bin -destination
-path svm_FSxOraTarget:dr_db1_bin -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

```
snapmirror create -source-path svm_FSxOraSource:db1_data -destination
-path svm_FSxOraTarget:dr_db1_data -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

```
snapmirror create -source-path svm_FSxOraSource:db1_log -destination
-path svm_FSxOraTarget:dr_db1_log -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

This SnapMirror setup can be automated with a NetApp Automation Toolkit for NFS database volumes. The toolkit is available for download from the NetApp public GitHub site.

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

Read the README instructions carefully before attempting setup and failover testing.

> ⓘ Replicating the Oracle binary from the primary to a standby cluster might have Oracle license implications. Contact your Oracle license representative for clarification. The alternative is to have Oracle installed and configured at the time of recovery and failover.

SnapCenter Deployment

SnapCenter installation

Follow Installing the SnapCenter Server to install SnapCenter server. This documentation covers how to install a standalone SnapCenter server. A SaaS version of SnapCenter is in beta review and could be available shortly. Check with your NetApp representative for availability if needed.

Configure SnapCenter plugin for EC2 Oracle host

1. After automated SnapCenter installation, log into SnapCenter as an administrative user for the Window host on which the SnapCenter server is installed.

2. From the left-side menu, click Settings, and then Credential and New to add ec2-user credentials for SnapCenter plugin installation.

3. Reset the ec2-user password and enable password SSH authentication by editing the `/etc/ssh/sshd_config` file on the EC2 instance host.

4. Verify that the "Use sudo privileges" checkbox is selected. You just reset the ec2-user password in the previous step.

5. Add the SnapCenter server name and the IP address to the EC2 instance host file for name resolution.

```
[ec2-user@ip-10-0-0-151 ~]$ sudo vi /etc/hosts
[ec2-user@ip-10-0-0-151 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
10.0.1.233  rdscustomvalsc.rdscustomval.com rdscustomvalsc
```

6. On the SnapCenter server Windows host, add the EC2 instance host IP address to the Windows host file
   `C:\Windows\System32\drivers\etc\hosts`.

```
10.0.0.151        ip-10-0-0-151.ec2.internal
```

7. In the left-side menu, select Hosts > Managed Hosts, and then click Add to add the EC2 instance host to
   SnapCenter.

Check Oracle Database, and, before you submit, click More Options.

Check Skip Preinstall Checks. Confirm Skipping Preinstall Checks, and then click Submit After Save.

You are prompted with Confirm Fingerprint, and then click Confirm and Submit.

After successful plugin configuration, the managed host's overall status show as Running.

Configure backup policy for Oracle database

Refer to this section Setup database backup policy in SnapCenter for details on configuring the Oracle database backup policy.

Generally you need create a policy for the full snapshot Oracle database backup and a policy for the Oracle archive-log-only snapshot backup.

> ⓘ You can enable Oracle archive log pruning in the backup policy to control log-archive space. Check "Update SnapMirror after creating a local Snapshot copy" in "Select secondary replication option" as you need to replicate to a standby location for HA or DR.

Configure Oracle database backup and scheduling

Database backup in SnapCenter is user configurable and can be set up either individually or as a group in a resource group. The backup interval depends on the RTO and RPO objectives. NetApp recommends that you run a full database backup every few hours and archive the log backup at a higher frequency such as 10-15 mins for quick recovery.

Refer to the Oracle section of Implement backup policy to protect database for a detailed step-by-step processes for implementing the backup policy created in the section Configure backup policy for Oracle database and for backup job scheduling.

The following image provides an example of the resources groups that are set up to back up an Oracle database.

Next: Database management.

EC2/FSx Oracle database management

Previous: Deployment procedures.

In addition to the AWS EC2 and FSx management console, the Ansible control node and the SnapCenter UI tool are deployed for database management in this Oracle environment.

An Ansible control node can be used to manage Oracle environment configuration, with parallel updates that keep primary and standby instances in sync for kernel or patch updates. Failover, resync, and failback can be automated with the NetApp Automation Toolkit to archive fast application recovery and availability with Ansible. Some repeatable database management tasks can be executed using a playbook to reduce human errors.

The SnapCenter UI tool can perform database snapshot backup, point-in-time recovery, database cloning, and so on with the SnapCenter plugin for Oracle databases. For more information about Oracle plugin features, see the SnapCenter Plug-in for Oracle Database overview.

The following sections provide details on how key functions of Oracle database management are fulfilled with the SnapCenter UI:

- Database snapshot backups
- Database point-in-time restore
- Database clone creation

Database cloning creates a replica of a primary database on a separate EC2 host for data recovery in the event of logical data error or corruption, and clones can also be used for application testing, debugging, patch validation, and so on.

Taking a snapshot

An EC2/FSx Oracle database is regularly backed up at intervals configured by the user. A user can also take a one-off snapshot backup at any time. This applies to both full-database snapshot backups as well as archive-log-only snapshot backups.

Taking a full database snapshot

A full database snapshot includes all Oracle files, including data files, control files, and archive log files.

1. Log into the SnapCenter UI and click Resources in the left-side menu. From the View dropdown, change to the Resource Group view.

2. Click the full backup resource name, and then click the Backup Now icon to initiate an add-hoc backup.

3. Click Backup and then confirm the backup to start a full database backup.

From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off backup completed successfully. A full database backup creates two snapshots: one for the data volume and one for the log volume.

Taking an archive log snapshot

An archive log snapshot is only taken for the Oracle archive log volume.

1. Log into the SnapCenter UI and click the Resources tab in the left-side menu bar. From the View dropdown, change to the Resource Group view.

2. Click the log backup resource name, and then click the Backup Now icon to initiate an add-hoc backup for archive logs.

3. Click Backup and then confirm the backup to start an archive log backup.

From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off archive log backup completed successfully. An archive log backup creates one snapshot for the log volume.

Restoring to a point in time

SnapCenter-based restore to a point in time is executed on the same EC2 instance host. Complete the following steps to perform the restore:

1. From the SnapCenter Resources tab > Database view, click the database name to open the database backup.
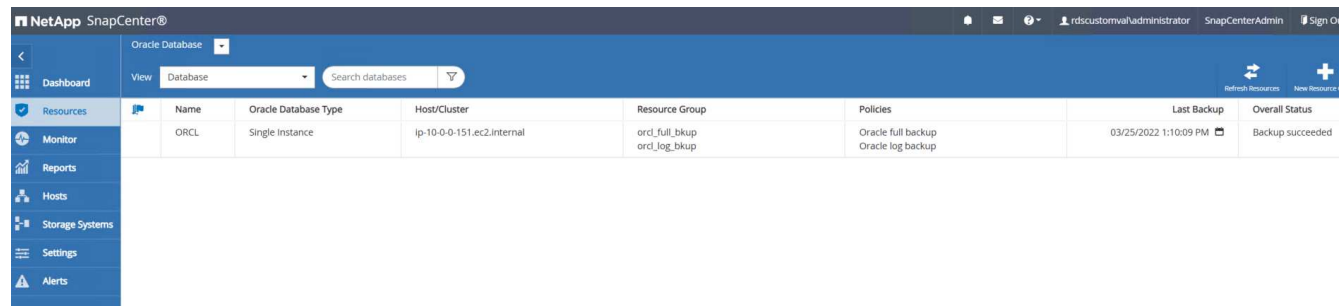
2. Select the database backup copy and the desired point in time to be restored. Also mark down the corresponding SCN number for the point in time. The point-in-time restore can be performed using either the time or the SCN.

3. Highlight the log volume snapshot and click the Mount button to mount the volume.

4. Choose the primary EC2 instance to mount the log volume.

5. Verify that the mount job completes successfully. Also check on the EC2 instance host to see the that log volume mounted and also the mount point path.

6. Copy the archive logs from the mounted log volume to the current archive log directory.

```
[ec2-user@ip-10-0-0-151 ~]$ cp /var/opt/snapcenter/sco/backup_mount/ip-10-0-0-151_03-25-2022_11.15.01.1503_1/ORCL/1/db/ORCL_A/arch/*.arc /ora_nfs_log/db/ORCL_A/arch/
```

7. Return to the SnapCenter Resource tab > database backup page, highlight the data snapshot copy, and click the Restore button to start the database restore workflow.

Backup Name	Count	Type		End Date	Verified	Mounted	RMAN Cataloged	SCN
ip-10-0-0-151_03-25-2022_12.10.01.1097_1	1	Log		03/25/2022 12:10:09 PM	Not Applicable	False	Not Cataloged	1782417
ip-10-0-0-151_03-25-2022_11.55.01.0500_1	1	Log		03/25/2022 11:55:09 AM	Not Applicable	False	Not Cataloged	1781160
ip-10-0-0-151_03-25-2022_11.40.01.0323_1	1	Log		03/25/2022 11:40:09 AM	Not Applicable	False	Not Cataloged	1780268
ip-10-0-0-151_03-25-2022_11.25.01.0430_1	1	Log		03/25/2022 11:25:09 AM	Not Applicable	False	Not Cataloged	1779368
ip-10-0-0-151_03-25-2022_11.15.01.1503_1	1	Log		03/25/2022 11:15:17 AM	Not Applicable	True	Not Cataloged	1778546
ip-10-0-0-151_03-25-2022_11.15.01.1503_0	1	Data		03/25/2022 11:15:11 AM	Unverified	False	Not Cataloged	1778504
ip-10-0-0-151_03-25-2022_11.10.01.1834_1	1	Log		03/25/2022 11:10:09 AM	Not Applicable	False	Not Cataloged	1778184

8. Check "All Datafiles" and "Change database state if needed for restore and recovery", and click Next.

9. Choose a desired recovery scope using either SCN or time. Rather than copying the mounted archive logs

to the current log directory as demonstrated in step 6, the mounted archive log path can be listed in "Specify external archive log files locations" for recovery.

10. Specify an optional prescript to run if necessary.

11. Specify an optional afterscript to run if necessary. Check the open database after recovery.

12. Provide an SMTP server and email address if a job notification is needed.

13. Restore the job summary. Click finish to launch the restore job.

14. Validate the restore from SnapCenter.

Job Details   ✕

Restore 'ip-10-0-0-151.ec2.internal\ORCL'

✔ ▾ Restore 'ip-10-0-0-151.ec2.internal\ORCL'
    ✔ ▾ ip-10-0-0-151.ec2.internal
    ✔ ▸ Prescripts
    ✔ ▸ Pre Restore
    ✔ ▸ Restore
    ✔ ▸ Post Restore
    ✔ ▸ Postscripts
    ✔ ▸ Post Restore Cleanup
    ✔ ▸ Data Collection
    ✔ ▸ Send EMS Messages

ⓘ Task Name: ip-10-0-0-151.ec2.internal Start Time: 03/25/2022 3:33:53 PM End Time: 03/25/2022 3:35:10 PM

View Logs   Cancel Job   Close

15. Validate the restore from the EC2 instance host.

16. To unmount the restore log volume, reverse the steps in step 4.

Creating a database clone

The following section demonstrates how to use the SnapCenter clone workflow to create a database clone from a primary database to a standby EC2 instance.

1. Take a full snapshot backup of the primary database from SnapCenter using the full backup resource group.

2. From the SnapCenter Resource tab > Database view, open the Database Backup Management page for the primary database that the replica is to be created from.

3. Mount the log volume snapshot taken in step 4 to the standby EC2 instance host.

4. Highlight the snapshot copy to be cloned for the replica, and click the Clone button to start the clone procedure.

5. Change the replica copy name so that it is different from the primary database name. Click Next.

6. Change the clone host to the standby EC2 host, accept the default naming, and click Next.

Clone from ORCL

1 Name
2 Locations
3 Credentials
4 PreOps
5 PostOps
6 Notification
7 Summary

Select the host to create a clone

Clone host   ip-10-0-0-47.ec2.internal

Datafile locations

/ora_nfs_data_ORCLREAD   Reset

Control files

/ora_nfs_data_ORCLREAD/ORCLREAD/control/control01.ctl   ✕   +
                                                                              Reset

Redo logs

Group		Size	Unit	Number of files	
⊿ RedoGroup 1	✕	128	MB	1	+
/ora_nfs_data_ORCLREAD/ORCLREAD/redolog/redo04.log				✕	
▸ RedoGroup 2	✕	128	MB	1	+

                                                                              +
                                                                              Reset

Previous   Next

7. Change your Oracle home settings to match those configured for the target Oracle server host, and click Next.

Clone from ORCL

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Database Credentials for the clone

Credential name for sys user: None

Database port: 1521

Oracle Home Settings

Oracle Home: /rdsdbbin/oracle

Oracle OS User: rdsdb

Oracle OS Group: database

Previous | Next

8. Specify a recovery point using either time or the SCN and mounted archive log path.

9. Send the SMTP email settings if needed.

10. Clone the job summary, and click Finish to launch the clone job.

Clone from ORCL

Summary

Clone from backup	ip-10-0-0-151_03-25-2022_17.50.55.0853_0
Clone SID	ORCLREAD
Clone server	ip-10-0-0-47.ec2.internal
Oracle home	/rdsdbbin/oracle
Oracle OS user	rdsdb
Oracle OS group	database
Datafile mountpaths	/ora_nfs_data_ORCLREAD
Control files	/ora_nfs_data_ORCLREAD/ORCLREAD/control/control01.ctl
Redo groups	RedoGroup =1 TotalSize =128 Path =/ora_nfs_data_ORCLREAD/ORCLREAD/redolog/redo04.log
	RedoGroup =2 TotalSize =128 Path =/ora_nfs_data_ORCLREAD/ORCLREAD/redolog/redo03.log
	RedoGroup =3 TotalSize =128 Path =/ora_nfs_data_ORCLREAD/ORCLREAD/redolog/redo02.log
	RedoGroup =4 TotalSize =128 Path =/ora_nfs_data_ORCLREAD/ORCLREAD/redolog/redo01.log
Recovery scope	Until SCN 1788879
Prescript full path	none
Prescript arguments	
Postscript full path	none
Postscript arguments	
Send email	No

Steps: 1 Name, 2 Locations, 3 Credentials, 4 PreOps, 5 PostOps, 6 Notification, 7 Summary

[Previous] [Finish]

11. Validate the replica clone by reviewing the clone job log.

Job Details

Clone from backup 'ip-10-0-0-151_03-25-2022_17.50.55.0853_0'

✓ ▼ Clone from backup 'ip-10-0-0-151_03-25-2022_17.50.55.0853_0'

  ✓ ▼ ip-10-0-0-47.ec2.internal

    ✓ ▶ Prescripts

    ✓ ▶ Query Host Information

    ✓ ▶ Prepare for Cloning

    ✓ ▶ Cloning Resources

    ✓ ▶ FileSystem Clone

    ✓ ▶ Application Clone

    ✓ ▶ Postscripts

    ✓ ▶ Register Clone

    ✓ ▶ Unmount Clone

    ✓ ▶ Data Collection

    ✓ ▶ Send EMS Messages

ⓘ Task Name: ip-10-0-0-47.ec2.internal Start Time: 03/25/2022 9:08:32 PM End Time: 03/25/2022 9:12:03 PM

View Logs   Cancel Job   Close

The cloned database is registered in SnapCenter immediately.

12. Turn off Oracle archive log mode. Log into the EC2 instance as oracle user and execute following command:

```
sqlplus / as sysdba
```

```
shutdown immediate;
```

```
startup mount;
```

```
alter database noarchivelog;
```

```
alter database open;
```

> (i) Instead primary Oracle backup copies, a clone can also be created from replicated secondary backup copies on target FSx cluster with same procedures.

HA failover to standby and resync

The standby Oracle HA cluster provides high availability in the event of failure in the primary site, either in the compute layer or in the storage layer. One significant benefit of the solution is that a user can test and validate the infrastructure at any time or with any frequency. Failover can be user simulated or triggered by real failure. The failover processes are identical and can be automated for fast application recovery.

See the following list of failover procedures:

1. For a simulated failover, run a log snapshot backup to flush the latest transactions to the standby site, as demonstrated in the section Taking an archive log snapshot. For a failover triggered by an actual failure, the last recoverable data is replicated to the standby site with the last successful scheduled log volume backup.

2. Break the SnapMirror between primary and standby FSx cluster.

3. Mount the replicated standby database volumes at the standby EC2 instance host.

4. Relink the Oracle binary if the replicated Oracle binary is used for Oracle recovery.

5. Recover the standby Oracle database to the last available archive log.

6. Open the standby Oracle database for application and user access.

7. For an actual primary site failure, the standby Oracle database now takes the role of the new primary site and database volumes can be used to rebuild the failed primary site as a new standby site with the reverse SnapMirror method.

8. For a simulated primary site failure for testing or validation, shut down the standby Oracle database after the completion of testing exercises. Then unmount the standby database volumes from the standby EC2 instance host and resync replication from the primary site to the standby site.

These procedures can be performed with the NetApp Automation Toolkit available for download at the public NetApp GitHub site.

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

Read the README instruction carefully before attempting setup and failover testing.

Database migration from on-prem to public cloud

Database migration is a challenging endeavor by any means. Migrating an Oracle database from on-premises to the cloud is no exception.

The following sections provide key factors to consider when migrating Oracle databases to the AWS public cloud with the AWS EC2 compute and FSx storage platform.

ONTAP storage is available on-premises

If the on-premises Oracle database is sitting on an ONTAP storage array, then it is easier to set up replication for database migration using the NetApp SnapCenter UI tool.

1. Build a target compute EC2 instance that matches the on-premises instance.
2. Provision matching, equally sized database volumes from FSx console.
3. Mount the FSx database volumes to the EC2 instance.
4. Set up SnapMirror replication between the on-premises database volumes to the target FSx database volumes. The initial sync might take some time to move the primary source data, but any following incremental updates are much quicker.
5. At the time of switchover, shut down the primary application to stop all transactions. From SnapCenter, run a log backup to flush the remaining transactions to the target.
6. Break up the mirrored volumes, run Oracle recovery at the target, and bring up the database for service.
7. Point applications to the Oracle database in the cloud.

ONTAP storage is not available on premises

If the on-premises Oracle database is hosted on third-party storage other than ONTAP, database migration is based on the restore of a Oracle database backup copy. You must play the archive log to make it current before switching over.

AWS S3 can be used as a staging storage area for database move and migration. See the following high level steps for this method:

1. Provision a new, matching EC2 instance that is comparable with the on-premises instance.
2. Provision equal database volumes from FSx storage and mount the volumes to the EC2 instance.
3. Create a disk-level Oracle backup copy.

4. Move the backup copy to AWS S3 storage.

5. Recreate the Oracle control file and restore and recover the database by pulling data and the archive log from S3 storage.

6. Sync the target Oracle database with the on-premises source database.

7. At switchover, shut down the application and source Oracle database. Copy the last few archive logs and apply them to the target Oracle database to bring it up to date.

8. Start up the target database for user access.

9. Redirect application to the target database to complete the switchover.

Consolidate Oracle databases in AWS with Oracle multitenancy CDB/PDB architecture

1. Create CDB in the AWS public cloud.

2. If the on-premises database is also deployed in CDB/PDB multitenancy, unplug the PDB to be migrated.

3. Transfer metadata as well as underlined Oracle data files to the target CDB instance.

4. Validate compatibility with Oracle validation procedures.

5. If compatibility validation passes, plug the unplugged PDB into the target CDB container.

6. Update the data dictionary if required.

7. Back up and open the migrated PDB for access.

> (i) PDB unplug and plug-in requires application downtime that should be taken into consideration during migration planning.

Again, the NetApp automation team provides a migration toolkit that can facilitate Oracle database migration from on-premises to the AWS cloud. Check the NetApp public GitHub site for the latest database migration tools.