# Table of Contents

Oracle Database Data Protection ............................................. 1
Solution Overview ............................................................ 1
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**
<table>
<thead>
<tr>
<th>Environment</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible environment</td>
<td>AWX/Tower</td>
</tr>
<tr>
<td></td>
<td>Ansible v.2.10 and higher</td>
</tr>
<tr>
<td></td>
<td>Python 3</td>
</tr>
<tr>
<td></td>
<td>Python libraries</td>
</tr>
<tr>
<td></td>
<td>- netapp-lib</td>
</tr>
<tr>
<td></td>
<td>- xmltodict</td>
</tr>
<tr>
<td></td>
<td>- jmespath</td>
</tr>
<tr>
<td>ONTAP</td>
<td>ONTAP version 9.8 +</td>
</tr>
<tr>
<td></td>
<td>Two data aggregates</td>
</tr>
<tr>
<td></td>
<td>NFS vlan and ifgrp created</td>
</tr>
<tr>
<td>Oracle server(s)</td>
<td>RHEL 7/8</td>
</tr>
<tr>
<td></td>
<td>Oracle Linux 7/8</td>
</tr>
<tr>
<td></td>
<td>Network interfaces for NFS, public,</td>
</tr>
<tr>
<td></td>
<td>and optional mgmt</td>
</tr>
<tr>
<td></td>
<td>Existing Oracle environment on</td>
</tr>
<tr>
<td></td>
<td>source, and the equivalent Linux</td>
</tr>
<tr>
<td></td>
<td>operating system at the destination (DR Site or Public Cloud)</td>
</tr>
</tbody>
</table>

**CVO**

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

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.

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

**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.
<table>
<thead>
<tr>
<th>Playbook</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| 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

AWX/Tower Oracle Data Protection

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:

   ```json
   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:

   ```yaml
   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:
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

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.

**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.

   ![info](https://via.placeholder.com/15) The items in blue must be changed to match your environment.
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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">ontap</span>

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

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

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

source_intercluster_network_port_details:
  using_dedicated_ports: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">yes</span>
  using_ifgrp: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">yes</span>
  using_vlans: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">yes</span>
  failover_for_shared_individual_ports: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">yes</span>

  ifgrp_name: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">a0a</span>
  vlan_id: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">10</span>

  ports:
    - <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">e0b</span>
    - <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">e0g</span>

  broadcast_domain: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">NFS</span>
  ipspace: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">Default</span>

  failover_group_name: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;">iclifs</span>

source_intercluster_lif_details:
- name: <i>icl_1</i>
  address: <i>10.0.0.1</i>
  netmask: <i>255.255.255.0</i>
  home_port: <i>a0a-10</i>
  node: <i>AFF-01</i>

- name: <i>icl_2</i>
  address: <i>10.0.0.2</i>
  netmask: <i>255.255.255.0</i>
  home_port: <i>a0a-10</i>
  node: <i>AFF-02</i>

# 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: <i>yes</i>

destination_intercluster_network_port_details:
  using_dedicated_ports: <i>yes</i>
  using_ifgrp: <i>no</i>
<i>yes</i>}

failover_for_shared_individual_ports: <i>yes</i>

ifgrp_name: <i>a0a</i>

vlan_id: <i>10</i>

ports:
  - <i>e0b</i>
  - <i>e0g</i>

broadcast_domain: <i>NFS</i>

ipspace: <i>Default</i>

failover_group_name: <i>iclifs</i>

destination_intercluster_lif_details:
  - name: <i>icl_1</i>
    address: <i>10.0.0.3</i>
    netmask: <i>255.255.255.0</i>
    home_port: <i>a0a-
node: DR-AFF-01
- name: icl_2
  address: 10.0.0.4
  netmask: 255.255.255.0
  home_port: a0a-10
node: DR-AFF-02

# Variables for SnapMirror Peering
# src_lif: Will be retrieve through Ansible Task
dst_lif: Will be retrieve through Ansible Task
passphrase: your-passphrase

dst_cluster_name: dst-cluster-name

dst_cluster_ip: dst_cluster_ip: dst_cluster_name

# Please Enter Destination Cluster

# Please Enter Destination SVM to create mirror relationship
dst_vserver: 

# Please Enter NFS Lif for dst vserver
dst_nfs_lif: 

# Please Enter Source Cluster Name
src_cluster_name: 

# Please Enter Source Cluster
src_cluster_ip: 

# Please Enter Source SVM
src_vserver: 

# Variable for Oracle Volumes and SnapMirror Details
# Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: 

# Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - 

# Please Enter Source Database Volume(s)

src_db_vols:

- <span>db_vol</span>

# Please Enter Source Archive Volume(s)

src_archivelog_vols:

- <span>log_vol</span>

# Please Enter Destination Snapmirror Policy

snapmirror_policy: <span>async_policy_oracle</span>

# Export Policy Details

# Enter the destination export policy details

export_policy_details:

  name: <span>nfs_export_policy</span>

  client_match: <span>0.0.0.0/0</span>

  ro_rule: sys
### Linux env specific config variables ###

**NFS Mount points for Oracle DB volumes**

```
mount_points:
  - /u01
  - /u02
  - /u03
```

# 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: 1234
```

# RedHat subscription username and password

```
redhat_sub_username: xxx
redhat_sub_password: xxx
```

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

# Recovery Type (leave as scn)

```
recovery_type: scn
```

# Oracle Control Files

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

</div></code></pre></div></div>

```javascript
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";
}
```
```javascript
function binaryvolsdropdown() {
    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 addbinaryvols() {
    var y = document.getElementById("number_of_extra_binary_vols").value;
    var j=0;
    var myHTML = ''; 
    var wrapper = document.getElementById("select_more_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;">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;">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"; 
}

CVO

<style>
div {
    position: relative;
}
div button {
    position: absolute;
top: 0;
right: 0;
}
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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">ontap</span></i></span>

# CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA
SIGNED CERTIFICATES)

can_signed_certs: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">false</span></span>

# Names of the Nodes in the Source ONTAP Cluster

src_nodes:
- <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">AFF-01</span>
- <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">AFF-02</span>

# Names of the Nodes in the Destination CVO Cluster

dst_nodes:
- <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">DR-AFF-01</span>
- <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">DR-AFF-02</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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">yes</span></span>

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

source_interclustering_lif_details:
- name: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>icl_1</span>
  address: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>10.0.0.1</span>
  netmask: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>255.255.255.0</span>
  home_port: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>a0a-10</span>
node: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">AFF-01</span>

- name: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">icl_2</span>
  address: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">10.0.0.2</span>
  netmask: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">255.255.255.0</span>
  home_port: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">a0a-10</span>

node: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">AFF-02</span>

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

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

# Region where your CVO will be deployed.
region_deploy: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">us-east-1</span>

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

# AWS Managed Policy required to give permission for IAM role creation.

# Specify your aws role name, a new role is created if one already does not exist.
aws_role_name: <span style="color:#04EF; font-weight:bold; font-style:italic; text-decoration:underline;">arn:aws:iam::1234567:policy/OCCM</span>
# Name your connector.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">awx_connector</span>

# Name of the key pair generated in AWS.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">key_pair</span>

# Name of the Subnet that has the range of IP addresses in your VPC.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">subnet-12345</span>

# ID of your AWS security group that allows access to on-prem resources.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">sg-123123123</span>

# You Cloud Manager Account ID.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">account-A23123A</span>

# Name of the your CVO instance

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">test_cvo</span>

# ID of the VPC in AWS.

type: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">vpc-123123123</span>

# For Federated users, Client ID from API Authentication Section of Cloud Central to generate access token.

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

```bash
community_access: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"
</i>pass</i></span>
```

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

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

# Please Enter Destination SVM to create mirror relationship
dst_vserver: <span style="color:#004EFF; font-weight:bold; font-style:italic; 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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"
# Please Enter Source Cluster Name
src_cluster_name: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>

# Please Enter Source Cluster
src_cluster_ip: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>

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

# Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>

# Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>
    <a id="more_binary_vols1" href="javascript:binaryvols1dropdown();">More Binary Vols</a></span><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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/>
More Database Vols</a></div><a id="more_database_vols1_button" href="javascript:adddatabasevols1();">Enter Volume details</a></div><a id="extra_database_vols1" href="javascript:adddatabasevols1();" class="extra"><div id="select_more_database_vols1"></div></a>

#Please Enter Source Archive Volume(s)

src_archivelog_vols:
  - <span><span><div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/>(log_vol)</div></span>
    <a id="more_log_vols1" href="javascript:logvols1dropdown();">More Log Vols</a></div><a id="more_log_vols1_button" href="javascript:addlogvols1();">Enter Volume details</a></div><a id="extra_log_vols1" href="javascript:adddatabasevols1();" class="extra"><div id="select_more_log_vols1"></div></a>

#Please Enter Destination Snapmirror Policy

snapmirror_policy: <span><span><div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/>(async_policy_oracle)</div></span>
  <div id="select_more_log_vols1"></div></a>

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

### Linux env specific config variables ###

#NFS Mount points for Oracle DB volumes

mount_points:
# 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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">1234</span>

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

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

#Oracle Control Files
control_files:
  - <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">/u02/oradata/CDB2/control01.ctl</span>
  - <span style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;">/u03/orareco/CDB2/control02.ctl</span>
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"
/
>\<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 databasevol1dropdown() {
    document.getElementById("more_database_vols1").style.display = "none";
}

function databasevols1dropdown() {
    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 style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/><i>db_vol</i><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) {

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

**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.

[Info] 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

**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**

**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**

**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.

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 parallel 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 software 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