



## **Oracle Database**

### **NetApp Solutions**

NetApp  
May 09, 2022

This PDF was generated from <https://docs.netapp.com/us-en/netapp-solutionshttps://www.netapp.com/pdf.html?item=/media/25782-nva-1155.pdf> on May 09, 2022. Always check docs.netapp.com for the latest.

# Table of Contents

- Oracle Database ..... 1
  - Deploying Oracle Database ..... 1
  - Oracle Database Data Protection ..... 10

# 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](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](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](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 it 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 - xmlltodict - 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](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.



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

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/host\_vars.adoc[]

- 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.



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

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/vars.adoc[]

2. Fill in all variables in blue fields.
3. 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.



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.

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/validation.adoc[]

## 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.

Unresolved directive in ent-apps-db/cli\_automation.adoc - include::ent-apps-db/host\_vars.adoc[]

### 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.

Unresolved directive in ent-apps-db/cli\_automation.adoc - include::ent-apps-db/vars.adoc[]

### 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](#).

Unresolved directive in ent-apps-db/cli\_automation.adoc - include::ent-apps-db/validation.adoc[]

## 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
<b>Ansible environment</b>	AWX/Tower
	Ansible v.2.10 and higher
	Python 3
	Python libraries - netapp-lib - xmlltodict - jmespath
<b>ONTAP</b>	ONTAP version 9.8 +
	Two data aggregates
	NFS vlan and ifgrp created
<b>Oracle server(s)</b>	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
<b>Ansible environment</b>	AWX/Tower
	Ansible v.2.10 and higher
	Python 3
	Python libraries - netapp-lib - xmlltodict - jmespath
<b>ONTAP</b>	ONTAP version 9.8 +
	Two data aggregates
	NFS vlan and ifgrp created
<b>Oracle server(s)</b>	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
<b>ontap_setup</b>	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
<b>ora_replication_cg</b>	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
<b>ora_replication_log</b>	Switch current log for each database in /etc/oratab
	Snapshot taken of Oracle Log volume
	Snapmirror Updated
<b>ora_recovery</b>	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
<b>cvo_setup</b>	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
<b>ora_replication_cg</b>	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
<b>ora_replication_log</b>	Switch current log for each database in /etc/oratab
	Snapshot taken of Oracle Log volume
	Snapmirror Updated
<b>ora_recovery</b>	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

#### 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

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_onprem\_creds.adoc[]

##### CVO

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_cvo\_creds.adoc[]

#### 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](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.



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

#### On-Prem

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_onprem\_vars.adoc[]

#### CVO

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_cvo\_vars.adoc[]

### 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

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_ontap\_cvo\_setup.adoc[]

#### Replication For Binary and Database Volumes

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_db\_replication.adoc[]

#### Replication for Log Volumes

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_log\_replication.adoc[]

#### Restore and Recover Database

Unresolved directive in ent-apps-db/db\_protection\_awx\_automation.adoc - include::ent-apps-db/db\_protection\_restore\_recovery.adoc[]

### 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 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

## Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.