Oracle Database Deployment on AWS EC2 and FSx Best Practices

NetApp Solutions

NetApp
July 31, 2024

Table of Contents

Oracle Database Deployment on AWS EC2 and FSx Best Practices

WP-7357: Oracle Database Deployment on EC2 and 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.

To learn more about the solution and use case, take a look at following overview video:

Modernize your Oracle database with hybrid cloud in AWS and FSx ONTAP, Part1 - Use case and solution architecture

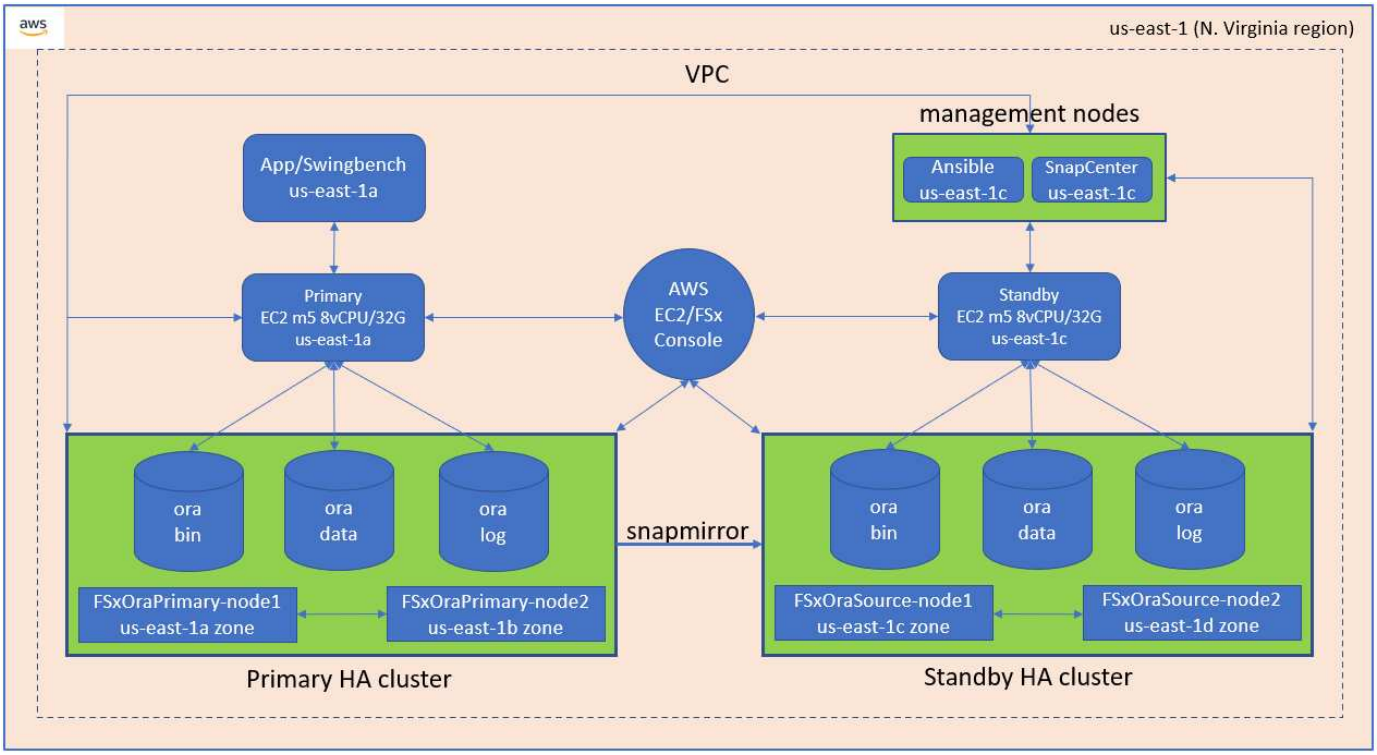
Solution architecture

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 of 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 tool kits. 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

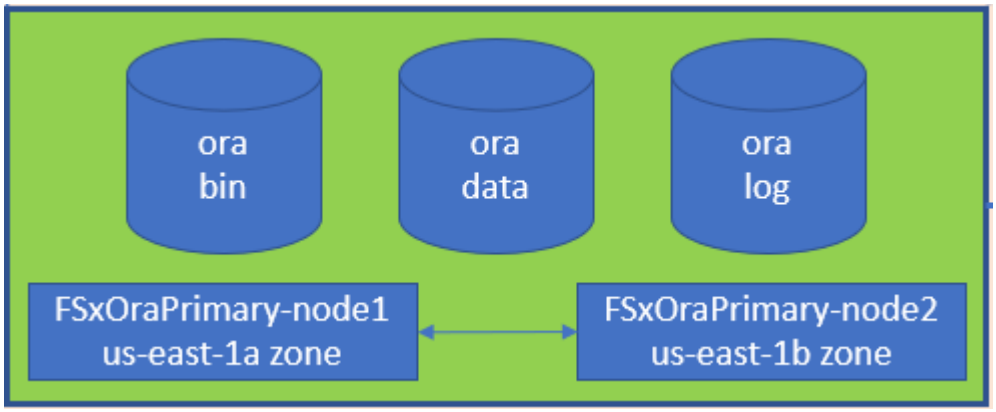
Selecting the right VM size is important for optimal performance of a relational database in a public cloud. For better performance, NetApp recommends using an EC2 M5 Series instance for Oracle deployment, which is optimized for database workloads. The same instance type is also used to power a RDS instance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

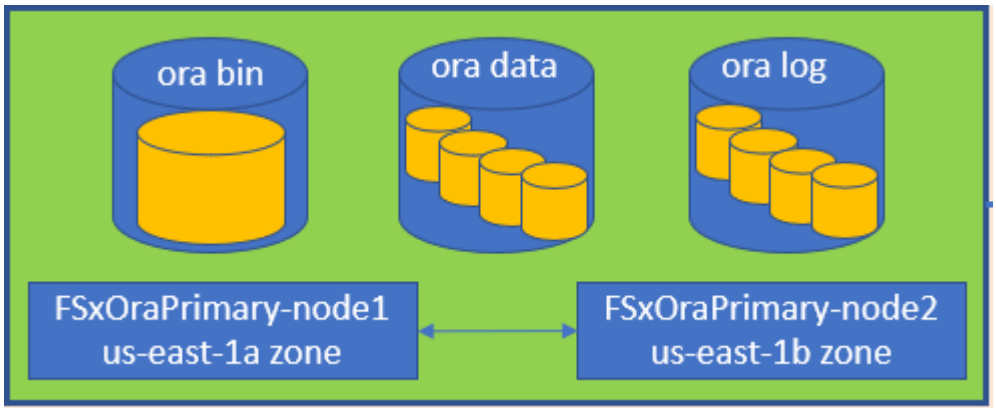
Storage layout and settings

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. dNFS is supported with NFSv3 with all versions of Oracle. NFSv4 is supported with all OS's that follow the NFSv4 standard. dNFS support for NFSv4 requires Oracle 12.1.0.2 or higher. NFSv4.1 requires specific OS support. Consult the NetApp Interoperability Matrix Tool (IMT) for supported OS's. dNFS support for NFSv4.1 requires Oracle version 19.3.0.0 or higher.

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. The NetApp Support matrix for NFSv3 and NFSv4 do not include specific operating systems. All OSs that obey the RFC are supported. When searching the online IMT for NFSv3 or NFSv4 support, do not select a specific OS because no matches will be displayed. All OSs are implicitly supported by the general policy.

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 and FSx

This section describes the deployment procedures of deploying Oracle RDS custom database with FSx storage.

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:

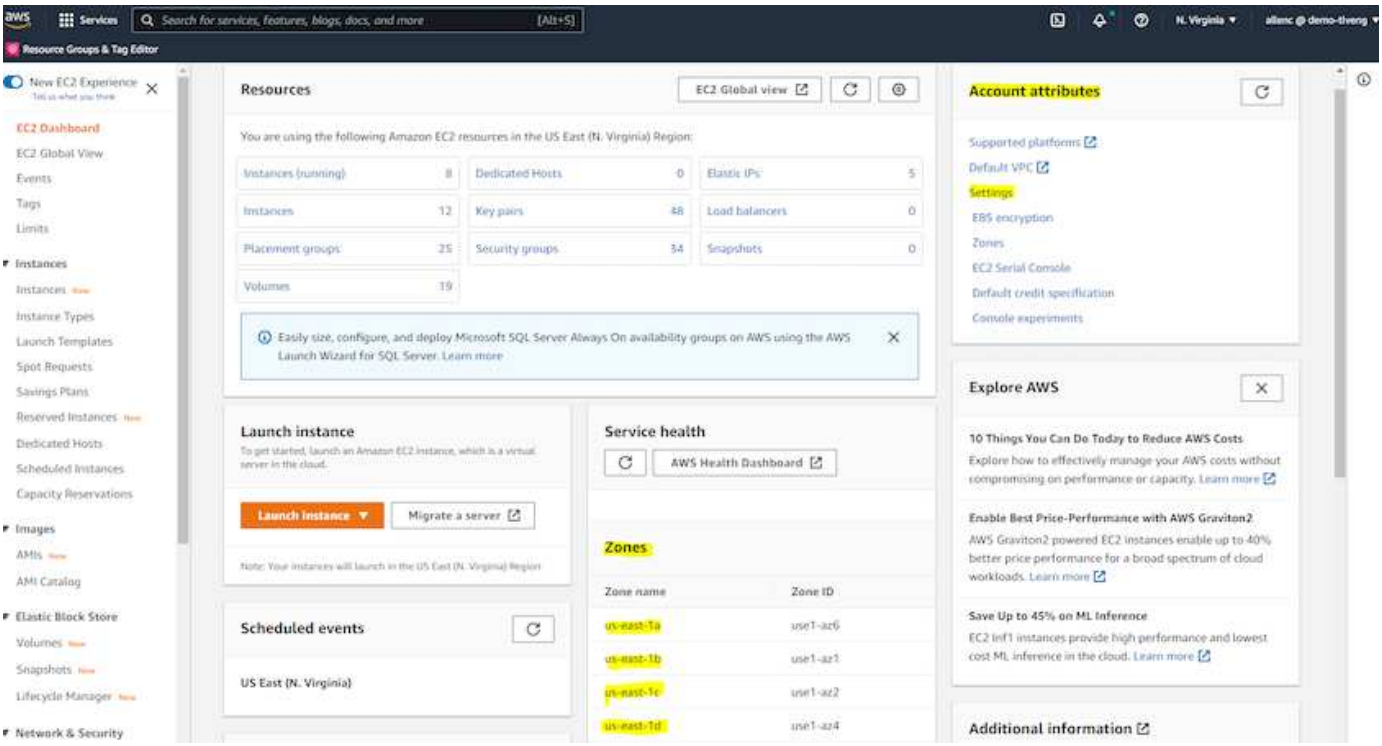
- Set up to use Amazon EC2

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

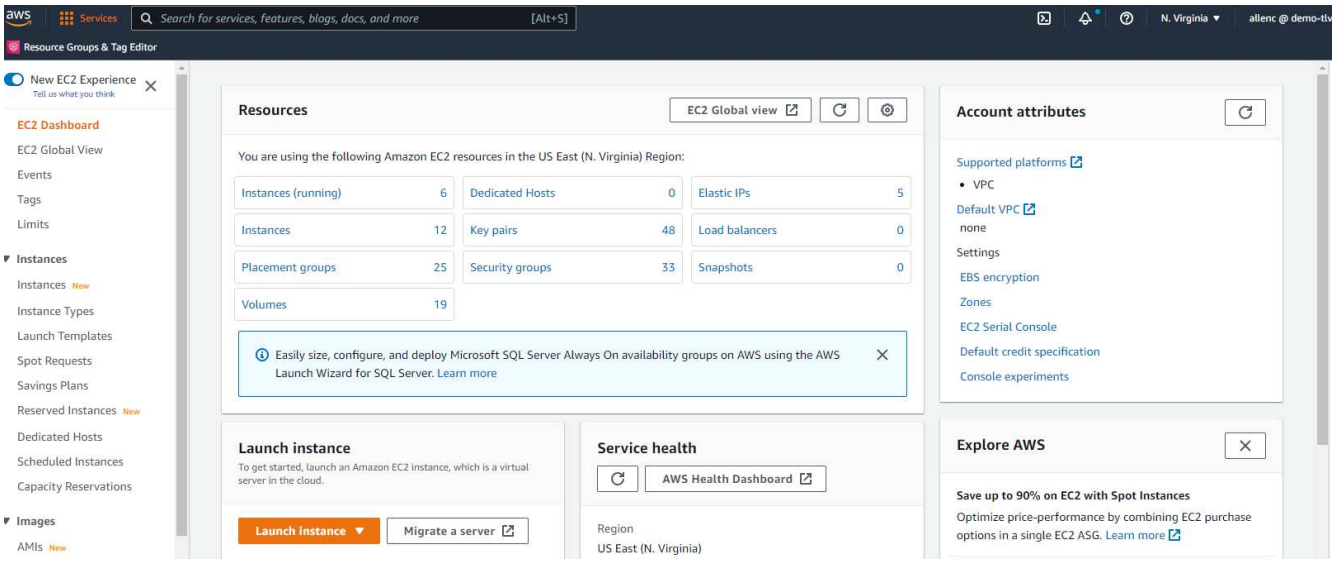
See the tutorial Get started with Amazon EC2 Linux instances for step-by-step deployment procedures and best practices.

Key topics:

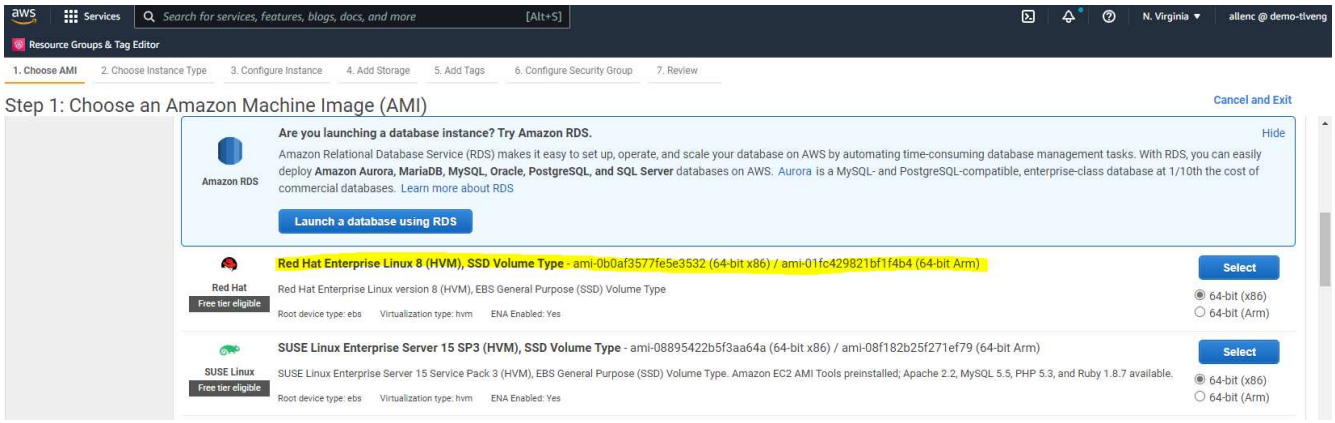
- Overview.
- Prerequisites.
- Step 1: Launch an instance.
- Step 2: Connect to your instance.
- Step 3: Clean up your instance.

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

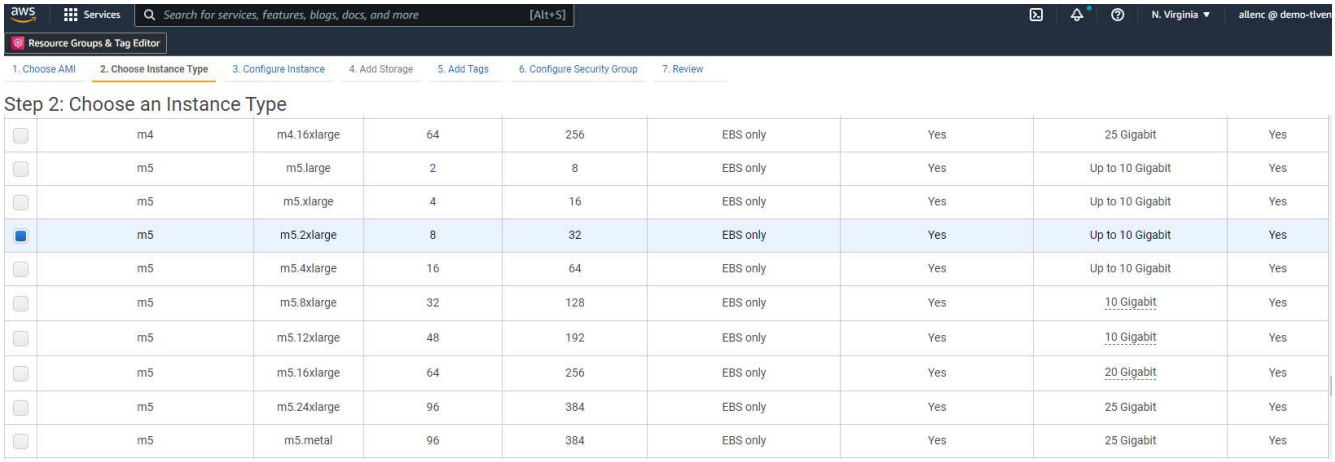
1. From the EC2 dashboard, click the yellow Launch Instance button to start the EC2 instance deployment workflow.



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

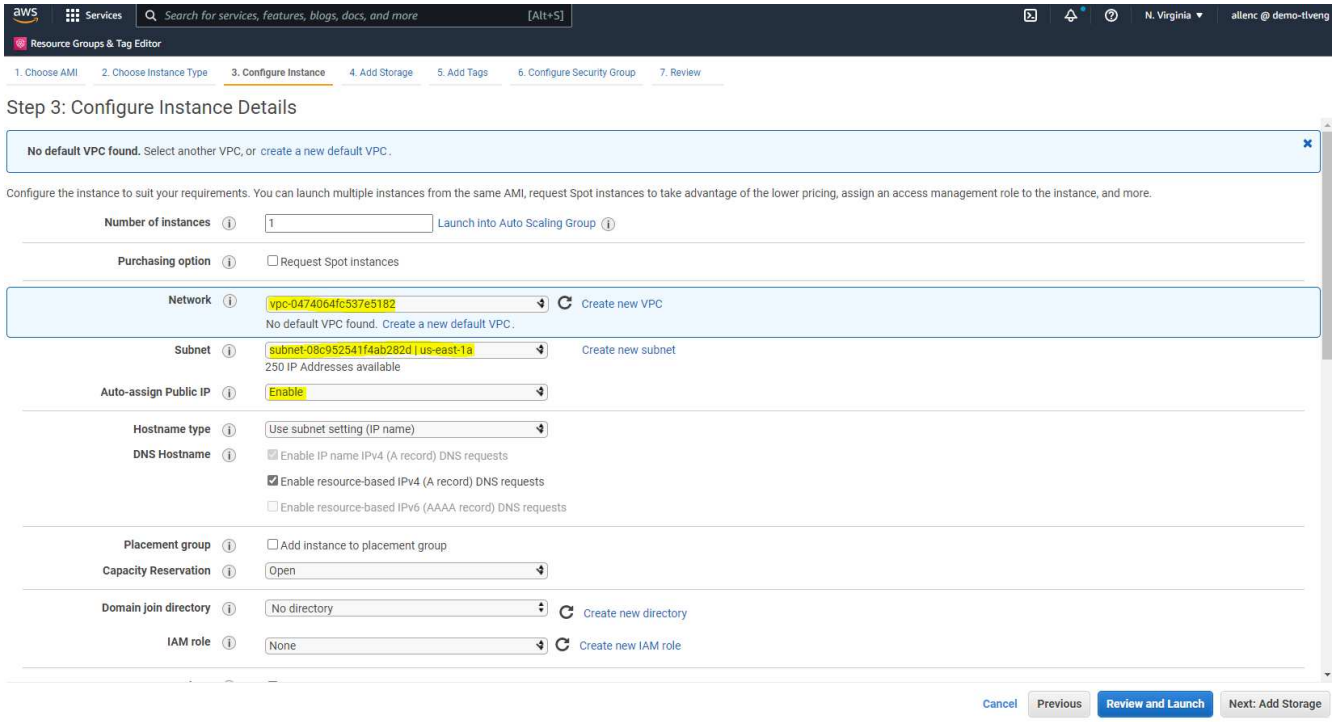


3. In Step 2, select an m5 instance type with the appropriate CPU and memory allocation based on your Oracle database workload. Click "Next: Configure Instance Details."

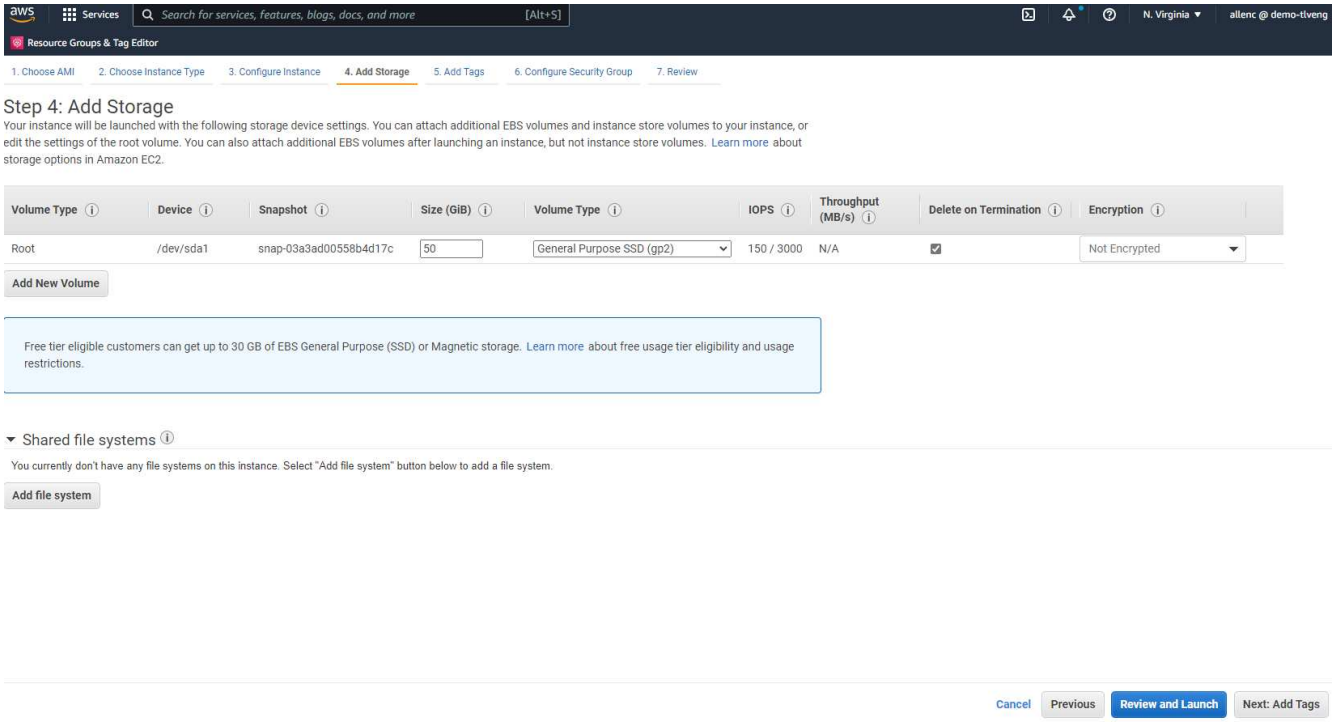


☐	m4	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes
☐	m5	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
☐	m5	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes
☑	m5	m5.2xlarge	8	32	EBS only	Yes	Up to 10 Gigabit	Yes
☐	m5	m5.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
☐	m5	m5.8xlarge	32	128	EBS only	Yes	10 Gigabit	Yes
☐	m5	m5.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes
☐	m5	m5.16xlarge	64	256	EBS only	Yes	20 Gigabit	Yes
☐	m5	m5.24xlarge	96	384	EBS only	Yes	25 Gigabit	Yes
☐	m5	m5.metal	96	384	EBS only	Yes	25 Gigabit	Yes

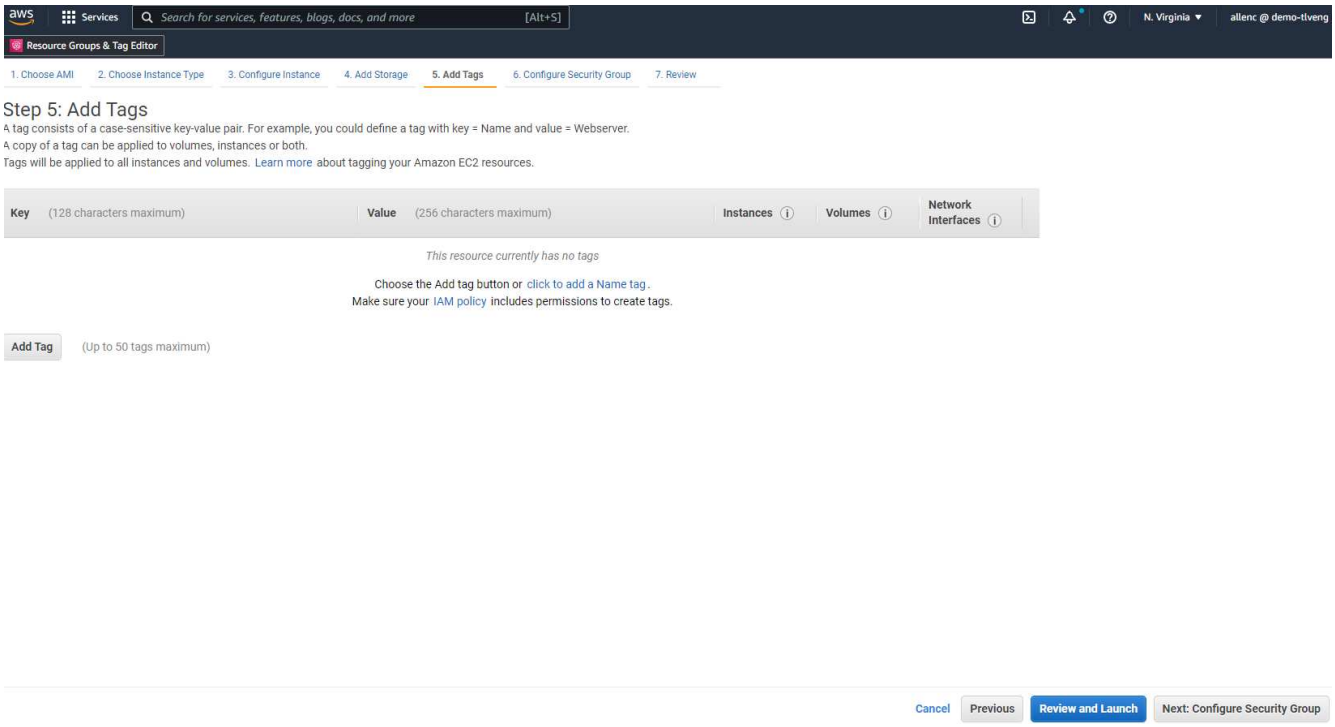
4. In Step 3, choose the VPC and subnet where the instance should be placed and enable public IP assignment. Click "Next: Add Storage."



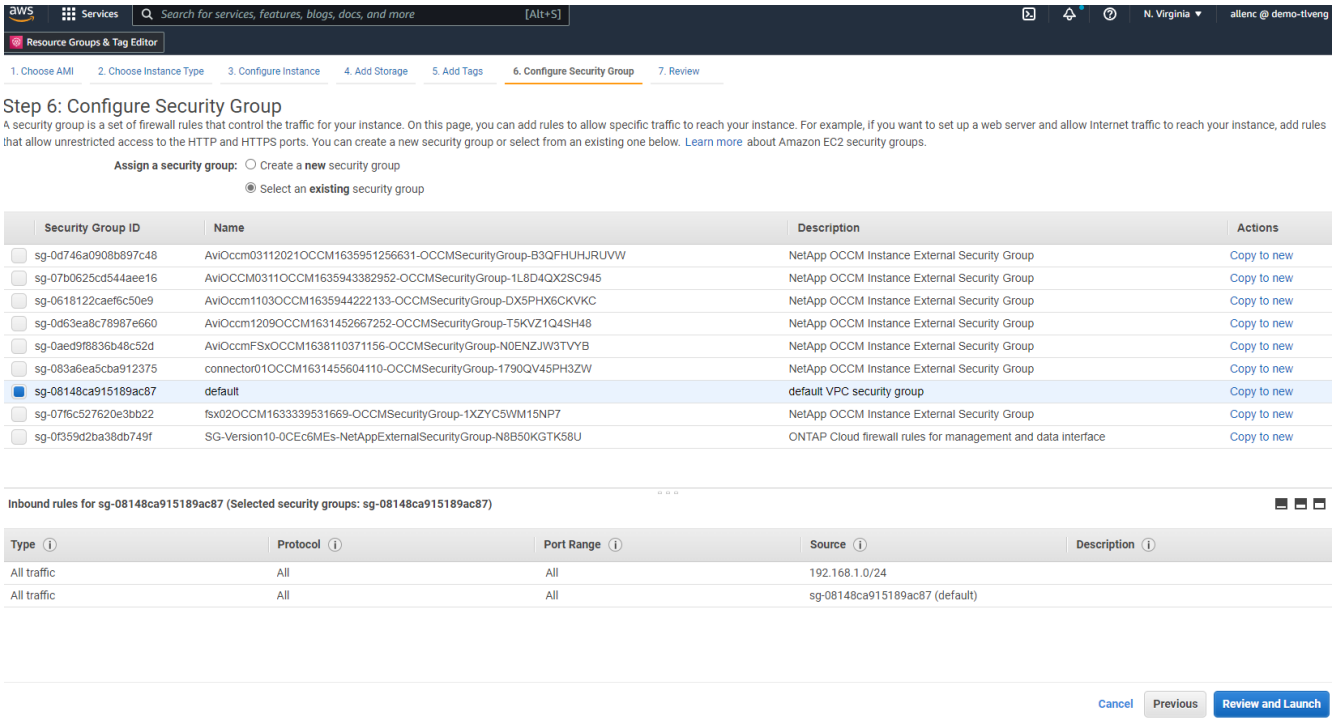
5. In Step 4, allocate enough space for the root disk. You may need the space to add a swap. By default, EC2 instance assign zero swap space, which is not optimal for running Oracle.



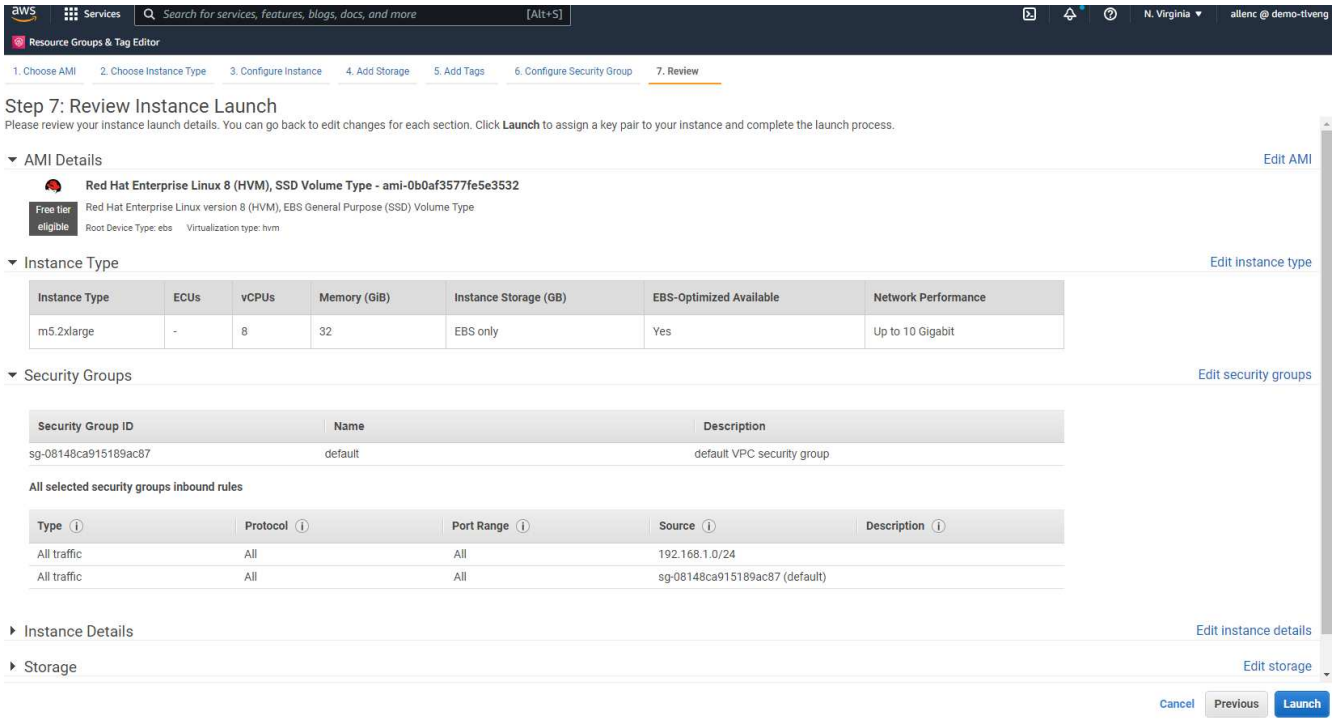
6. In Step 5, add a tag for instance identification if needed.

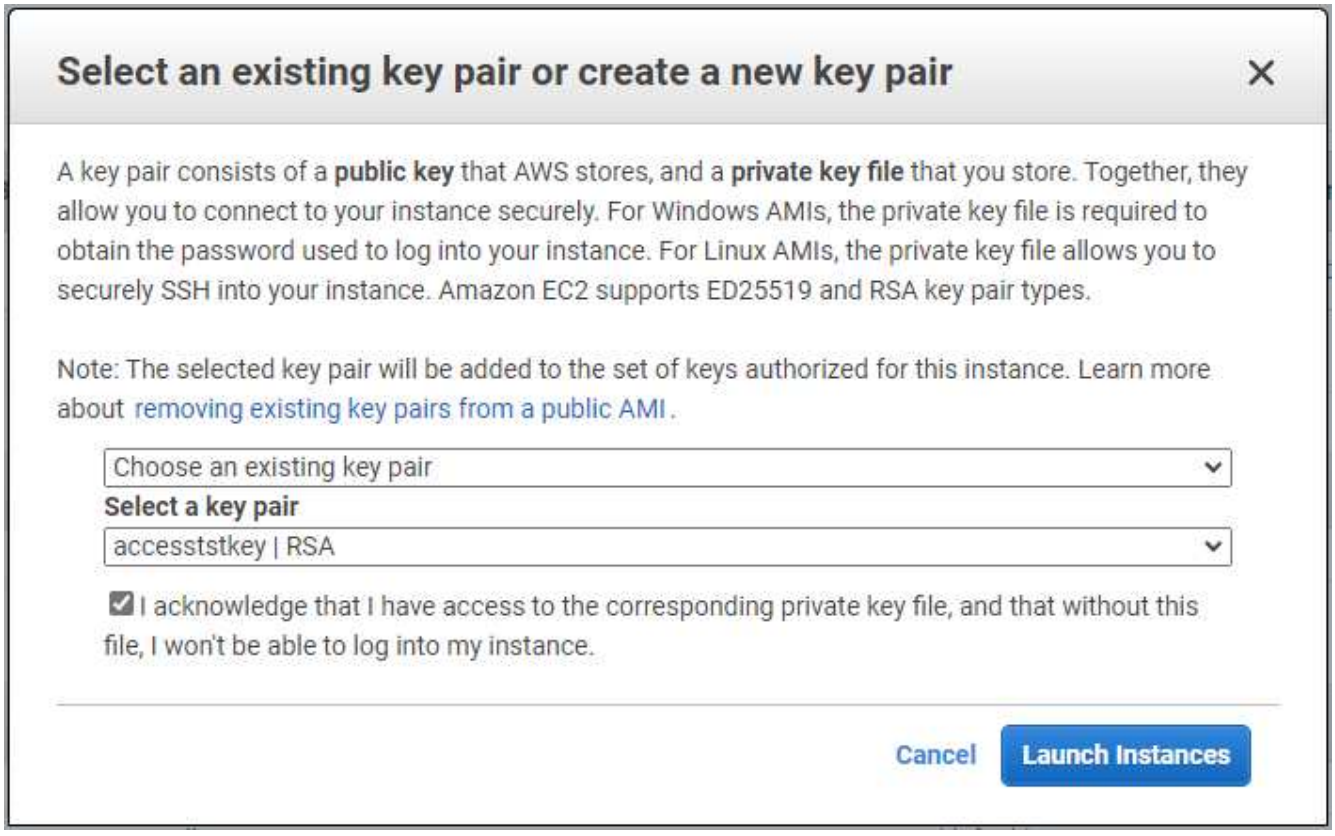


7. In Step 6, select an existing security group or create a new one with the desired inbound and outbound policy for the instance.



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

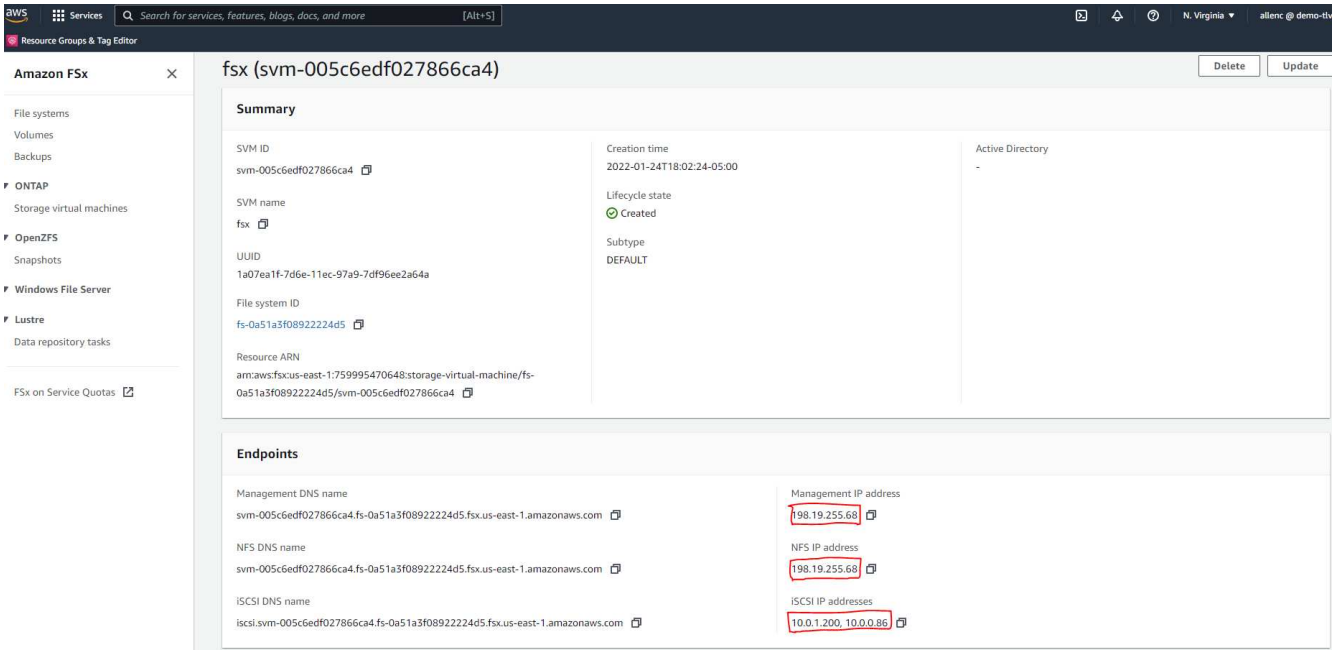
EC2 instance deployment allocates an EBS root volume for the OS. FSx for ONTAP file systems provides Oracle database storage volumes, including the Oracle binary, data, and log volumes. The FSx storage NFS volumes can be either provisioned from the AWS FSx console or from Oracle installation, and configuration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

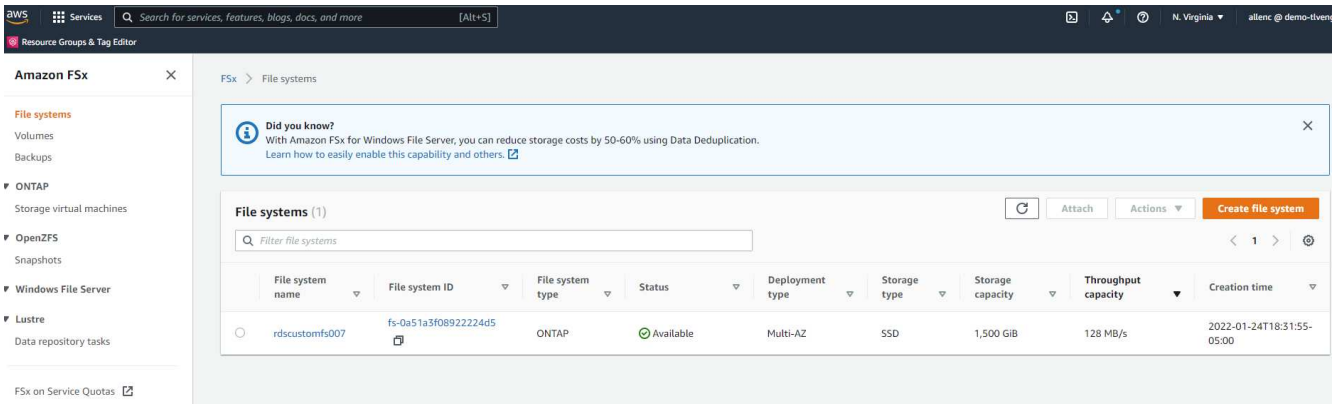
Key considerations:

- 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.

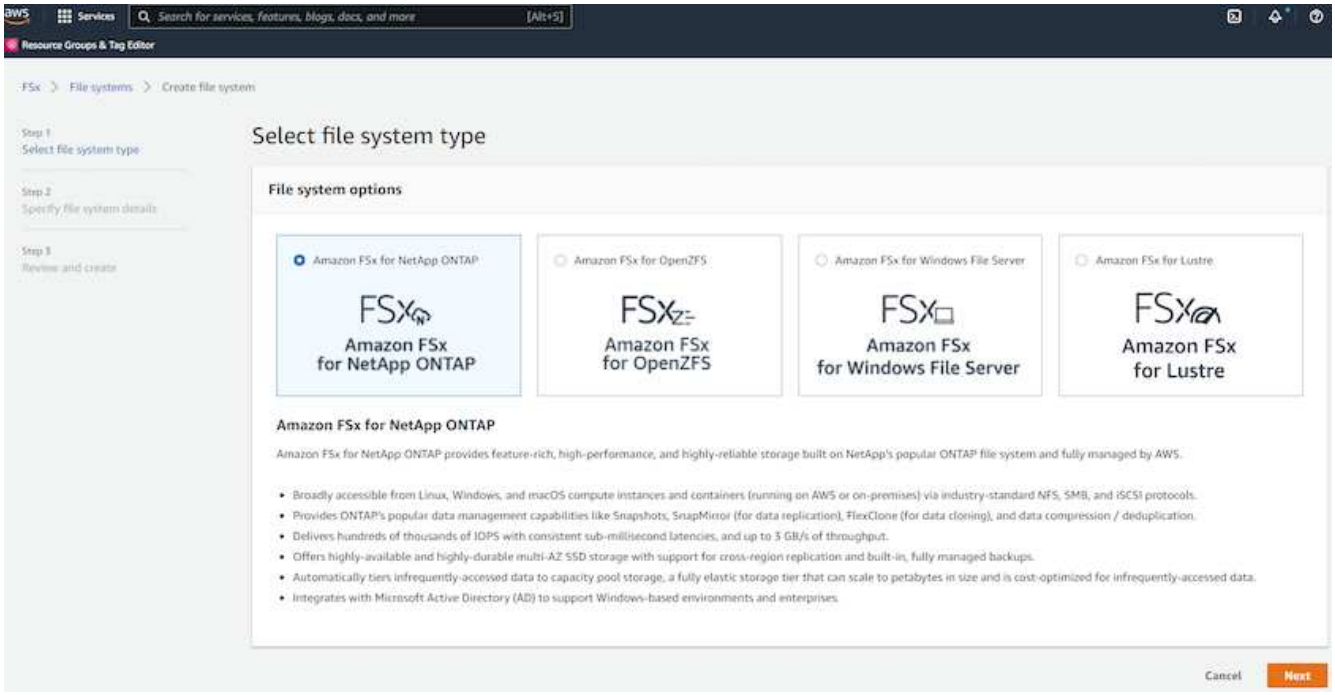


See the following step-by-step procedures for setting up either a primary or standby HA FSx cluster.

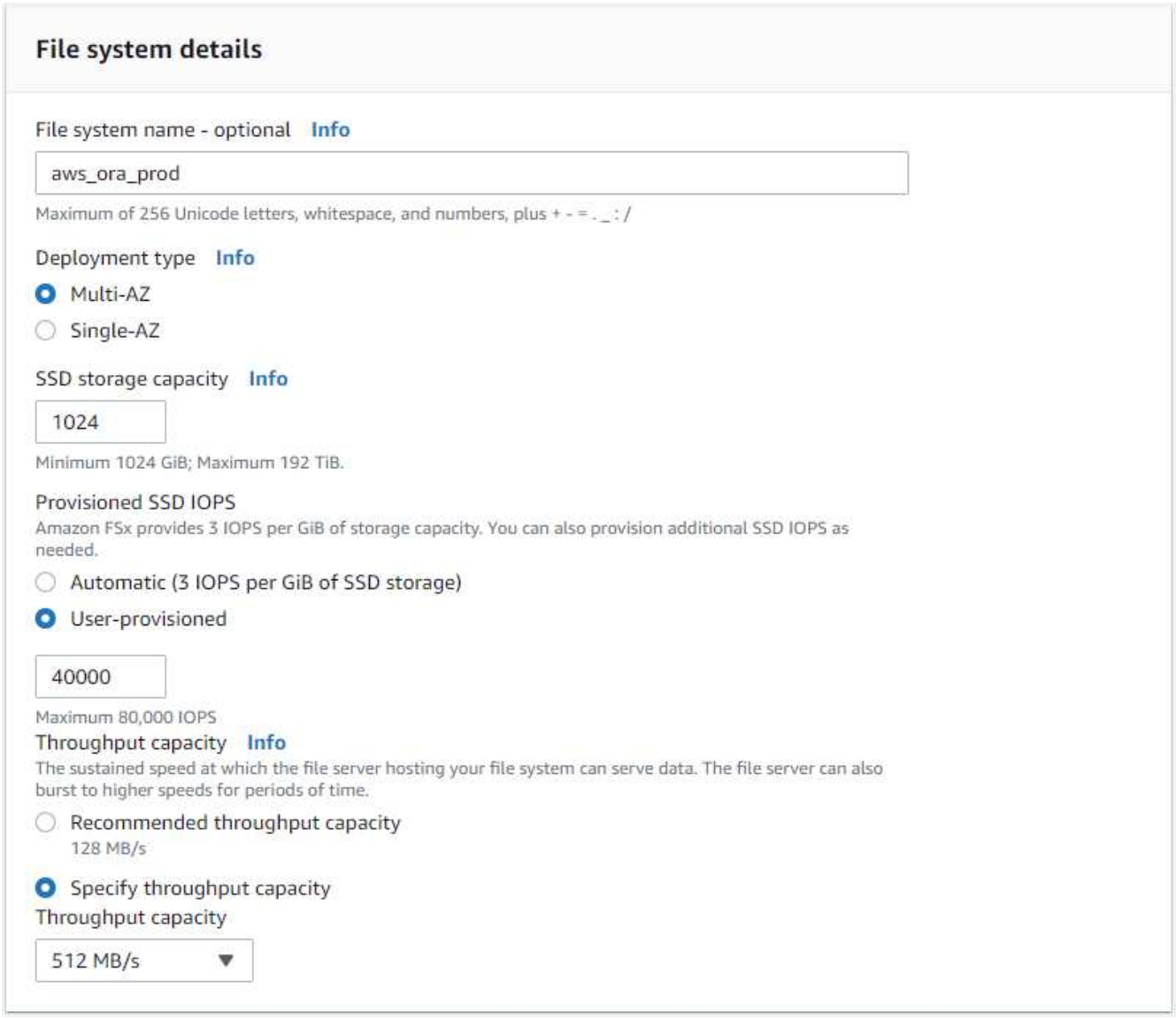
1. From the FSx console, click Create File System to start the FSx provision workflow.



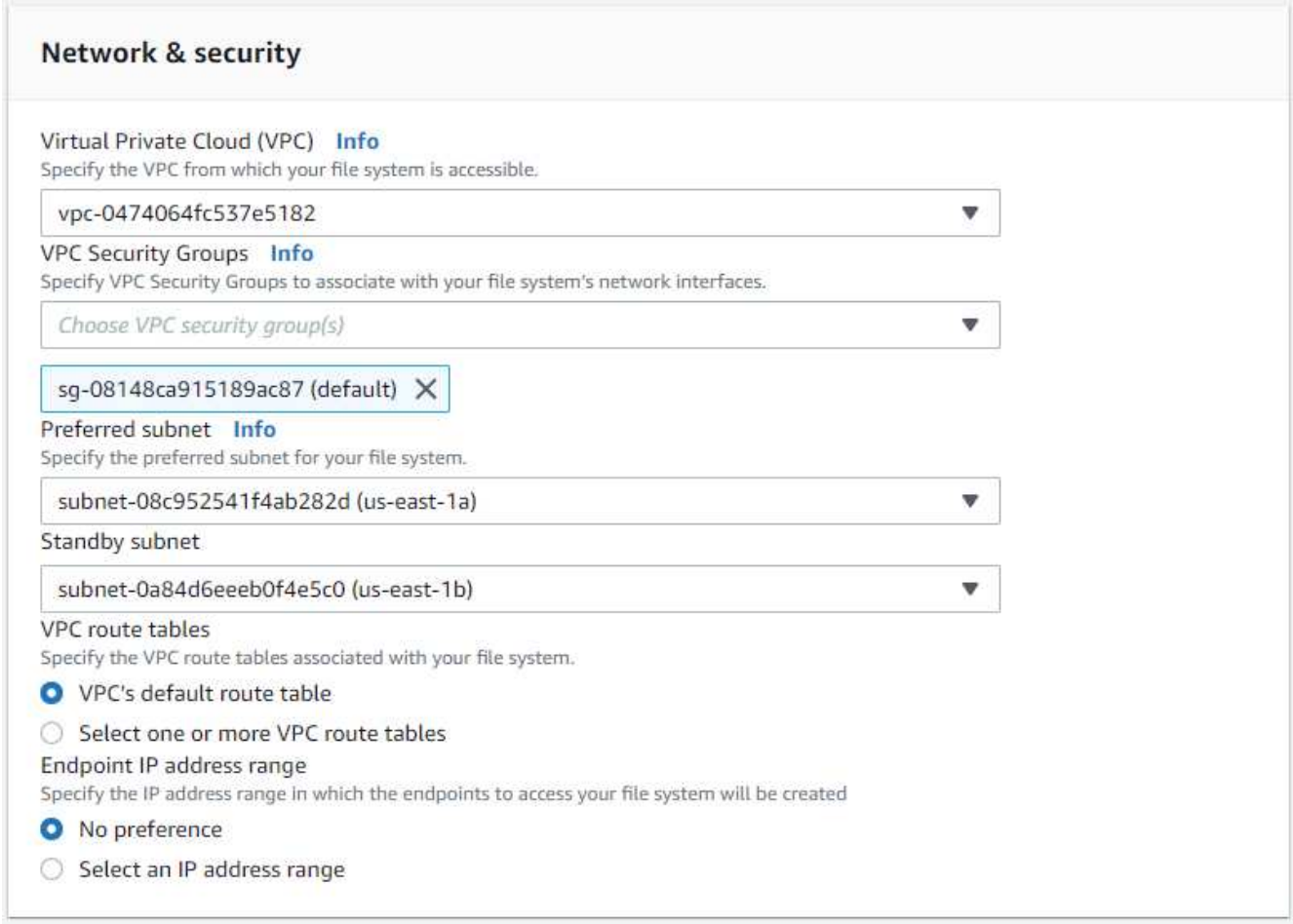
2. Select Amazon FSx for NetApp ONTAP. Then click Next.



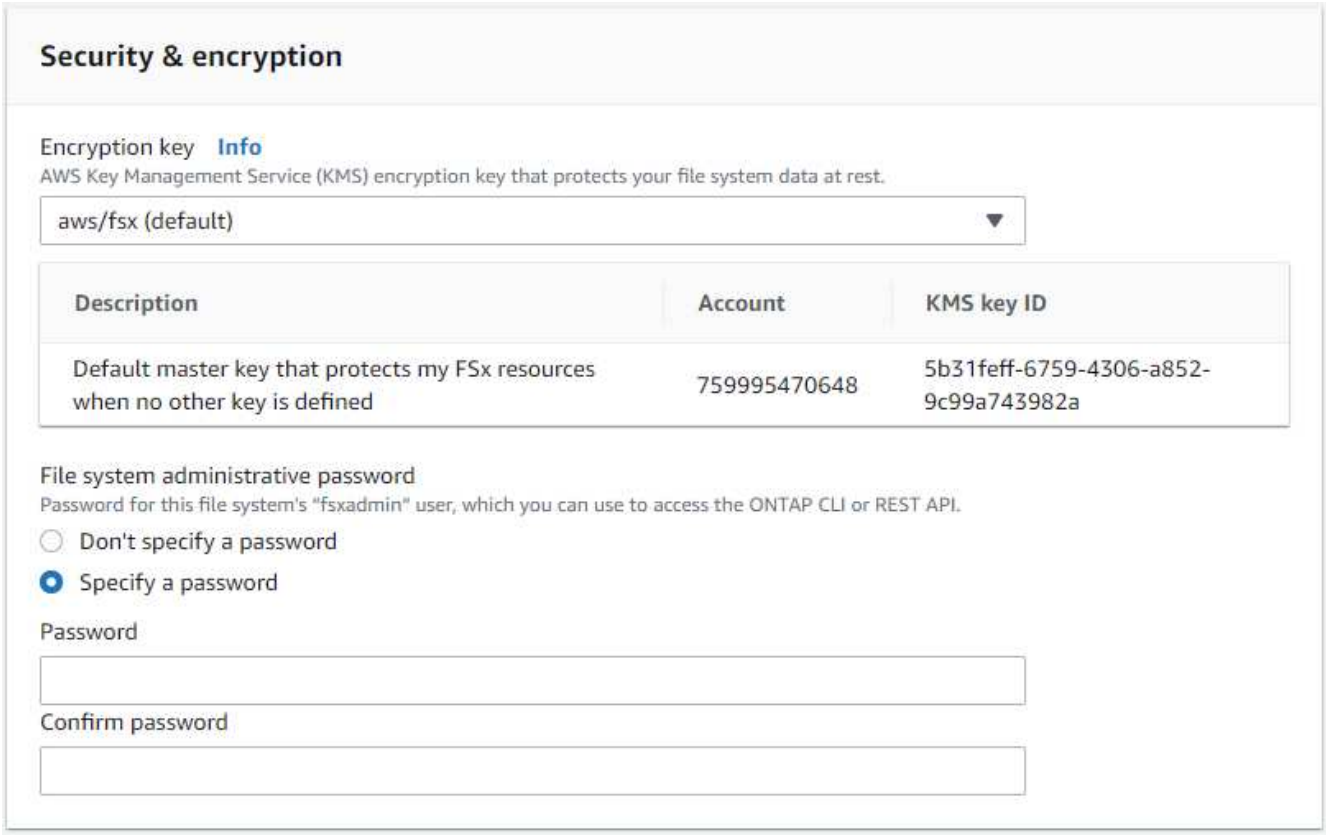
3. Select Standard Create and, in File System Details, name your file system, Multi-AZ HA. Based on your database workload, choose either Automatic or User-Provisioned IOPS up to 80,000 SSD IOPS. FSx storage comes with up to 2TiB NVMe caching at the backend that can deliver even higher measured IOPS.



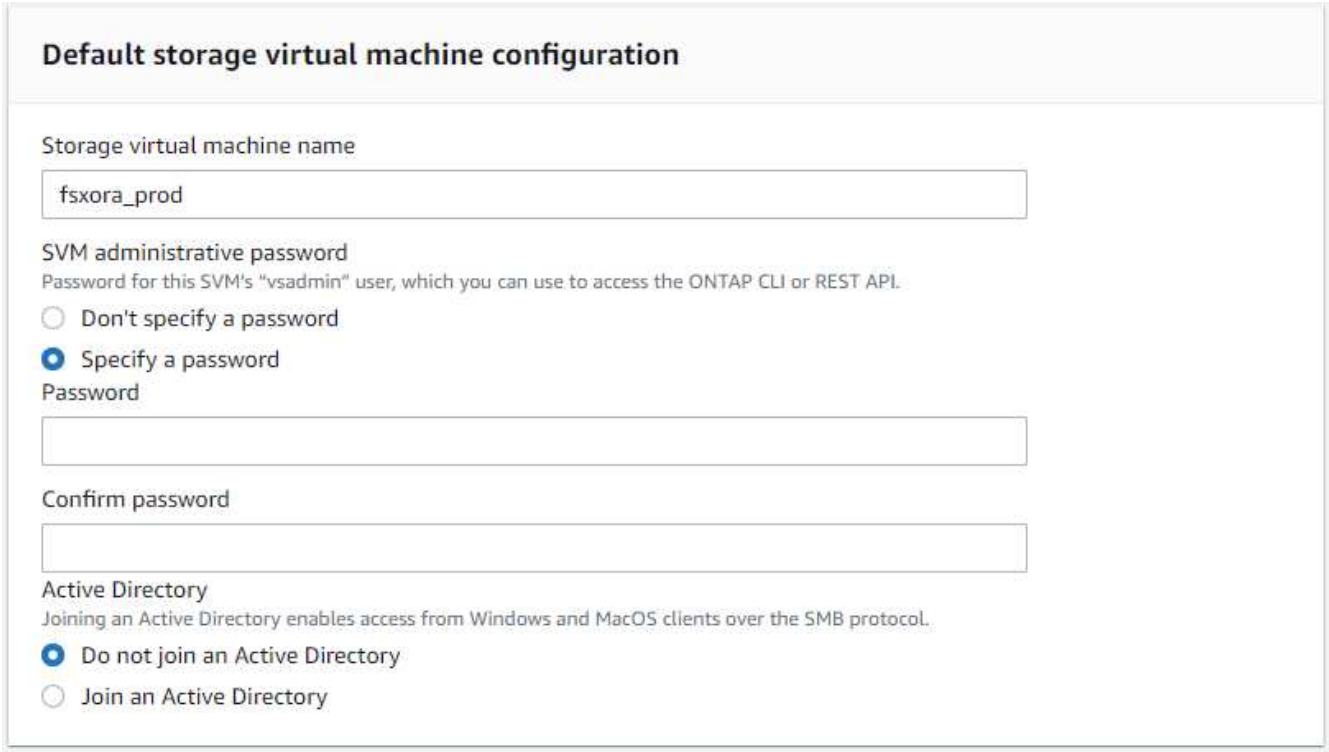
4. In the Network & Security section, select the VPC, security group, and subnets. These should be created before FSx deployment. Based on the role of the FSx cluster (primary or standby), place the FSx storage nodes in the appropriate zones.



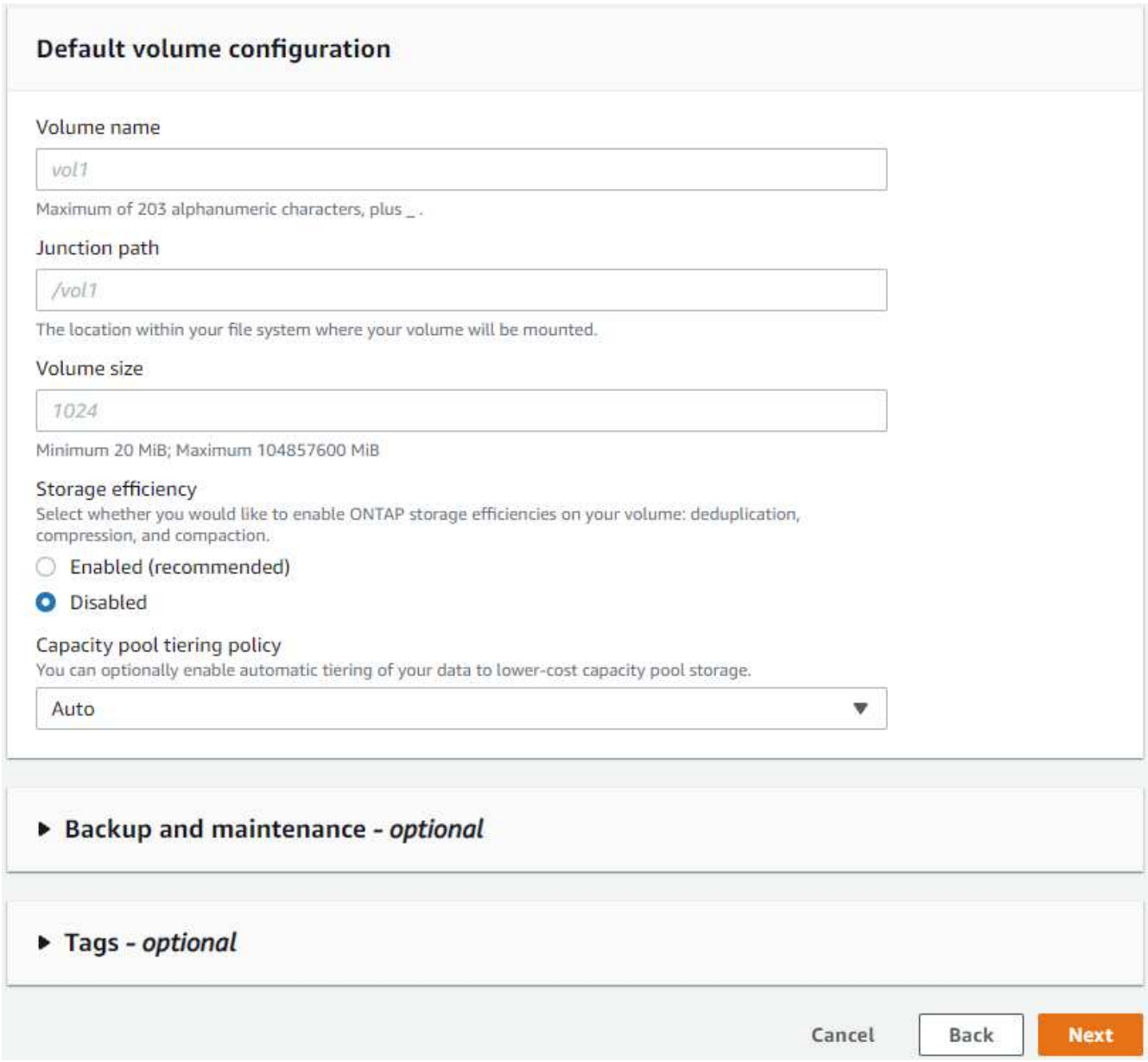
5. In the Security & Encryption section, accept the default, and enter the fsxadmin password.



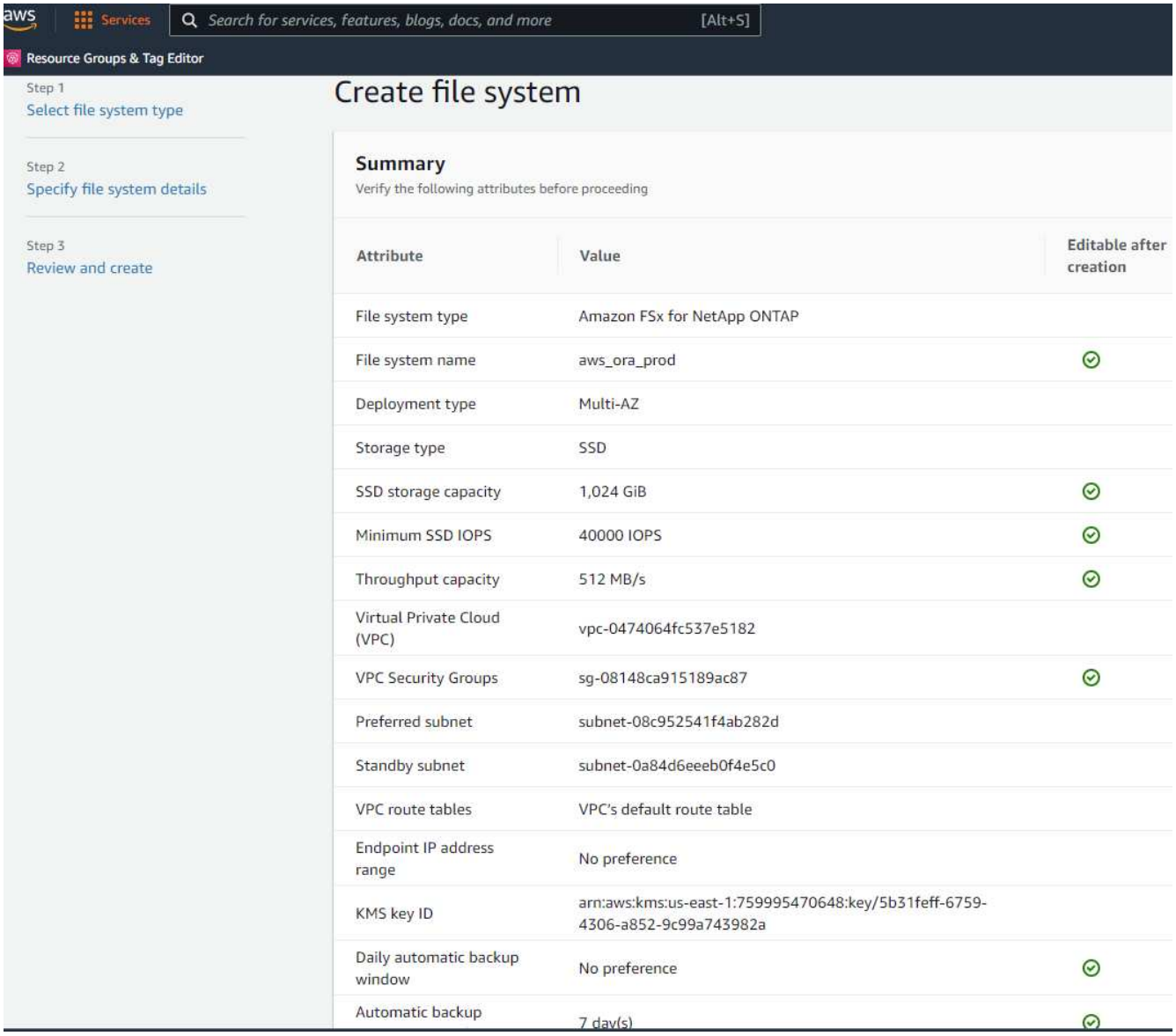
6. Enter the SVM name and the vsadmin password.



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



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



Provisioning of database volumes for Oracle database

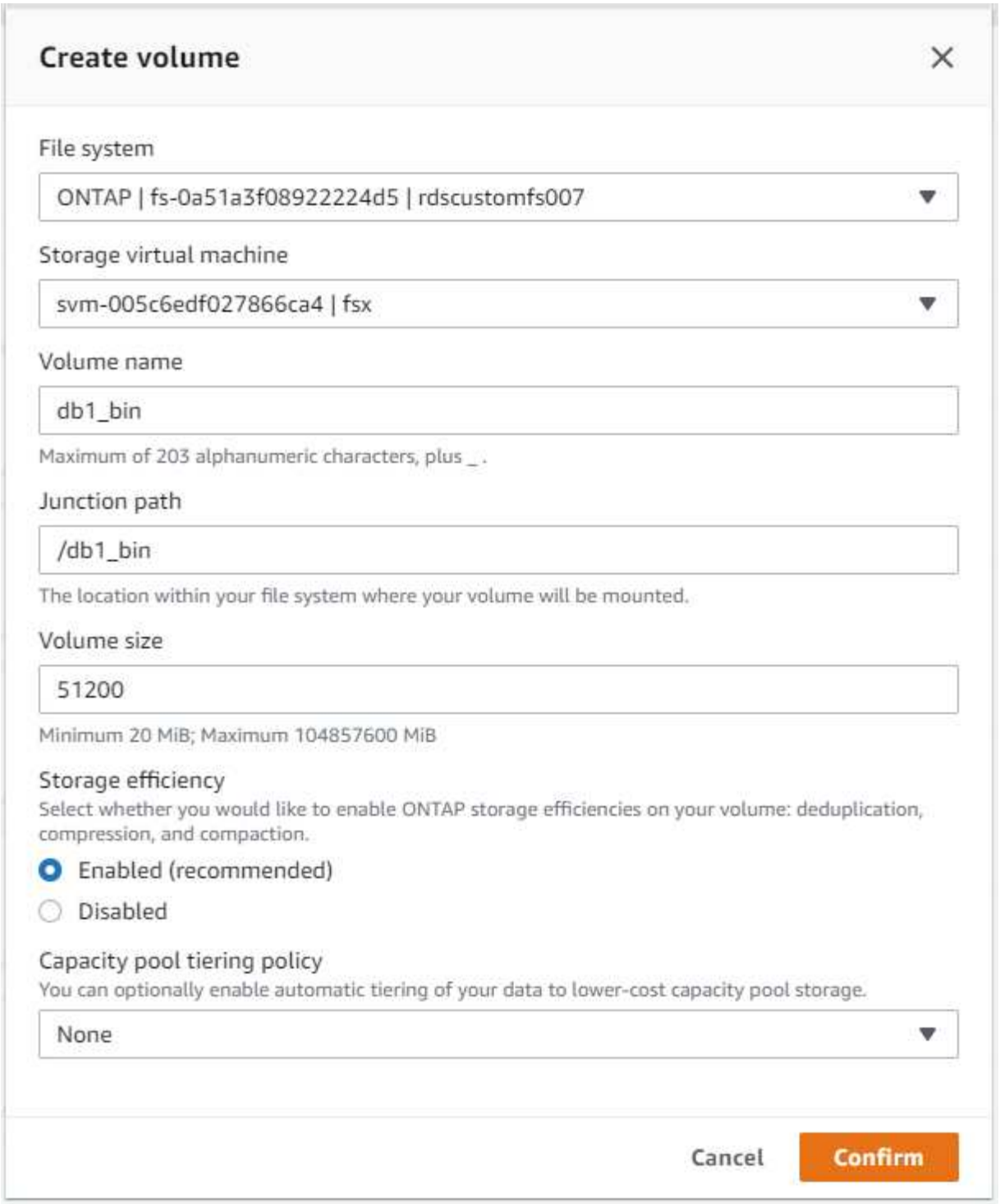
See Managing FSx for ONTAP volumes - creating a volume for details.

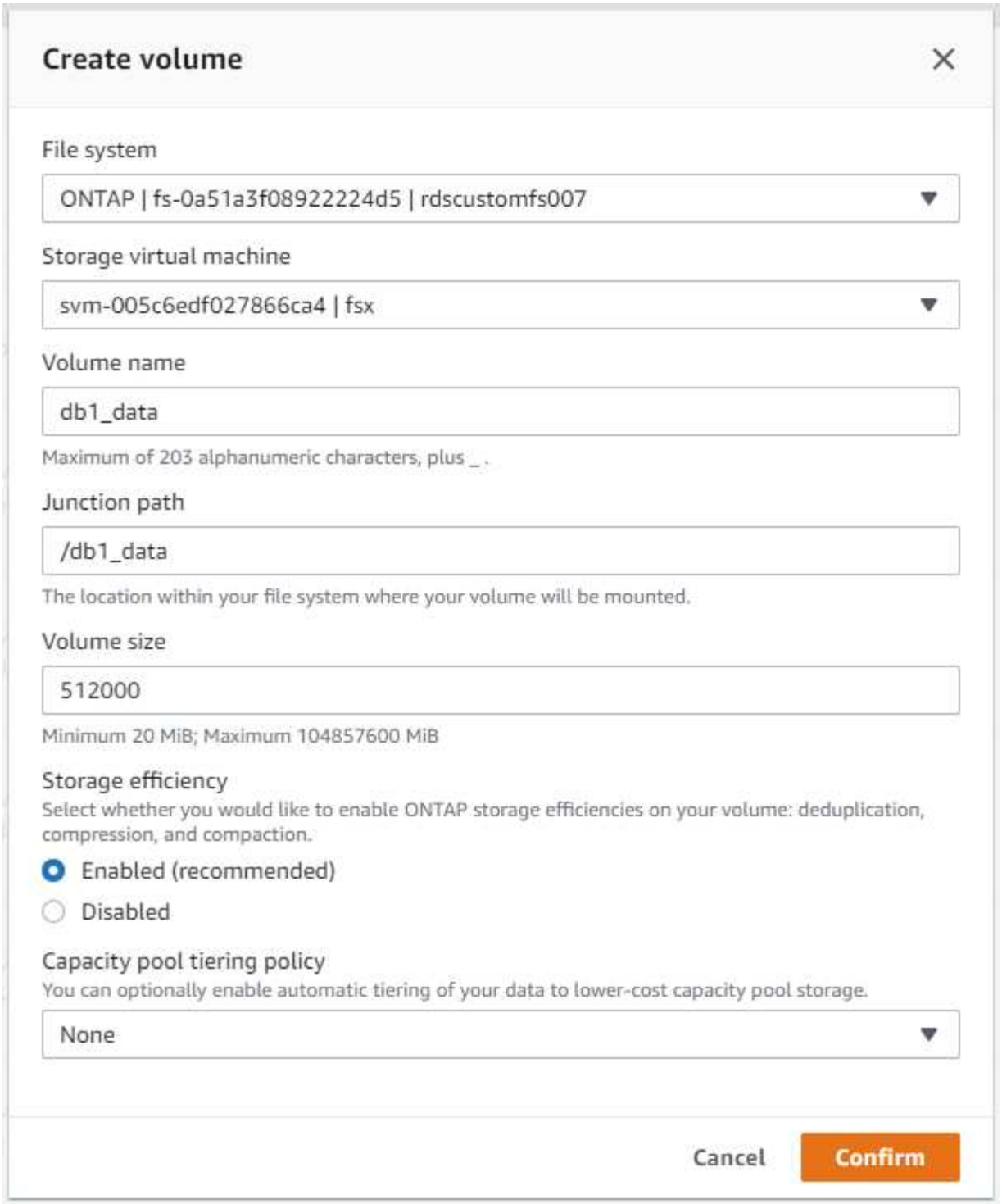
Key considerations:

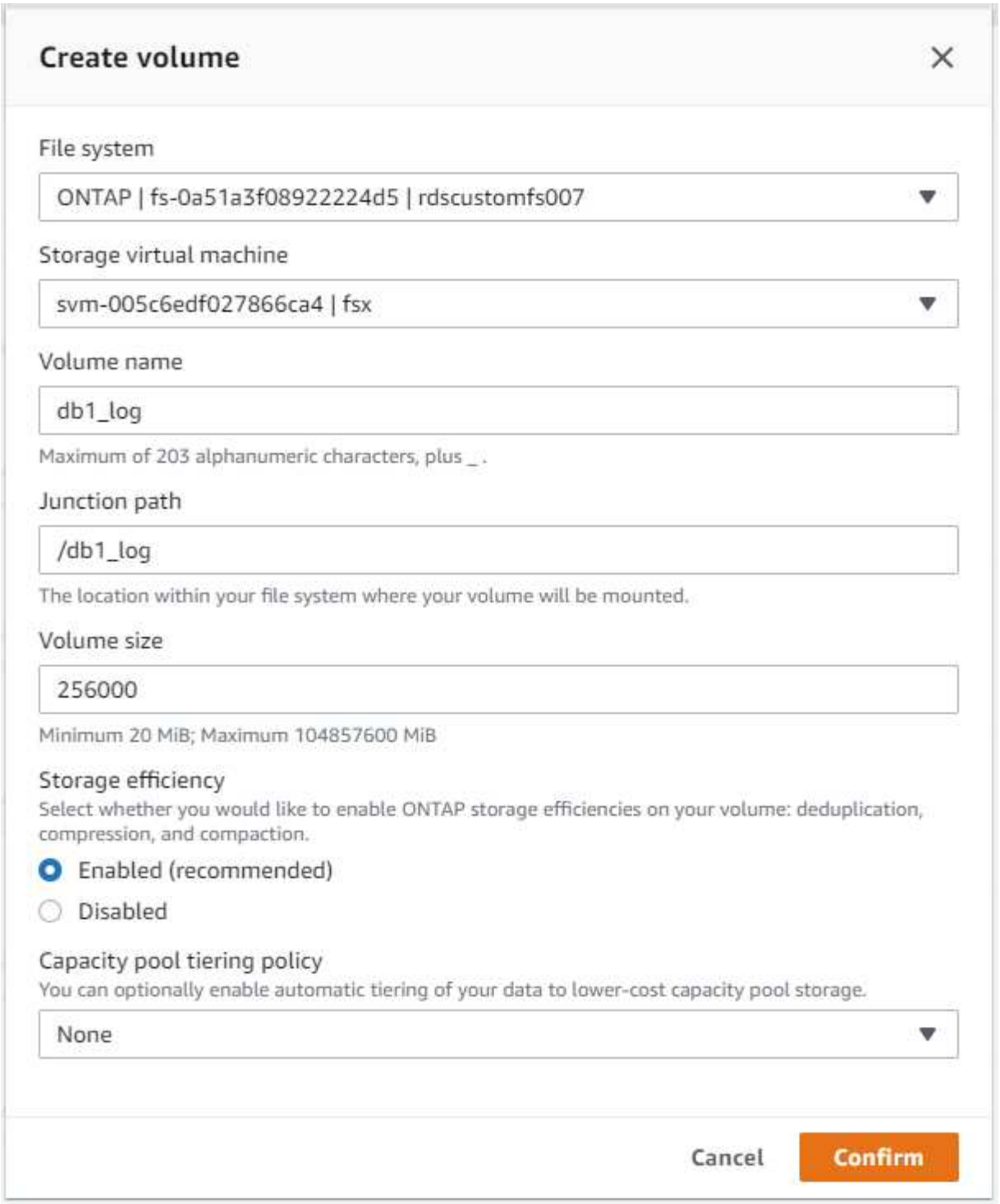
- Sizing the database volumes appropriately.
- Disabling capacity pool tiering policy for performance configuration.
- Enabling Oracle dNFS for NFS storage volumes.
- Setting up multipath for iSCSI storage volumes.

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



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 `accessstkey.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 accessstkey.pem -e @vars/fsx_vars.yml -t requirements_config
   ```

   ```
   ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private
   -key accessstkey.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.

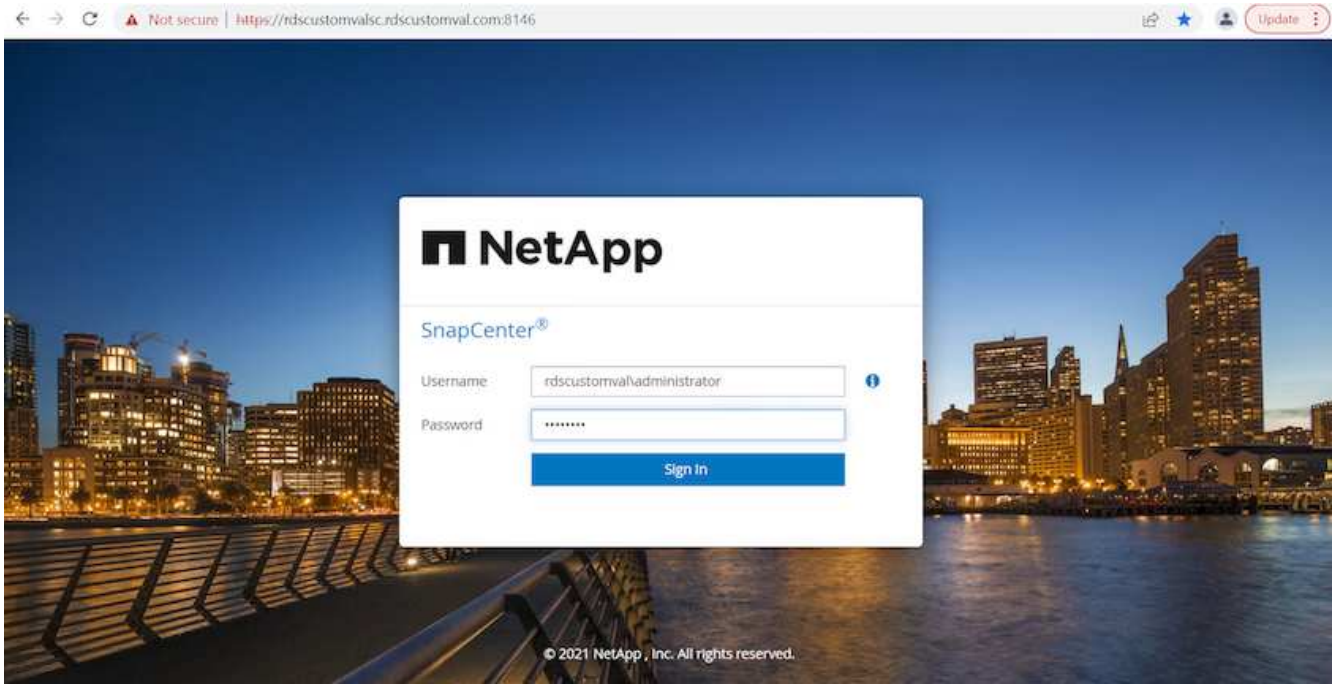
SnapCenter Deployment

SnapCenter installation

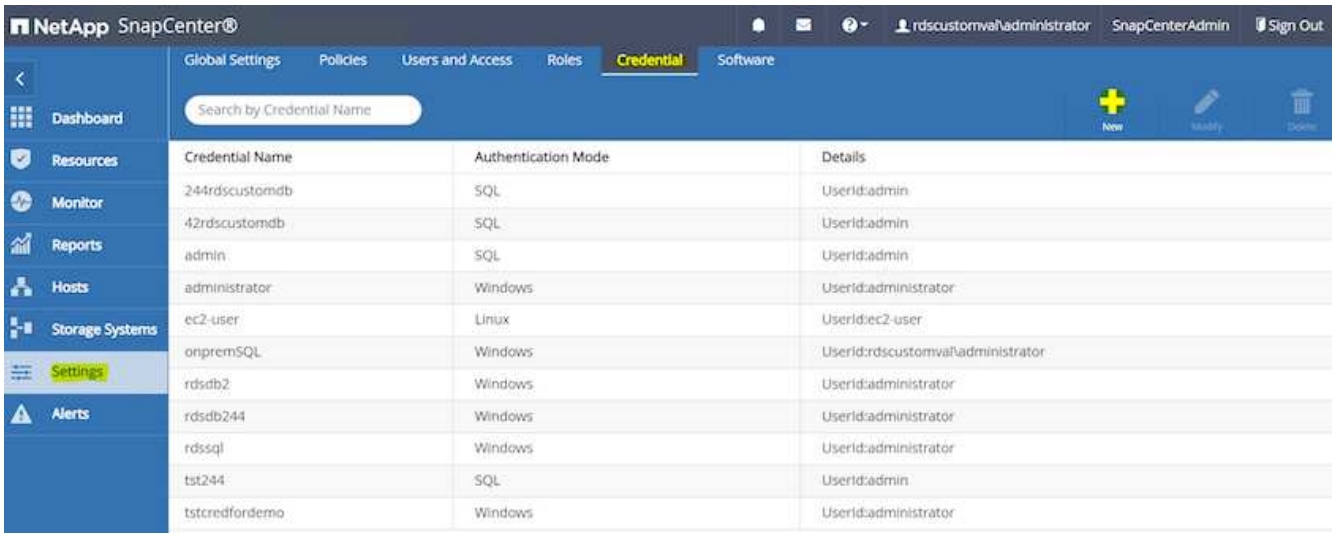
Follow Installing the SnapCenter Server to install SnapCenter server. This documentation covers how to install a standalone SnapCenter server. A SaaS version of SnapCenter is in beta review and could be available shortly. Check with your NetApp representative for availability if needed.

Configure SnapCenter plugin for EC2 Oracle host

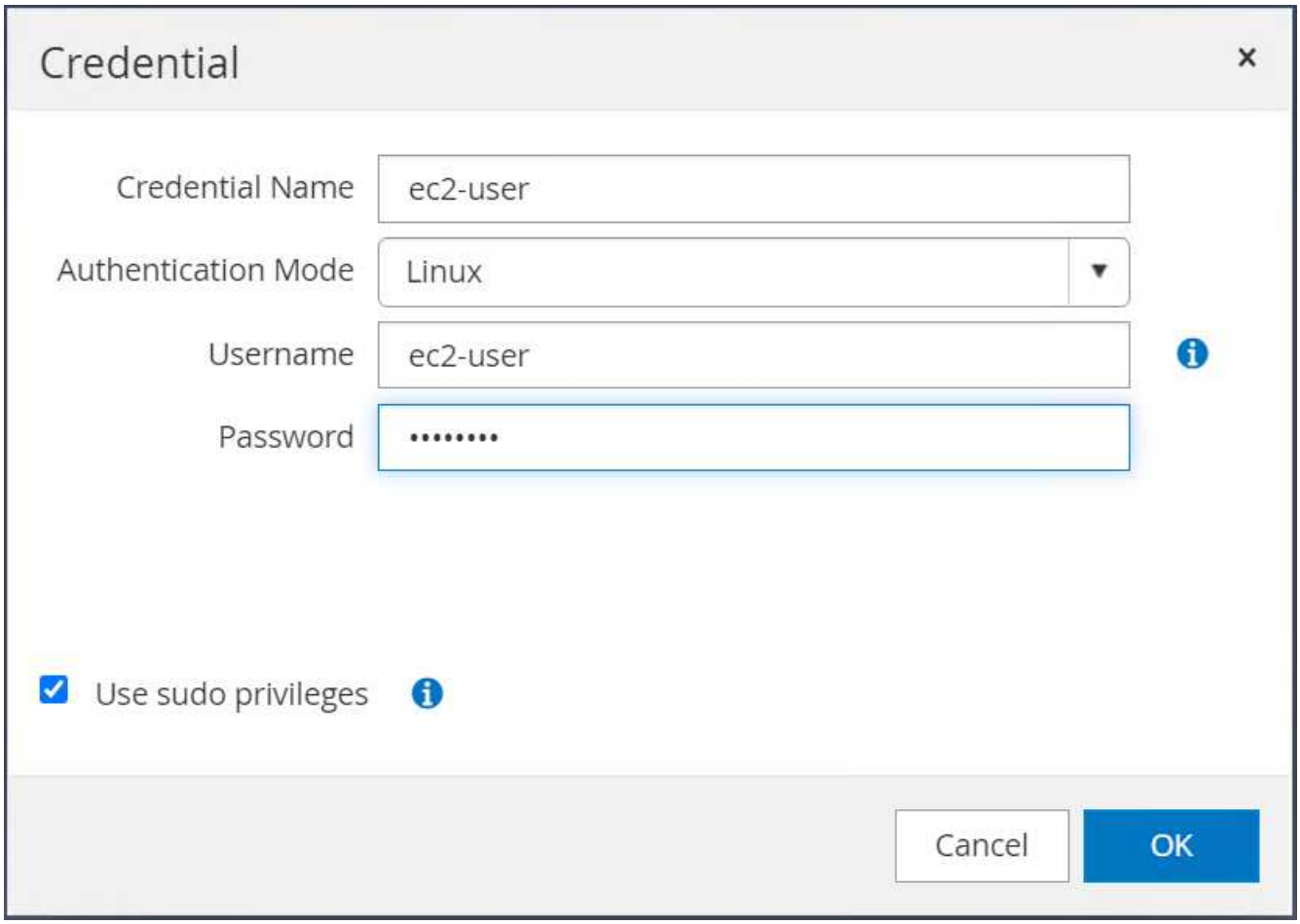
1. After automated SnapCenter installation, log into SnapCenter as an administrative user for the Window host on which the SnapCenter server is installed.



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.



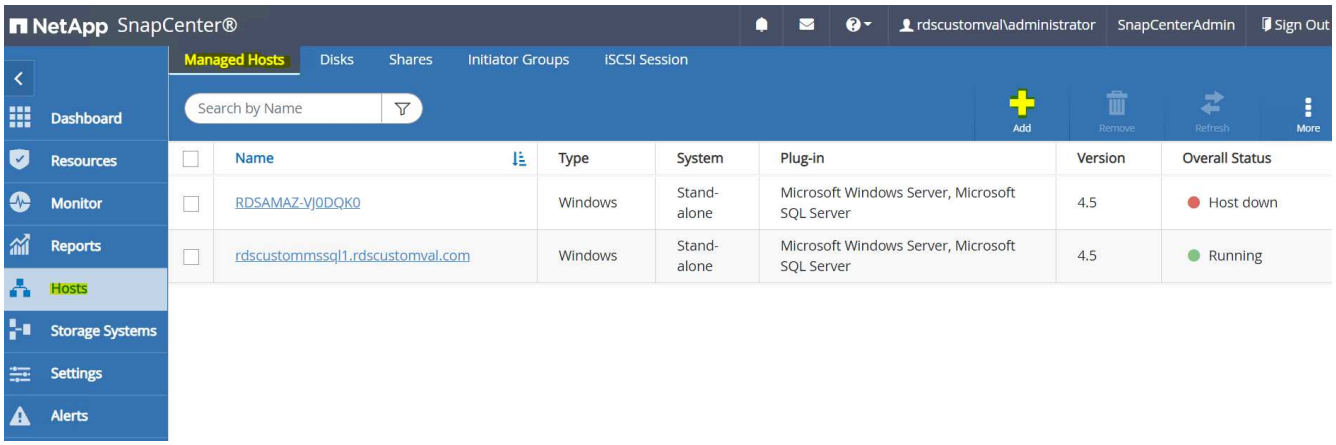
5. Add the SnapCenter server name and the IP address to the EC2 instance host file for name resolution.

```
[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
```

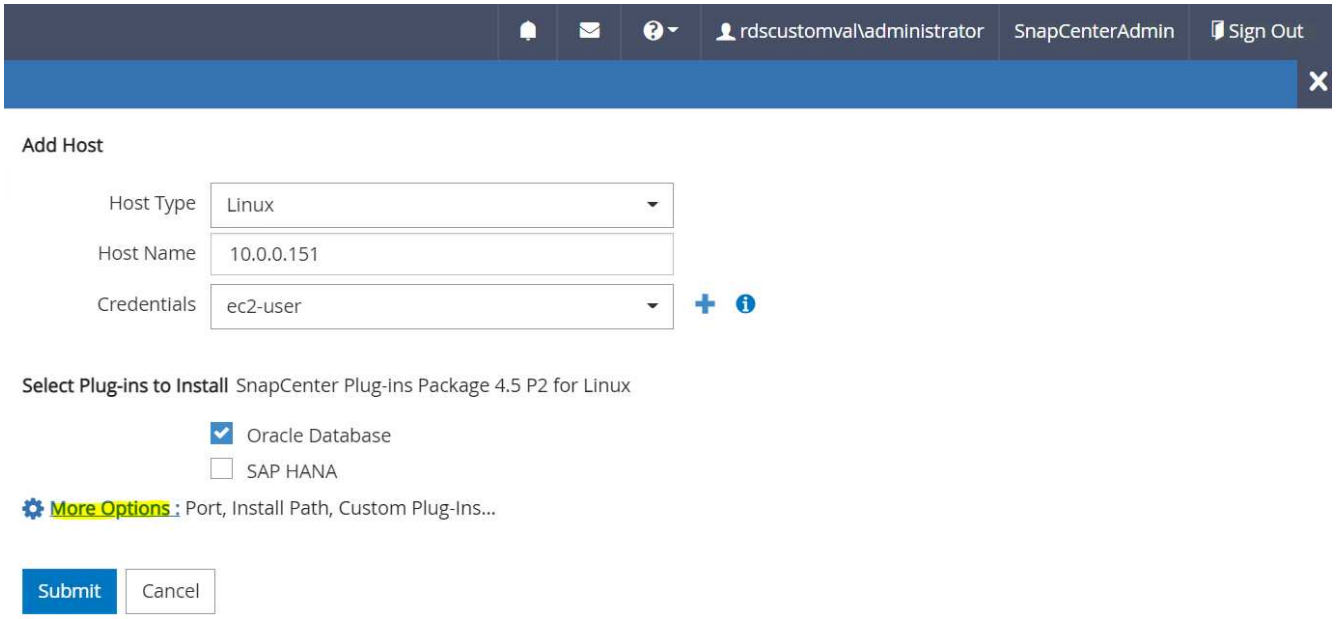
6. On the SnapCenter server Windows host, add the EC2 instance host IP address to the Windows host file `C:\Windows\System32\drivers\etc\hosts`.

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

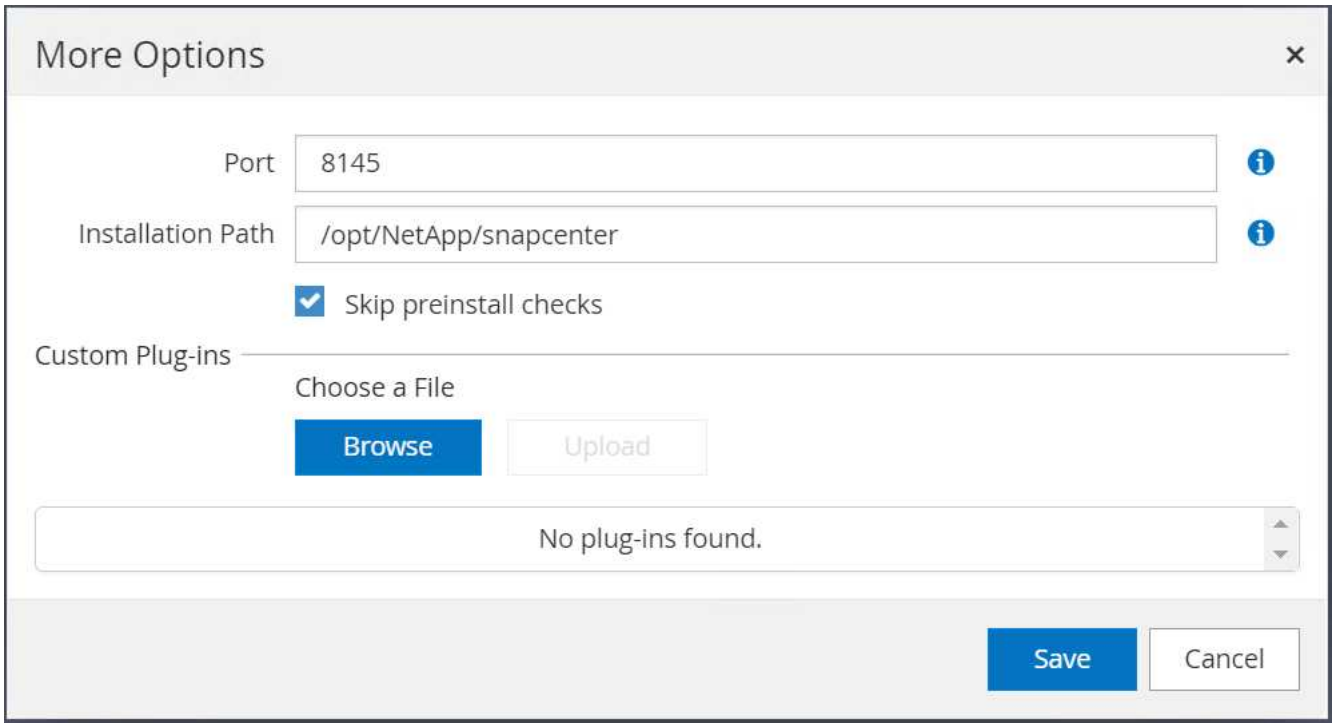
7. In the left-side menu, select Hosts > Managed Hosts, and then click Add to add the EC2 instance host to SnapCenter.



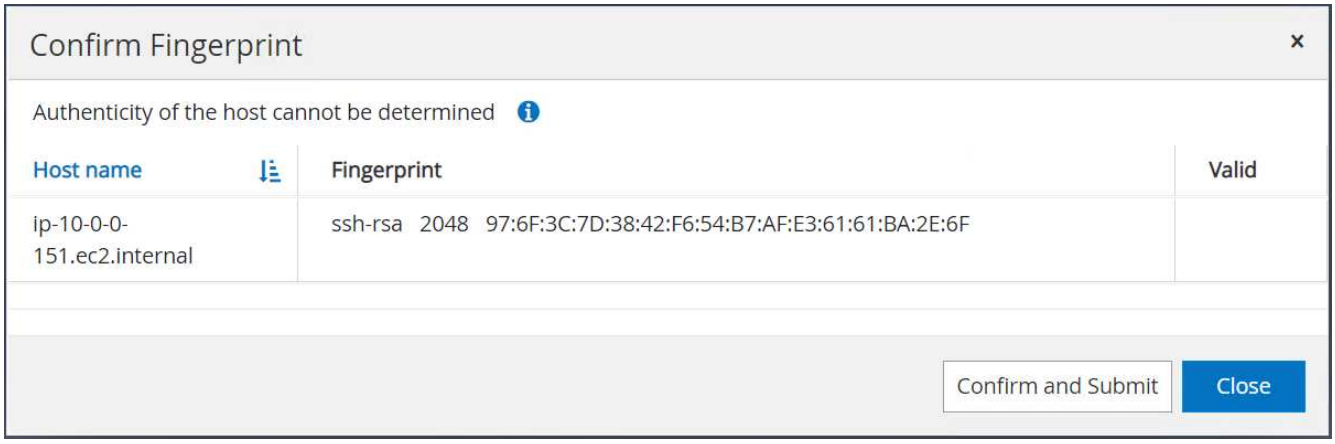
Check Oracle Database, and, before you submit, click More Options.



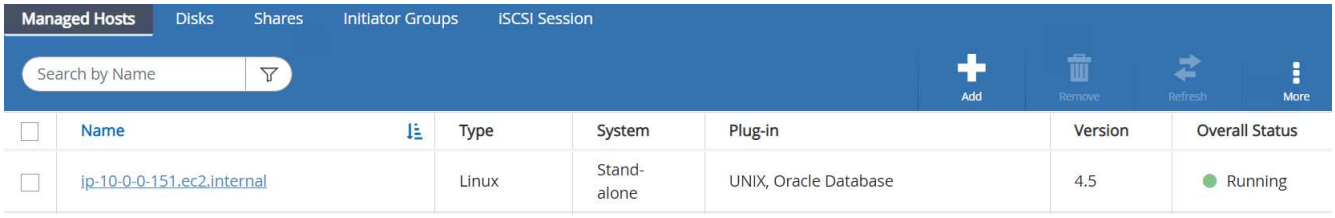
Check Skip Preinstall Checks. Confirm Skipping Preinstall Checks, and then click Submit After Save.



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.

EC2 and FSx Oracle database management

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.

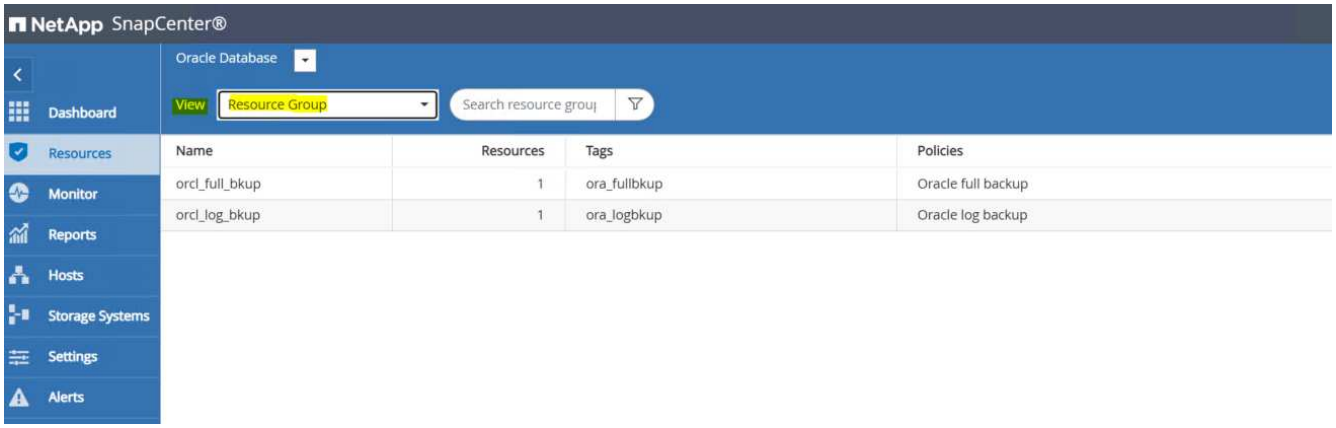
Taking a snapshot

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.

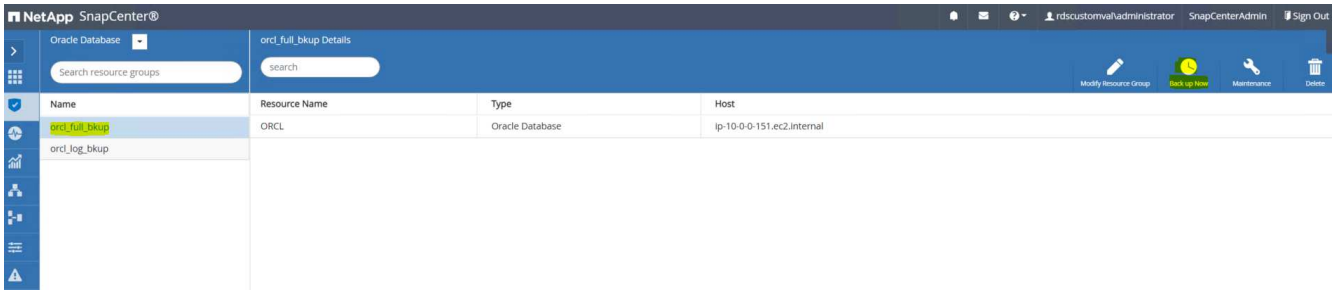
Taking a full database snapshot

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

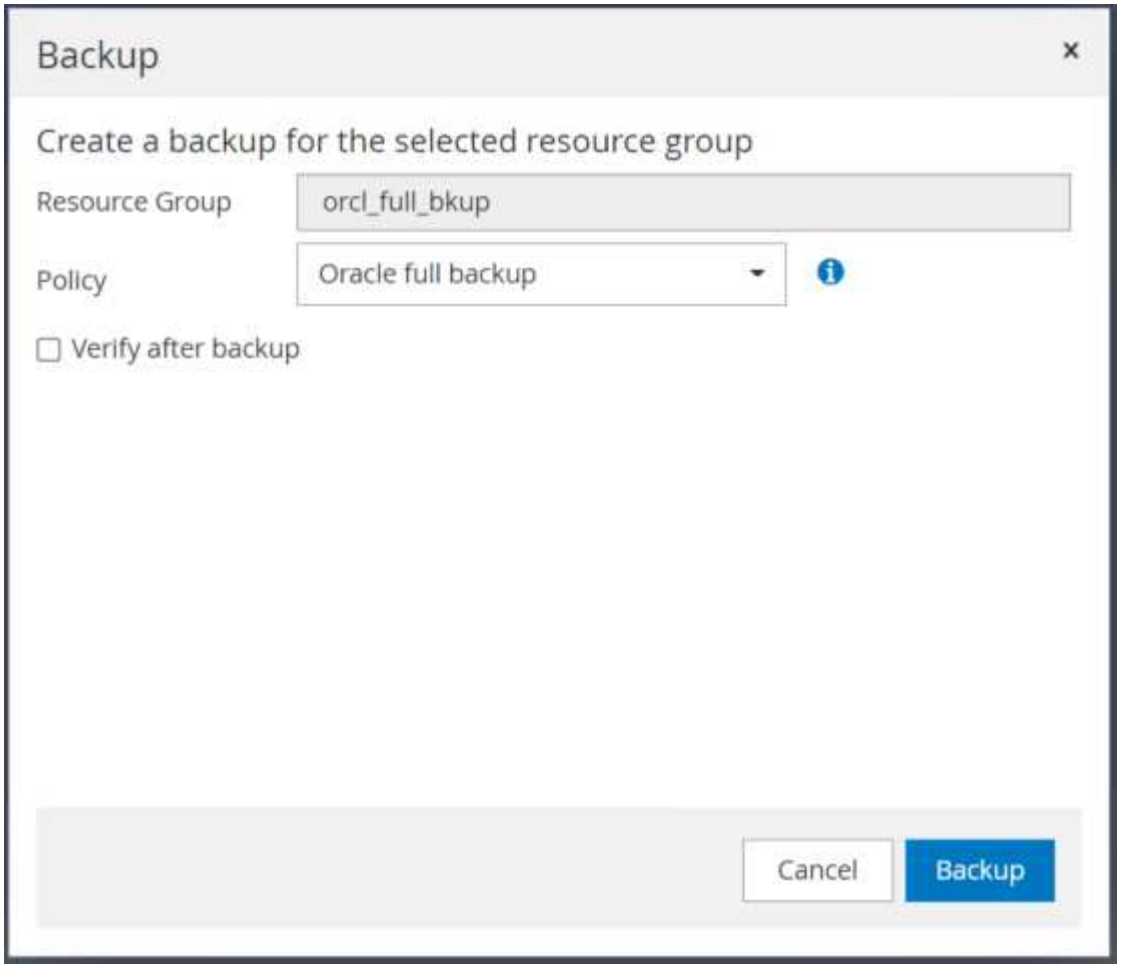
1. Log into the SnapCenter UI and click Resources in the left-side menu. From the View dropdown, change to the Resource Group view.



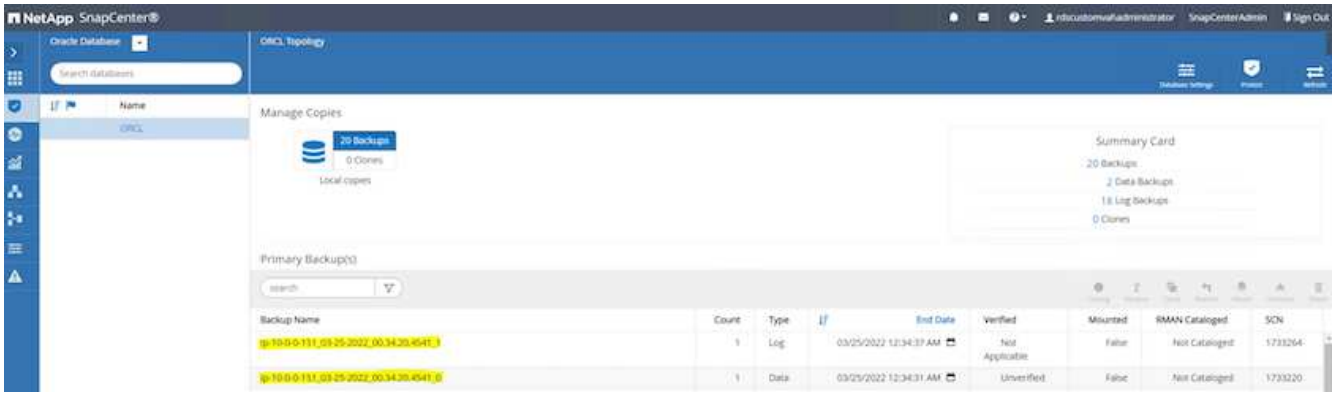
2. Click the full backup resource name, and then click the Backup Now icon to initiate an add-hoc backup.



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



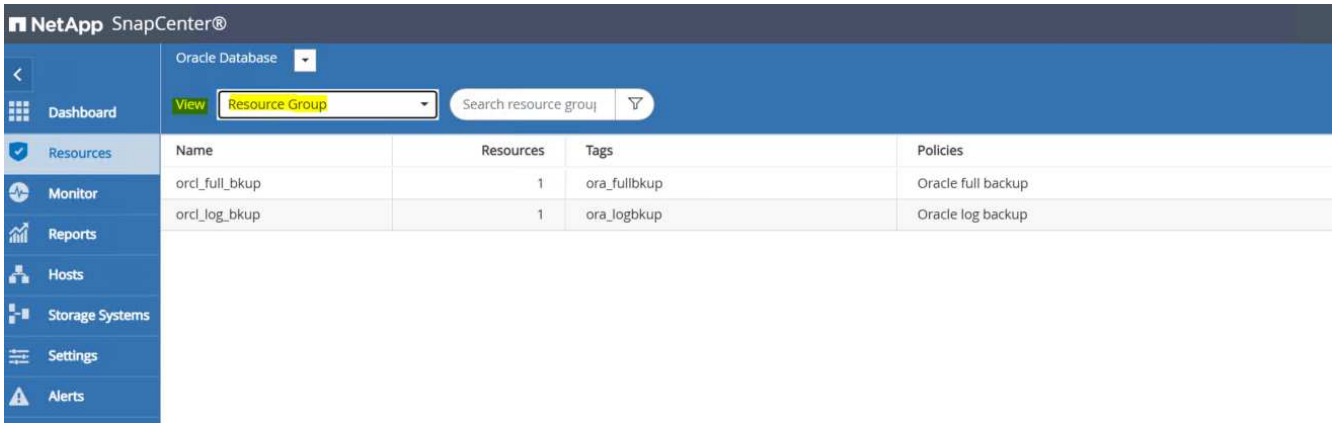
From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off backup completed successfully. A full database backup creates two snapshots: one for the data volume and one for the log volume.



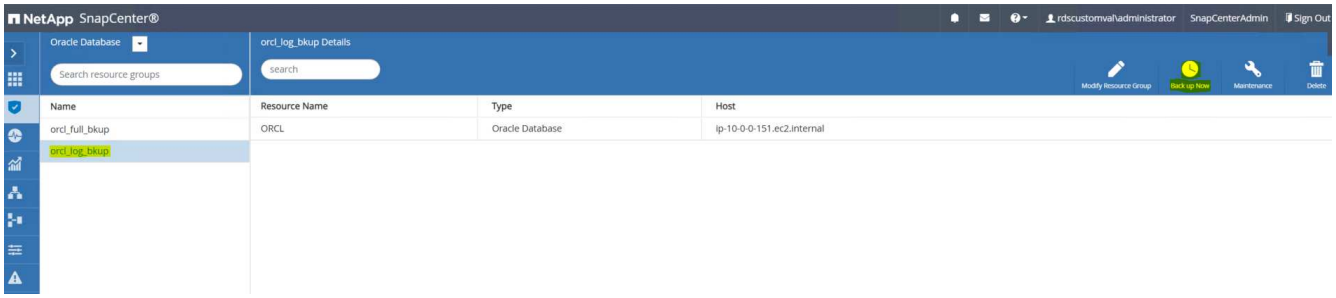
Taking an archive log snapshot

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

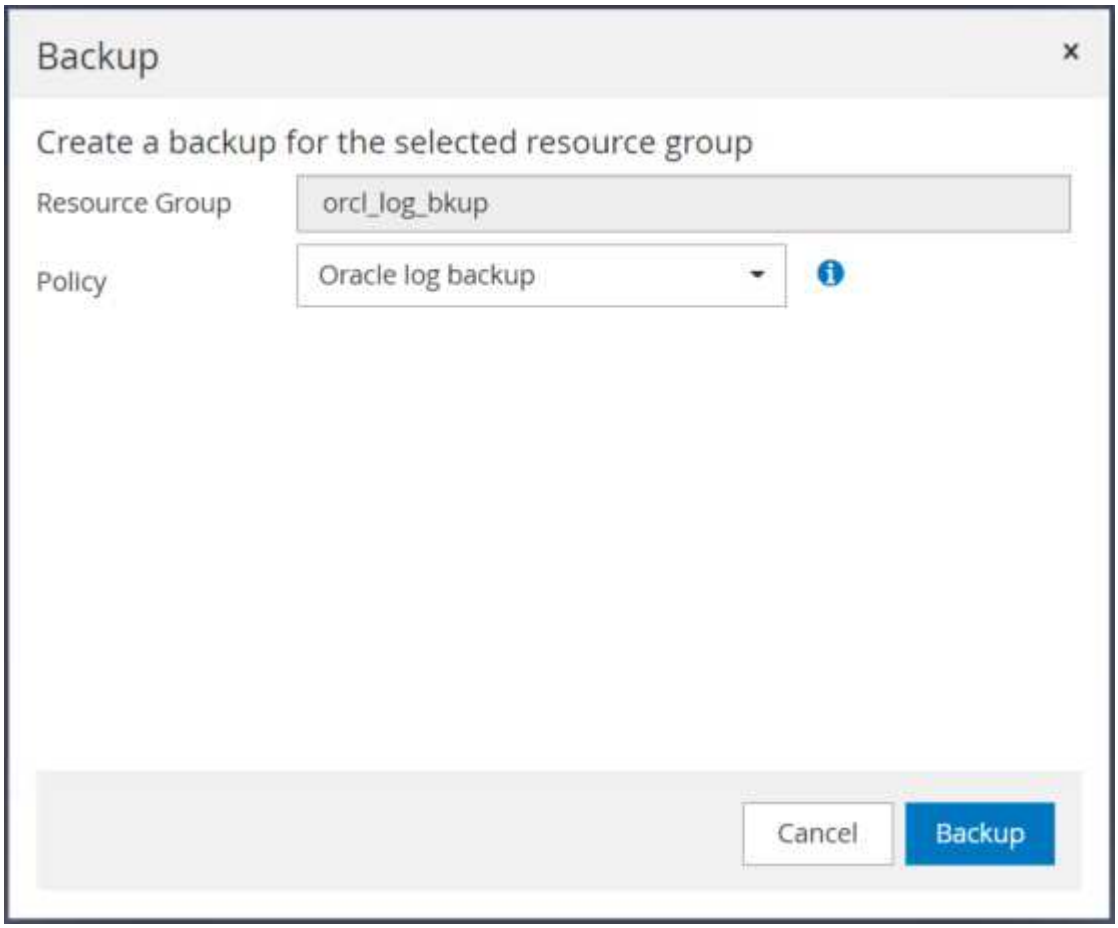
1. Log into the SnapCenter UI and click the Resources tab in the left-side menu bar. From the View dropdown, change to the Resource Group view.



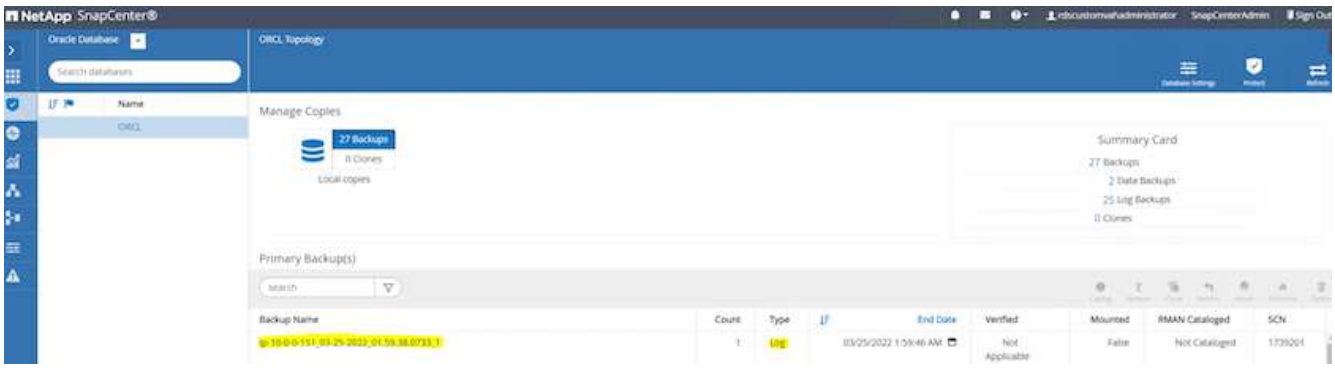
2. Click the log backup resource name, and then click the Backup Now icon to initiate an add-hoc backup for archive logs.



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.

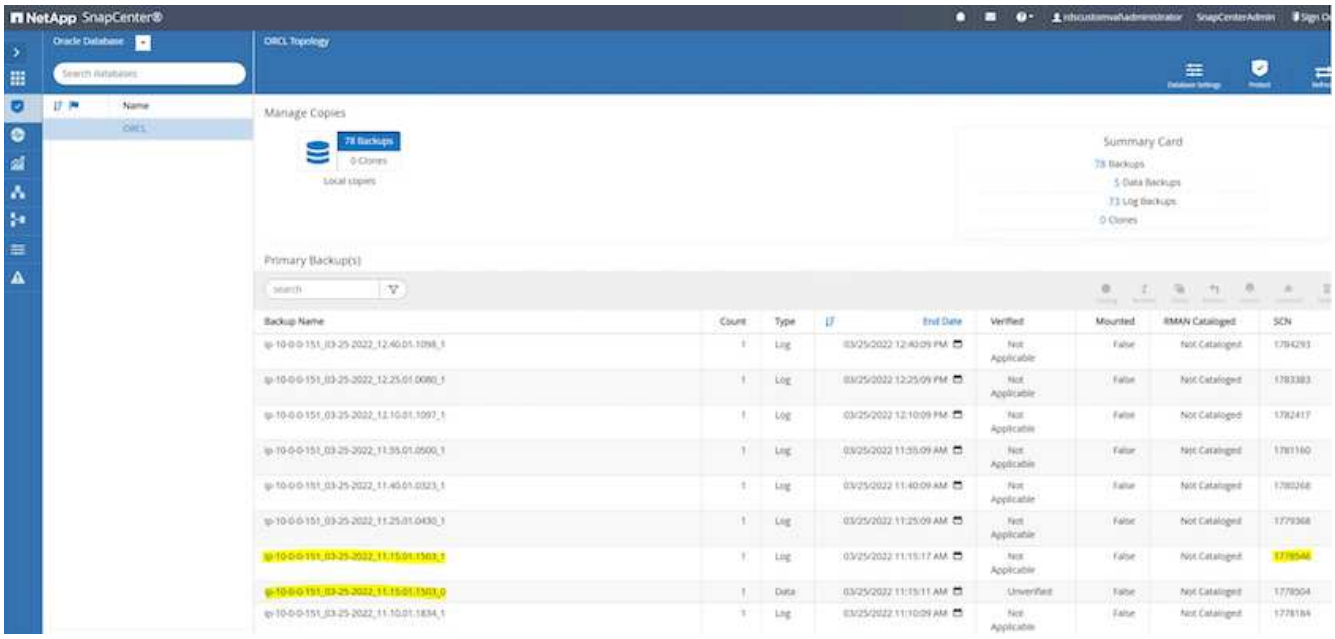


Restoring to a point in time

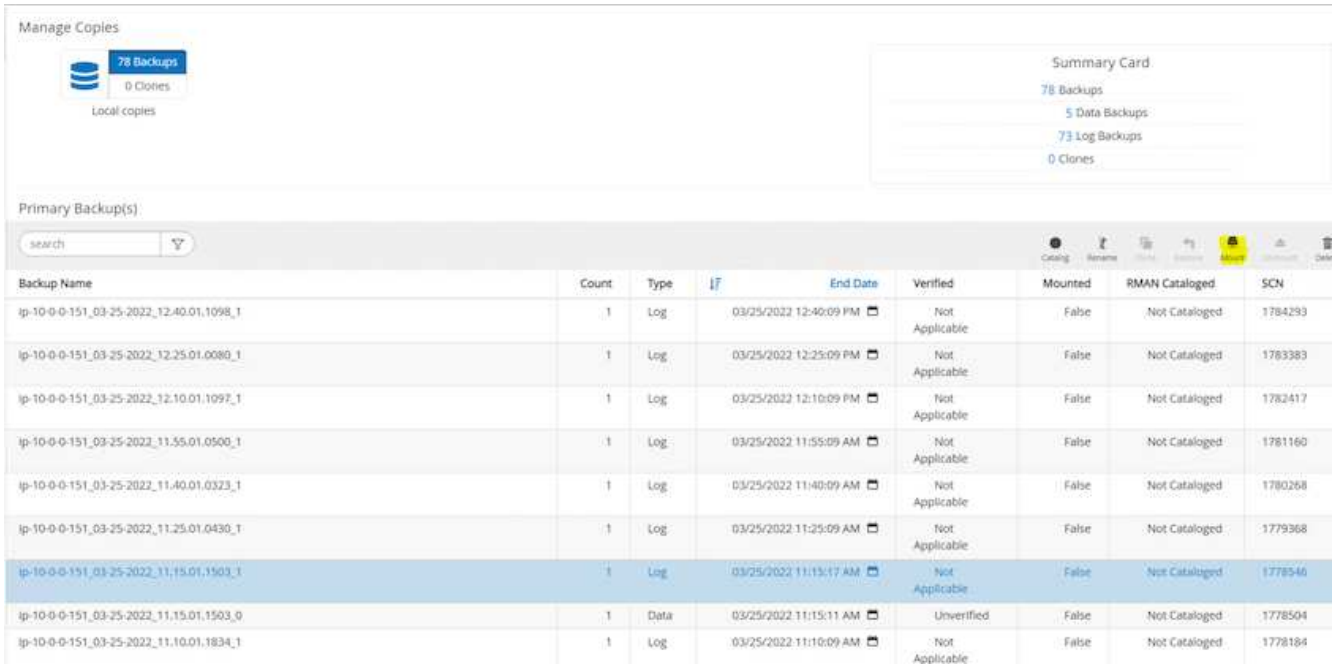
SnapCenter-based restore to a point in time is executed on the same EC2 instance host. Complete the following steps to perform the restore:

1. From the SnapCenter Resources tab > Database view, click the database name to open the database backup.

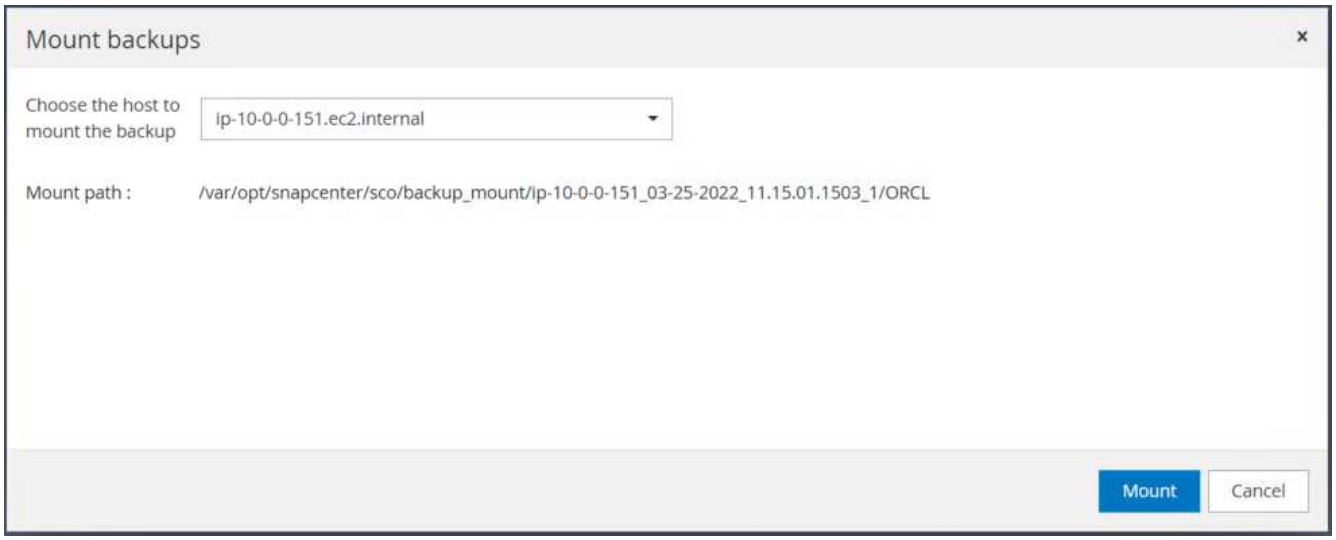
2. Select the database backup copy and the desired point in time to be restored. Also mark down the corresponding SCN number for the point in time. The point-in-time restore can be performed using either the time or the SCN.



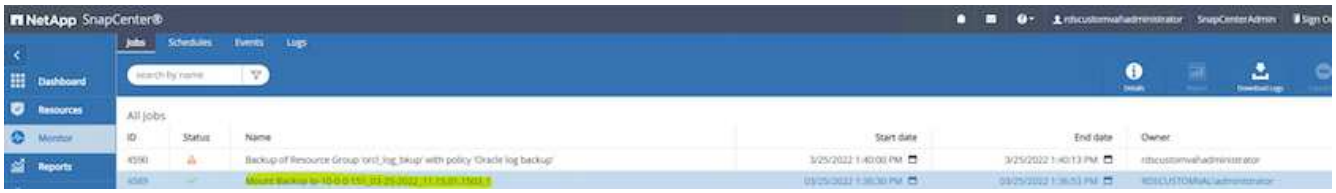
3. Highlight the log volume snapshot and click the Mount button to mount the volume.



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.

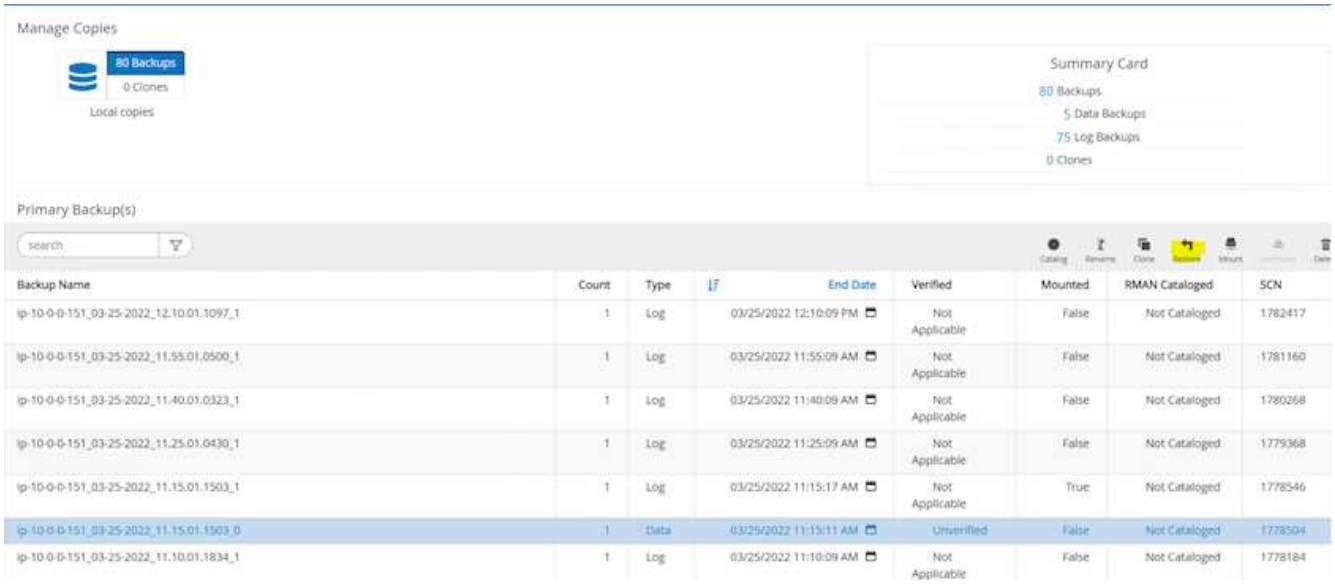



```
[root@ip-10-0-0-151 ec2-user]# df -h
Filesystem                                      Size  Used Avail Use% Mounted on
devtmpfs                                        7.6G     0  7.6G   0% /dev
tmpfs                                            16G  7.0G  8.3G  46% /dev/shm
tmpfs                                           7.7G  604K  7.6G   1% /run
tmpfs                                           7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/nvme0n1p1                                  9.8G  5.4G  4.3G  56% /
198.19.255.68:/ora_nfs_log                       48G   95M   48G   1% /ora_nfs_log
198.19.255.68:/ora_nfs_data                      48G  3.4G   45G   8% /ora_nfs_data
/dev/mapper/dbdata01-lvdbdata01                  40G  471M   39G   2% /rdsdbdata
/dev/nvme5n1                                     25G   12G   13G  49% /rdsdbbin
tmpfs                                           1.6G     0  1.6G   0% /run/user/61001
tmpfs                                           1.6G     0  1.6G   0% /run/user/61005
198.19.255.68:/5cef91c793-5583-480d-9a34-6275dab17f5b   48G   91M   48G   1% /var/opt/snapcenter/sco/backup_mount/ip-10-0-0-151_03-25-2022_11.15.01.1503_1/ORCL/1
[root@ip-10-0-0-151 ec2-user]#
```

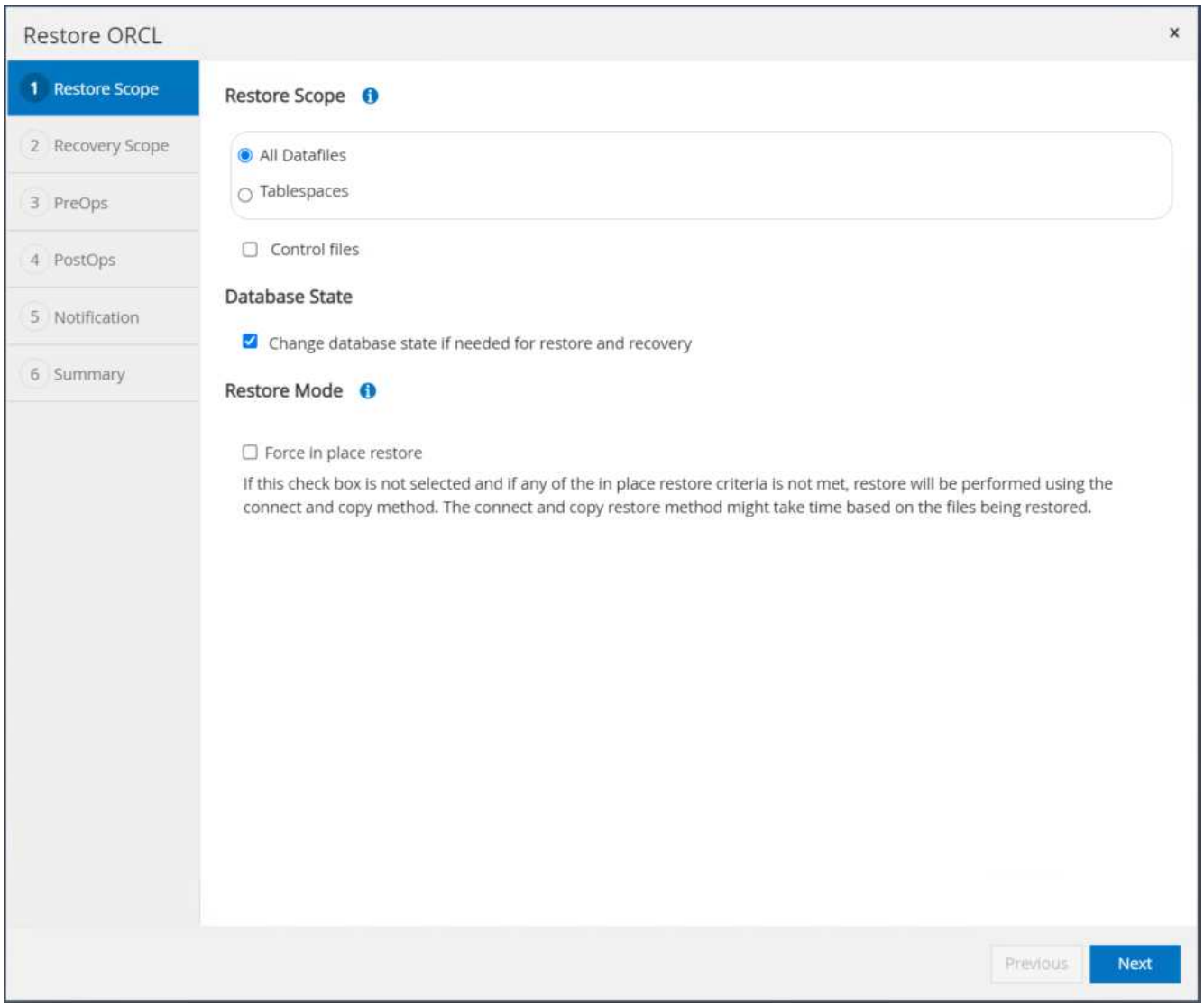
6. Copy the archive logs from the mounted log volume to the current archive log directory.

```
[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.

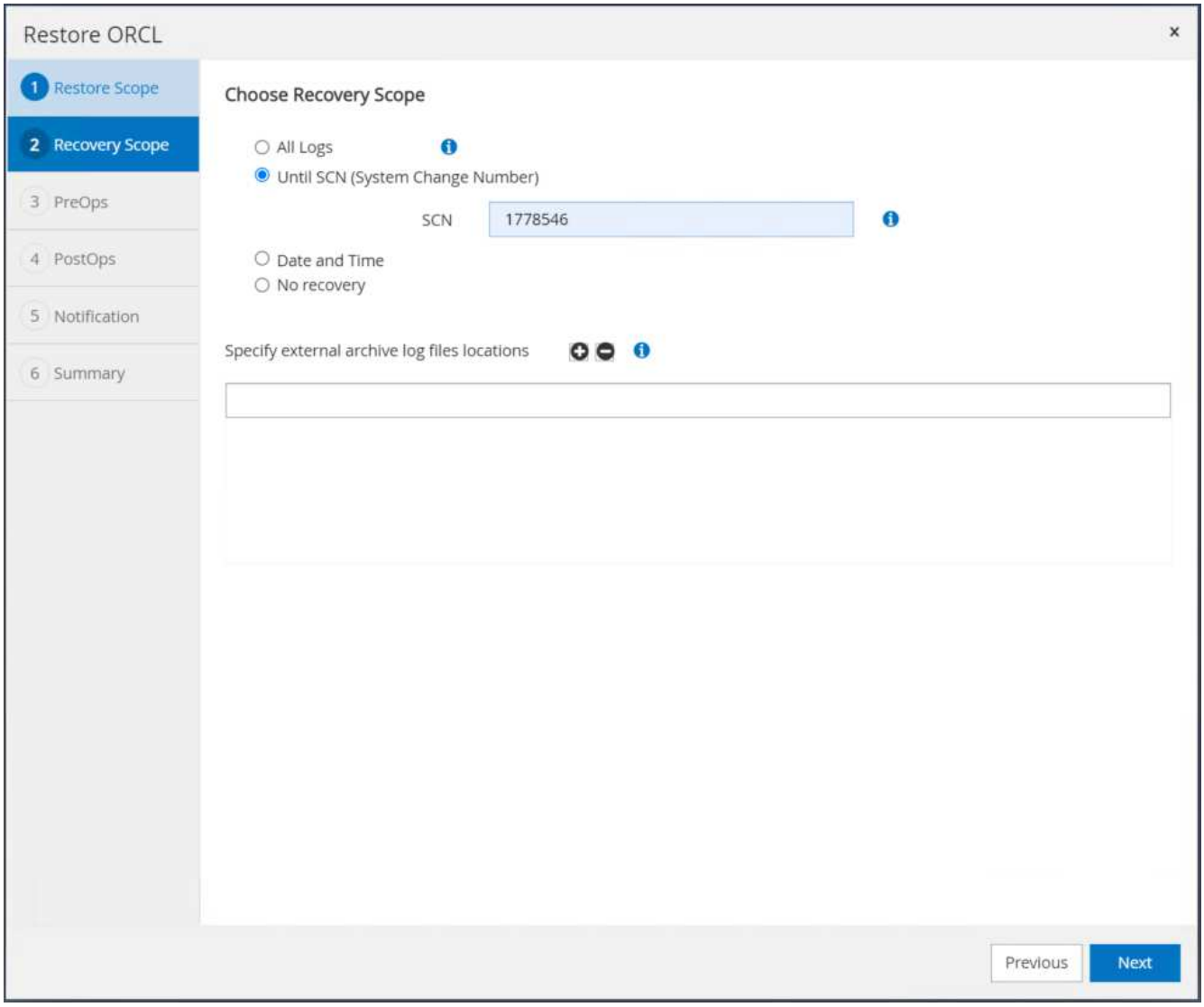


8. Check "All Datafiles" and "Change database state if needed for restore and recovery", and click Next.

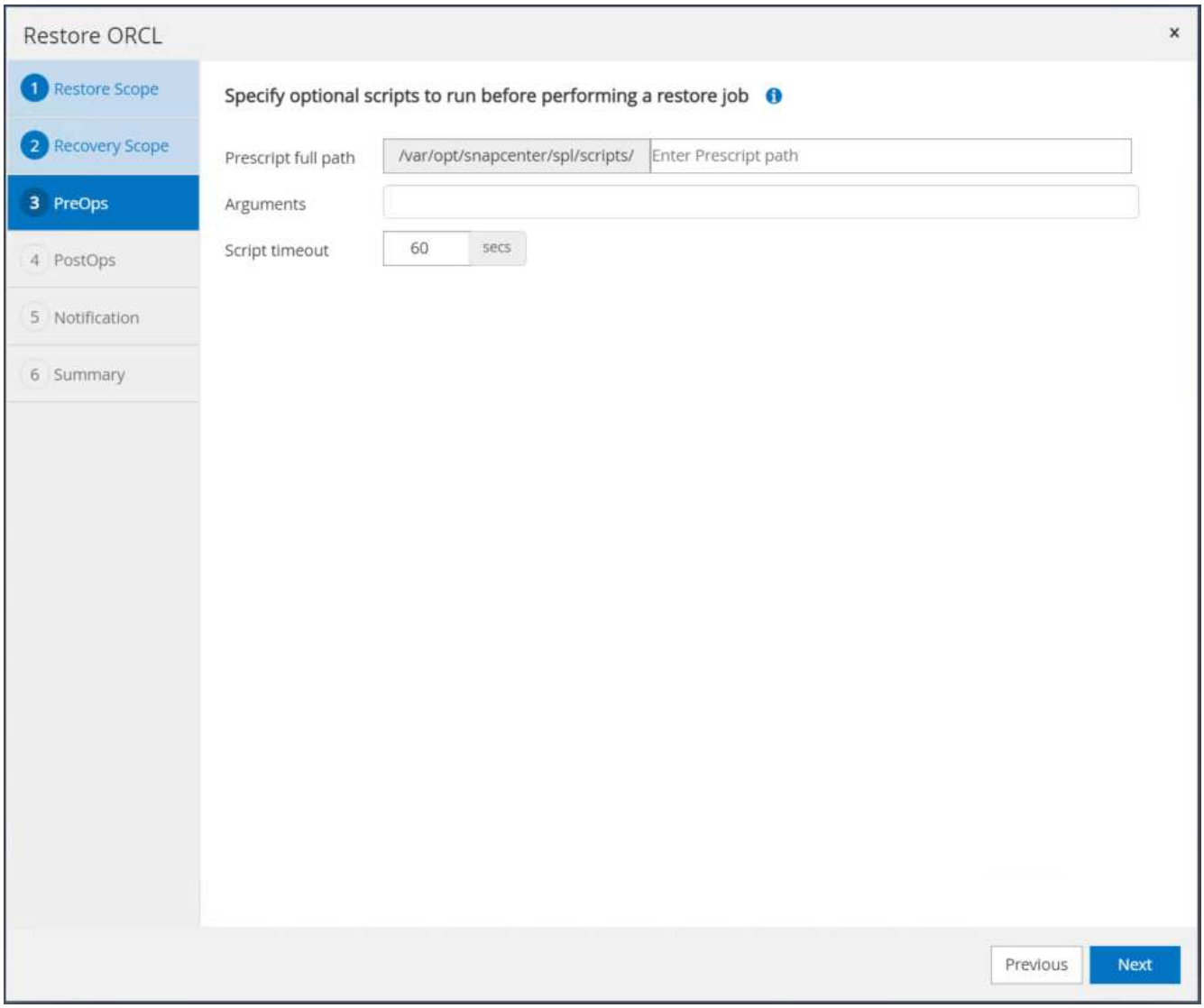


9. Choose a desired recovery scope using either SCN or time. Rather than copying the mounted archive logs to the current log directory as demonstrated in step 6, the mounted archive log path can be listed in

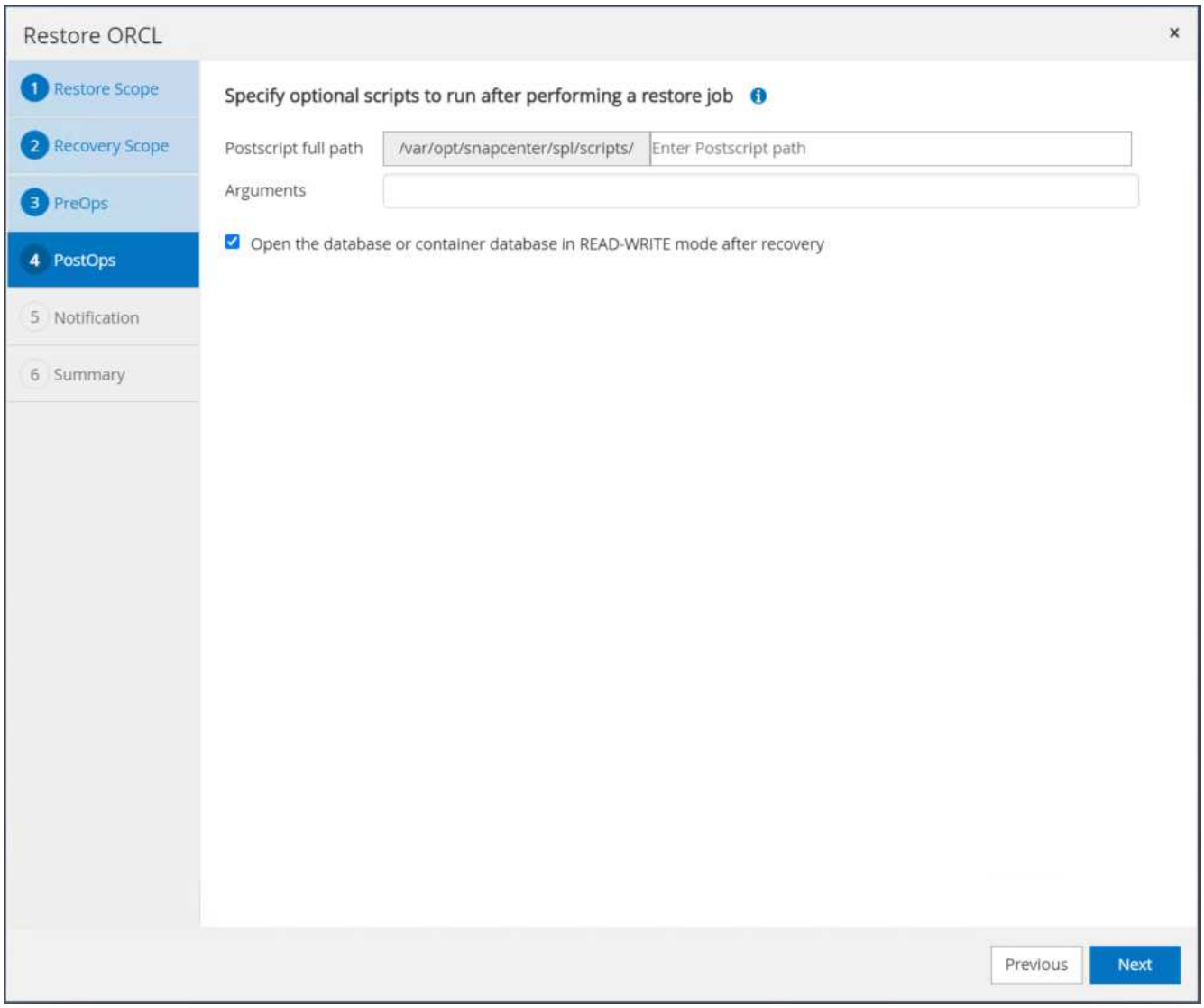
"Specify external archive log files locations" for recovery.



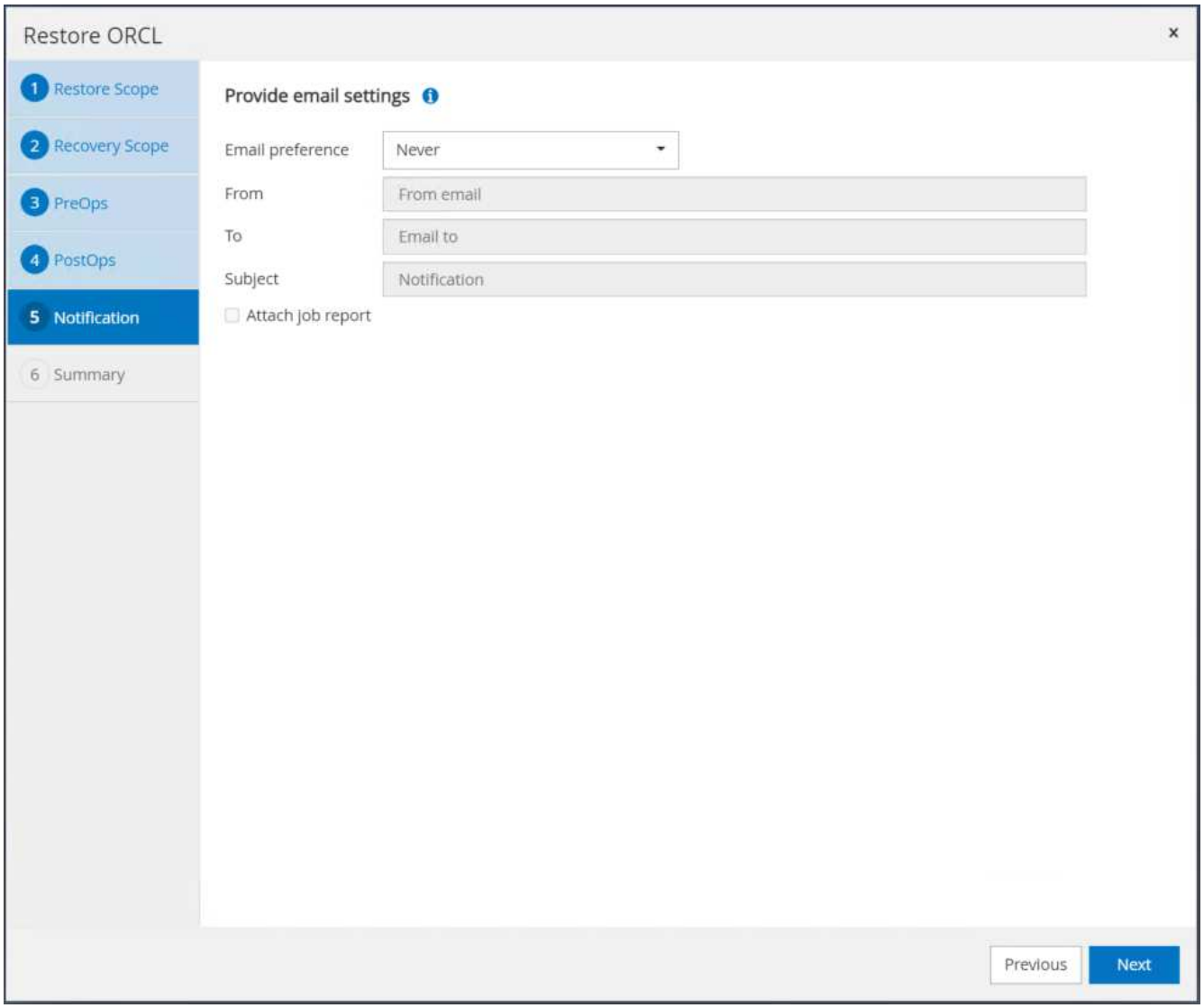
10. Specify an optional prescript to run if necessary.



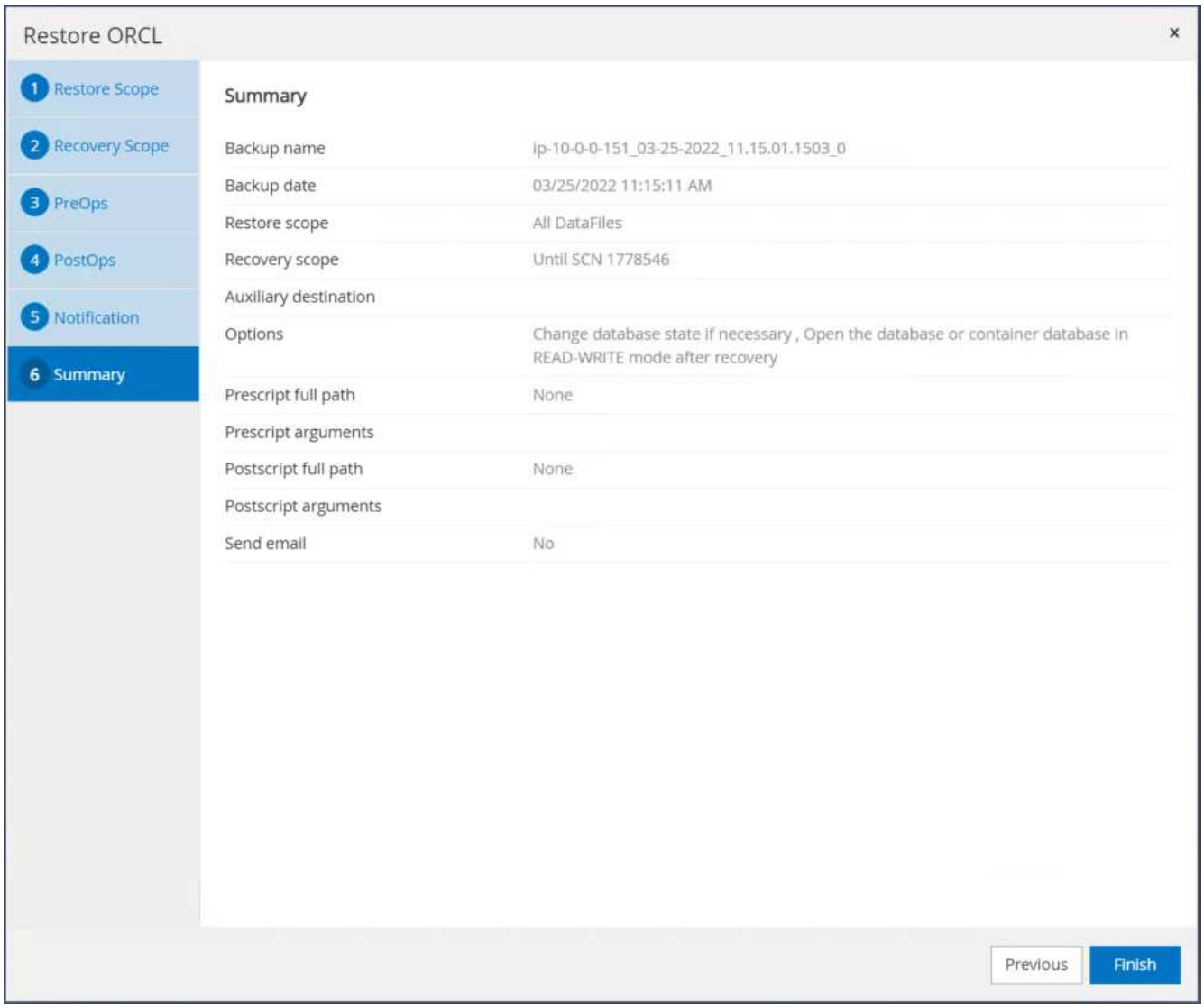
11. Specify an optional afterscript to run if necessary. Check the open database after recovery.



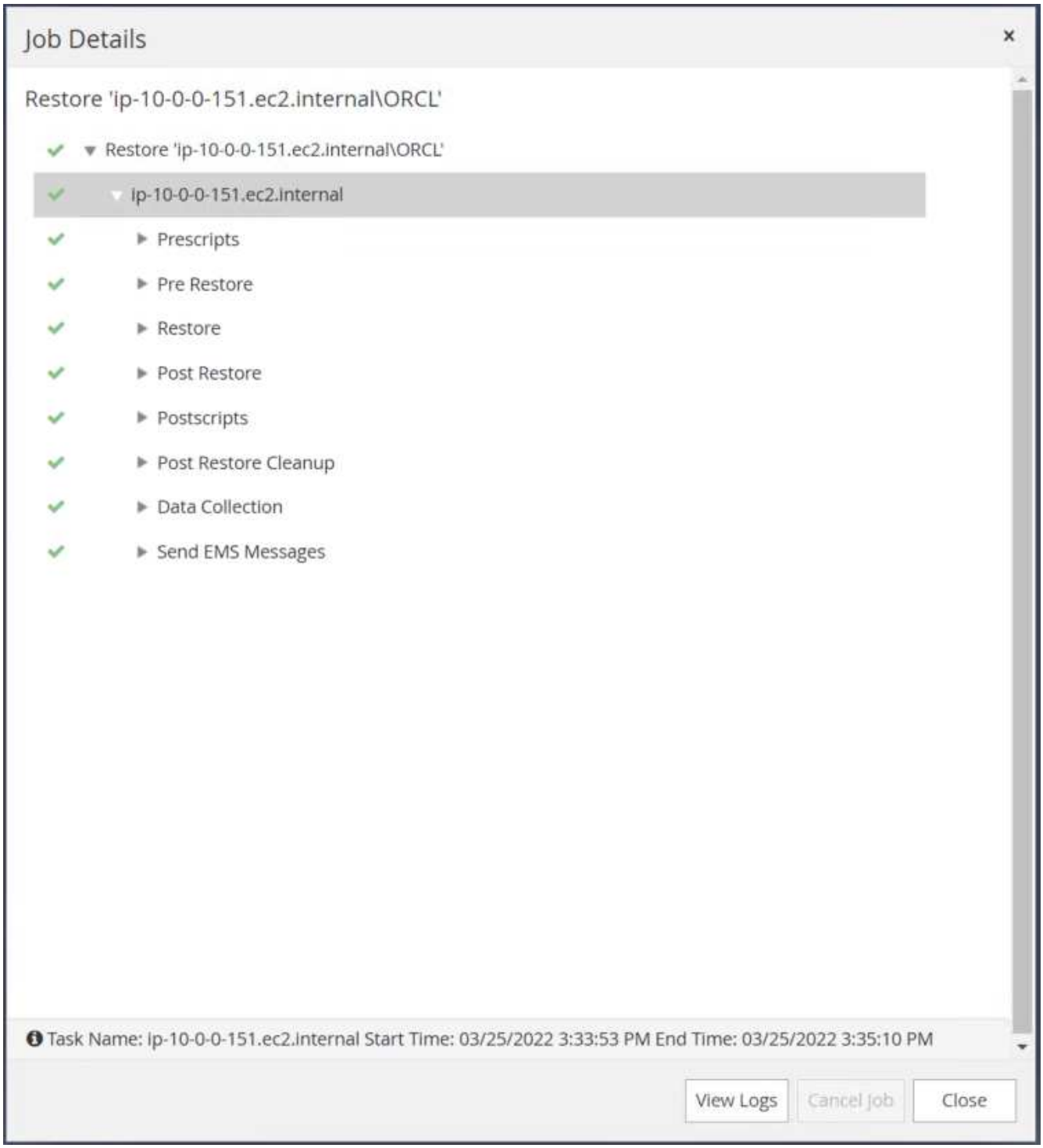
12. Provide an SMTP server and email address if a job notification is needed.



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



14. Validate the restore from SnapCenter.



15. Validate the restore from the EC2 instance host.

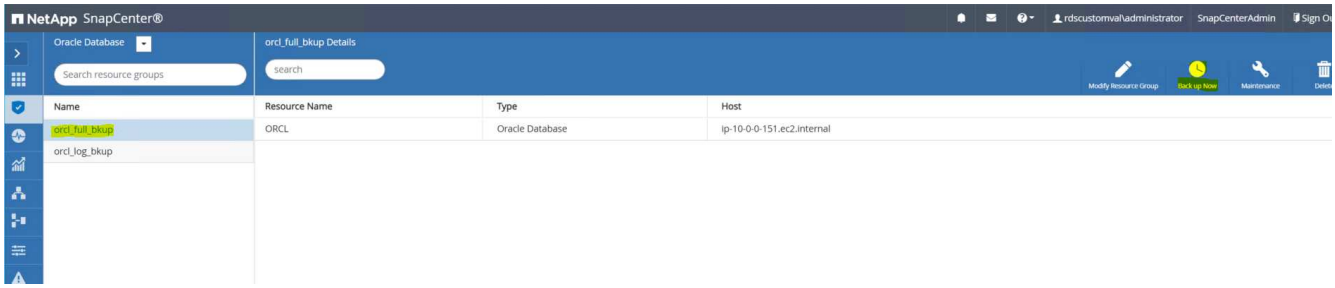
```
-bash-4.2$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 25 15:44:08 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2020, 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, RESETLOGS_CHANGE#, RESETLOGS_TIME, open_mode from v$database;

NAME      RESETLOGS_CHANGE# RESETLOGS OPEN_MODE
--------- ----------------- --------- --------------------
ORCL                1778547 25-MAR-22 READ WRITE

SQL>
```

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

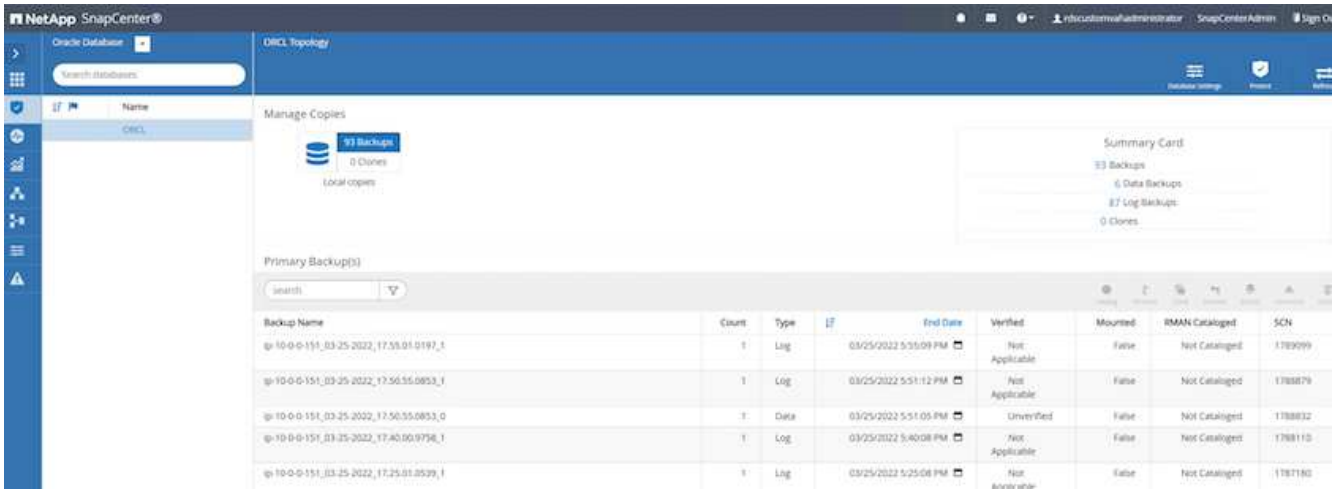
Creating a database clone

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.

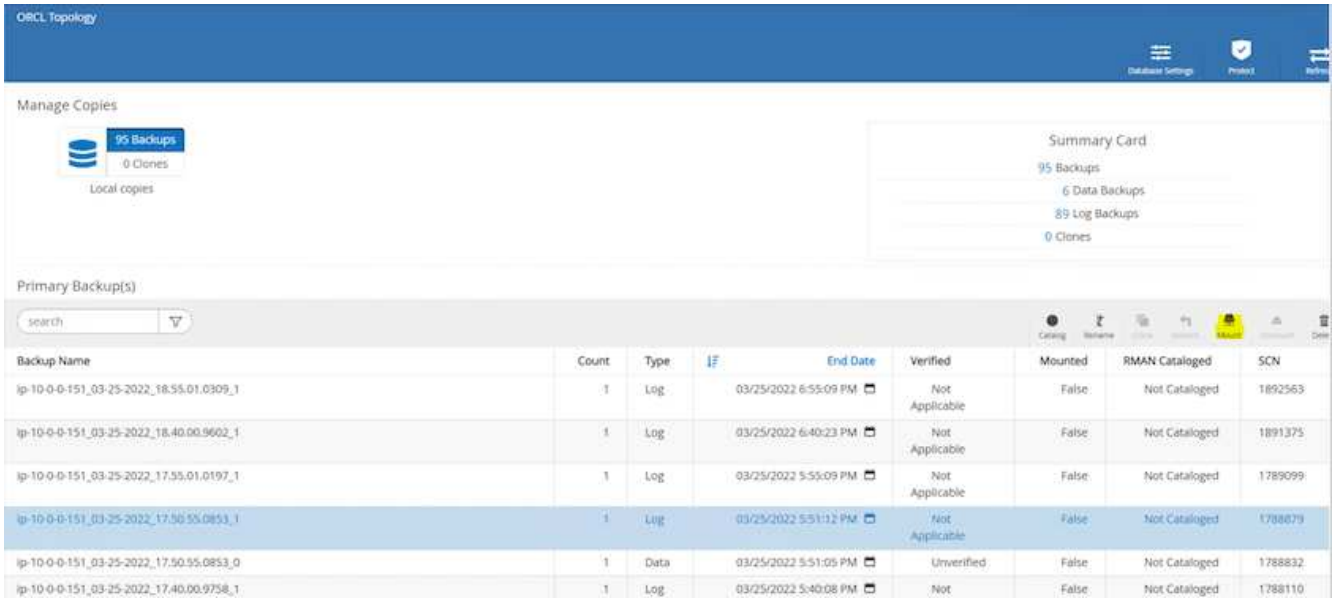
1. Take a full snapshot backup of the primary database from SnapCenter using the full backup resource group.

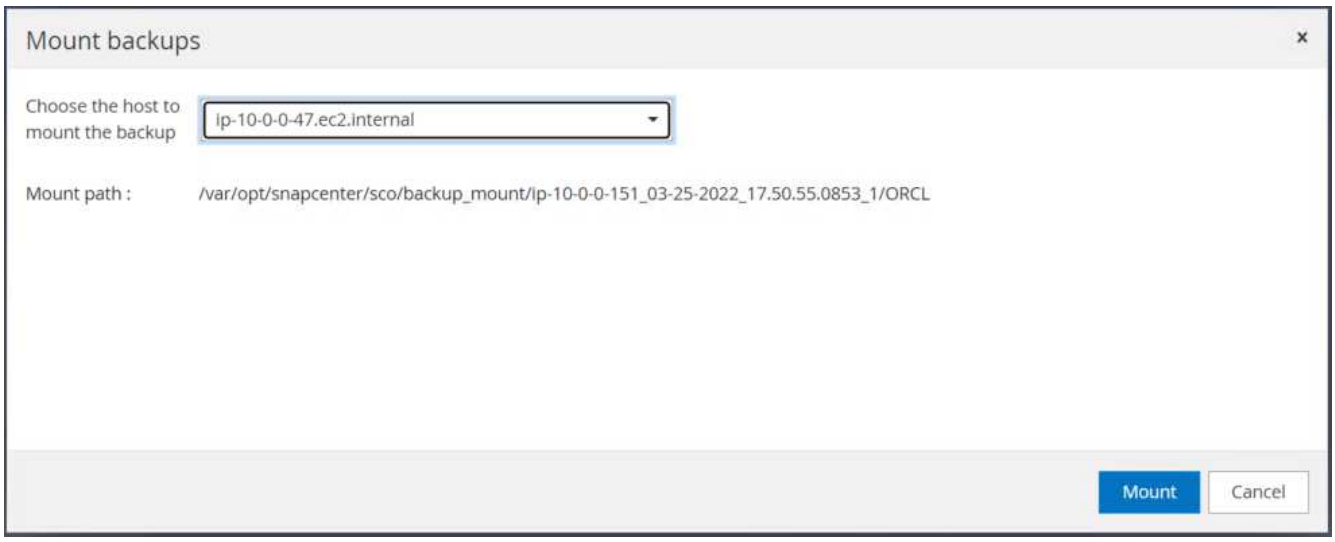


2. From the SnapCenter Resource tab > Database view, open the Database Backup Management page for the primary database that the replica is to be created from.

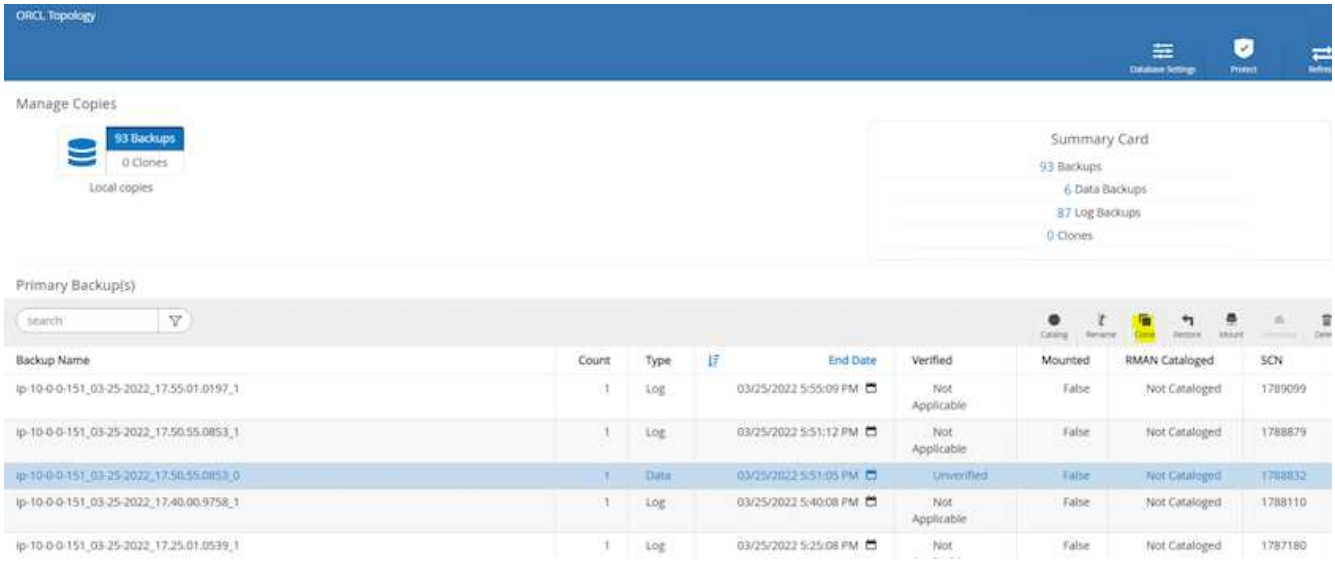


3. Mount the log volume snapshot taken in step 4 to the standby EC2 instance host.

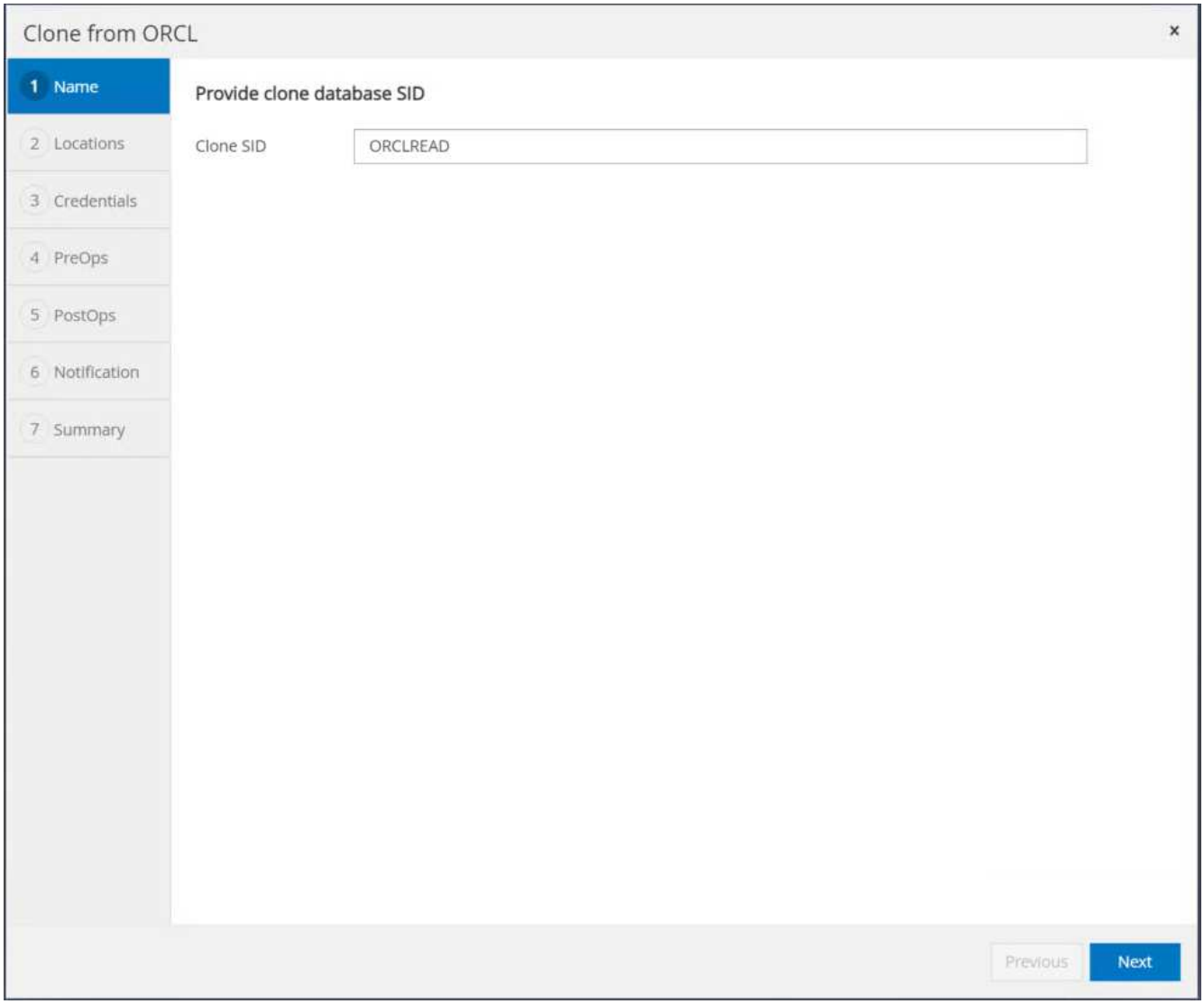




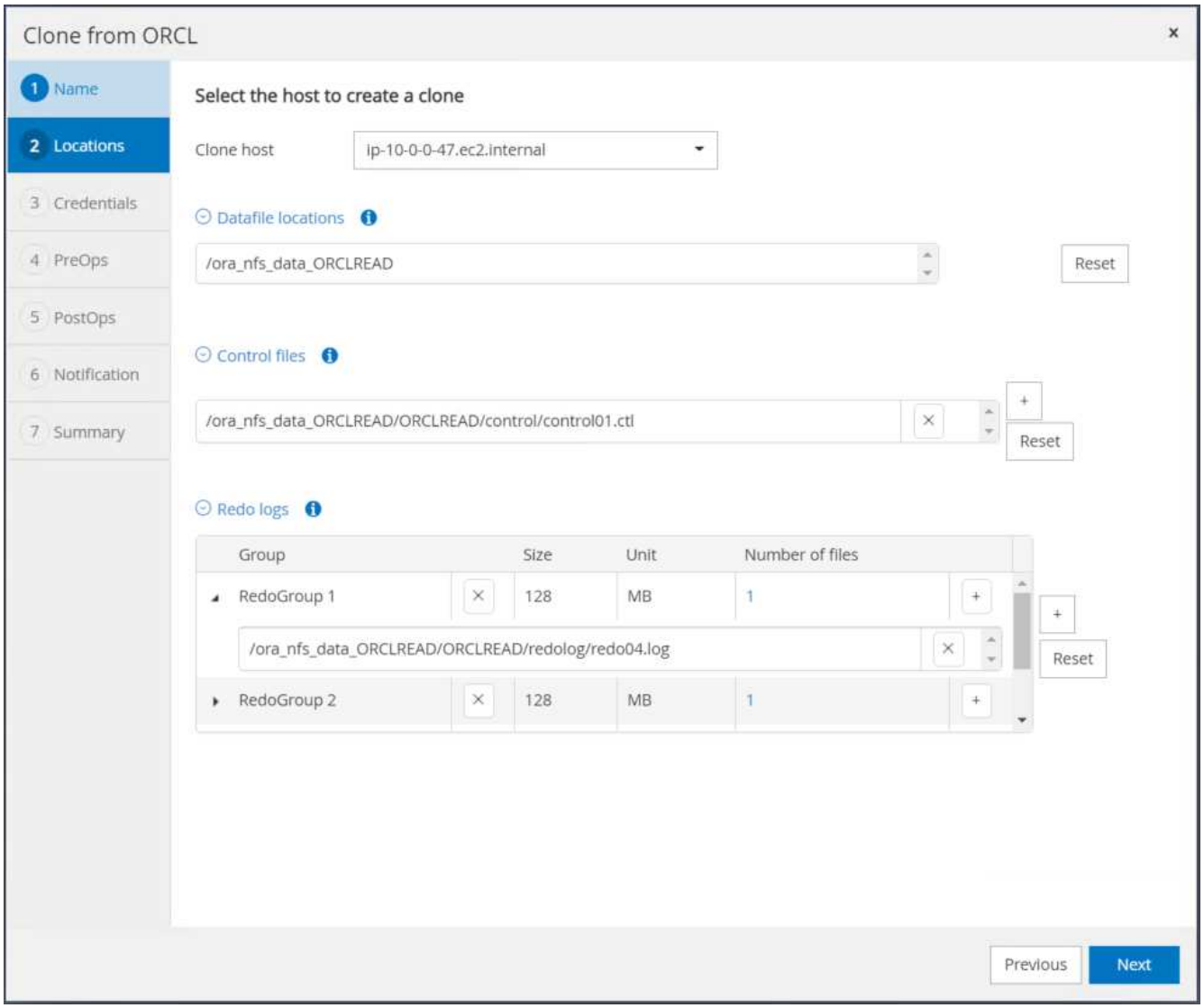
4. Highlight the snapshot copy to be cloned for the replica, and click the Clone button to start the clone procedure.



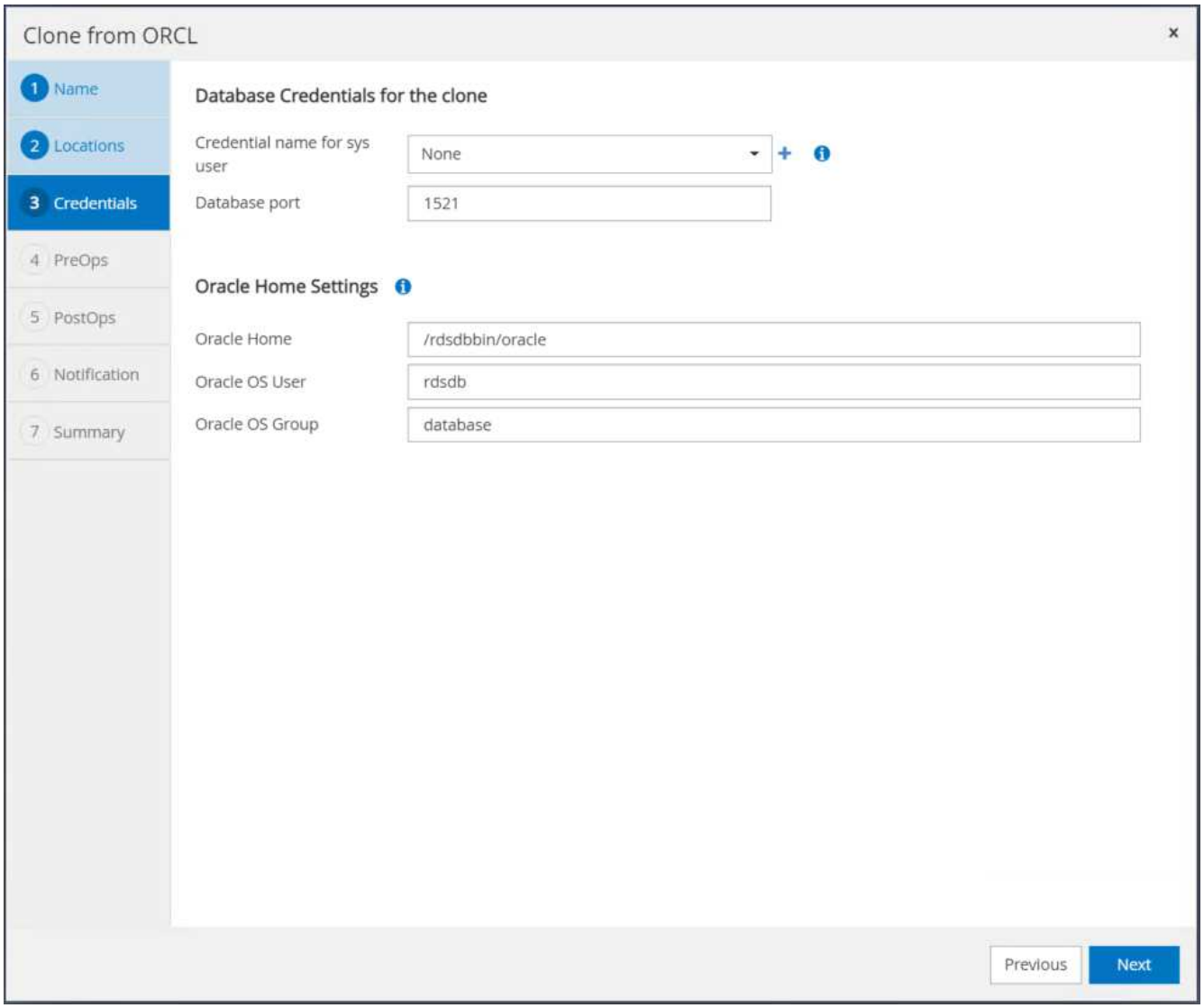
5. Change the replica copy name so that it is different from the primary database name. Click Next.



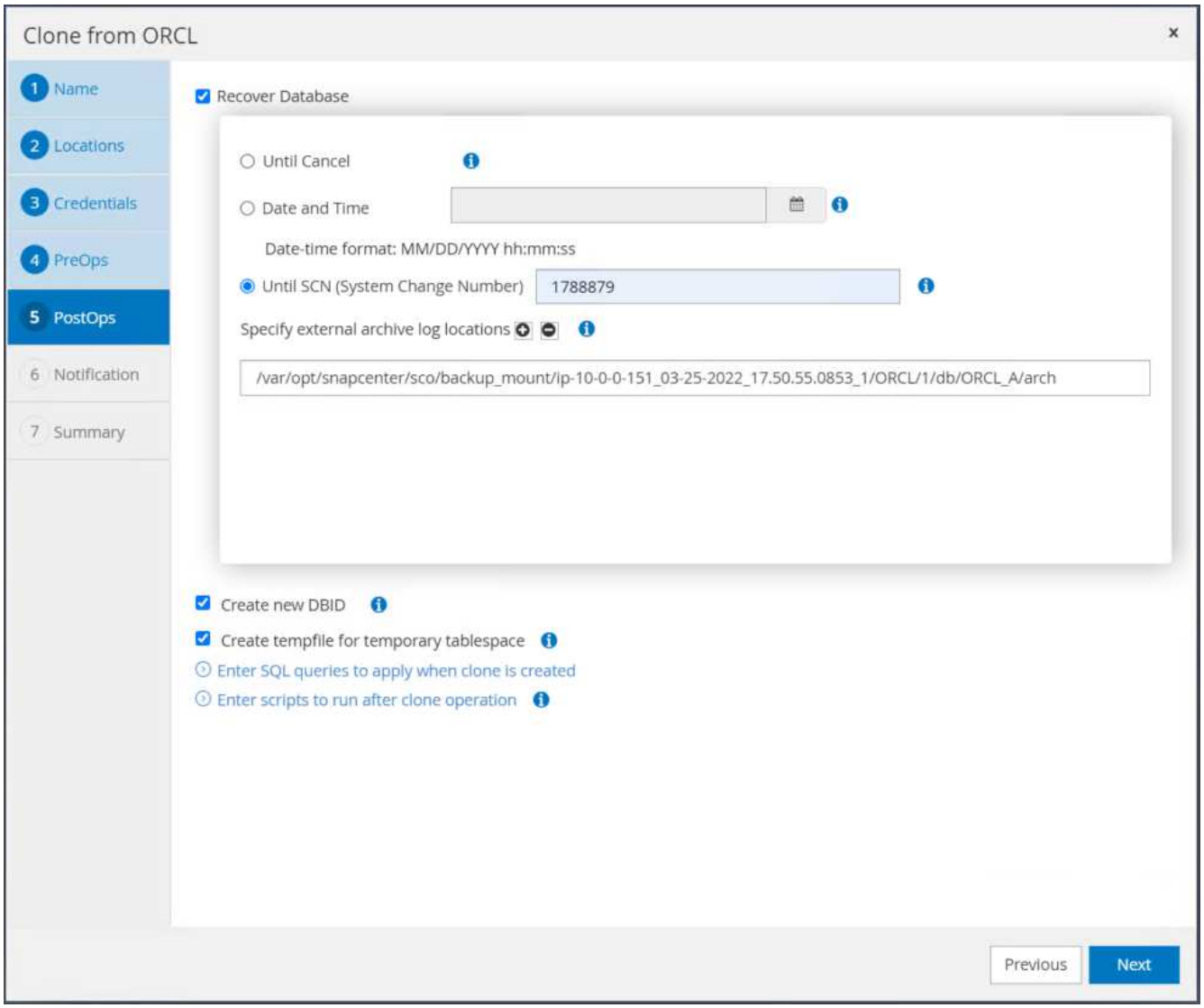
6. Change the clone host to the standby EC2 host, accept the default naming, and click Next.



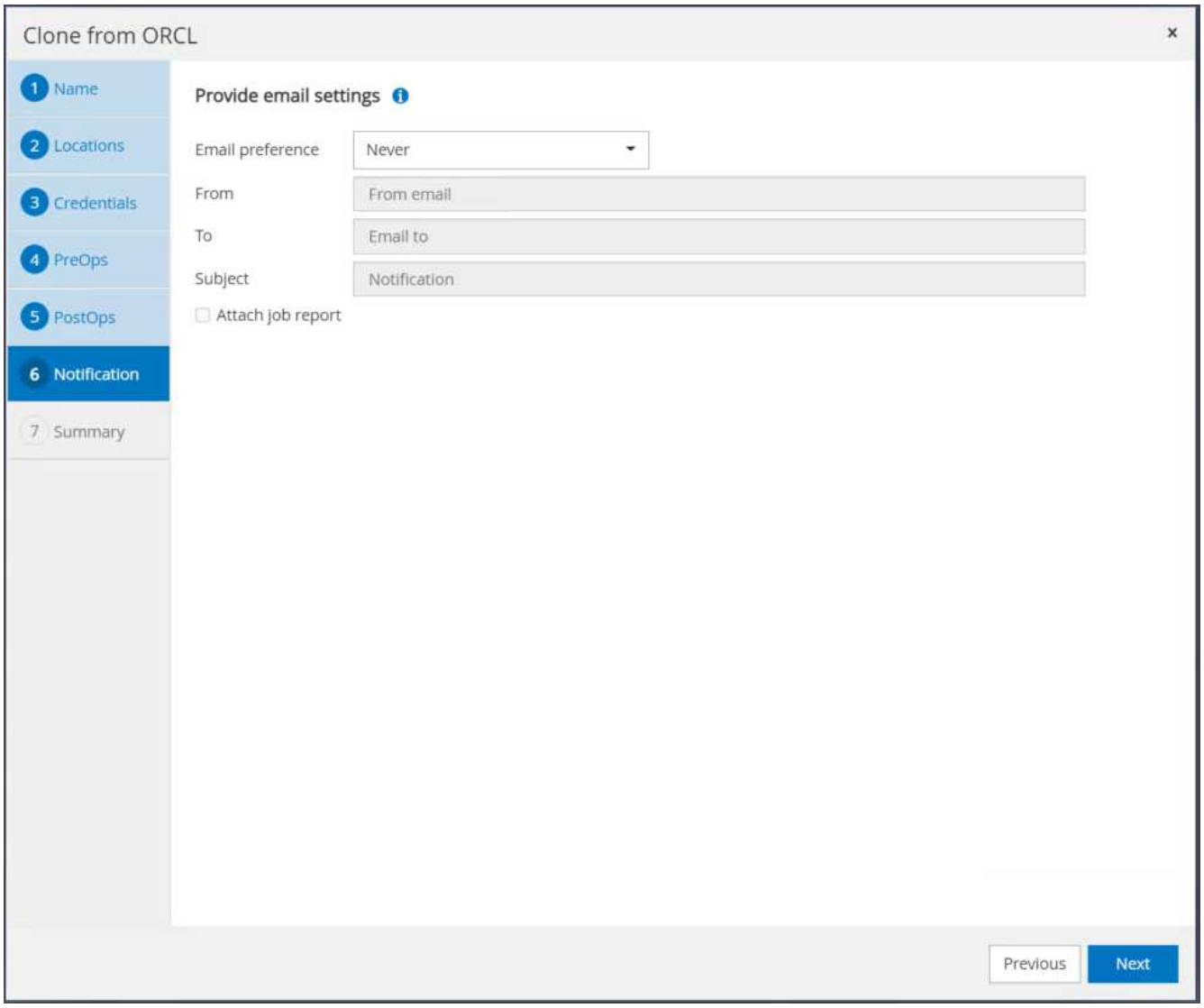
7. Change your Oracle home settings to match those configured for the target Oracle server host, and click Next.



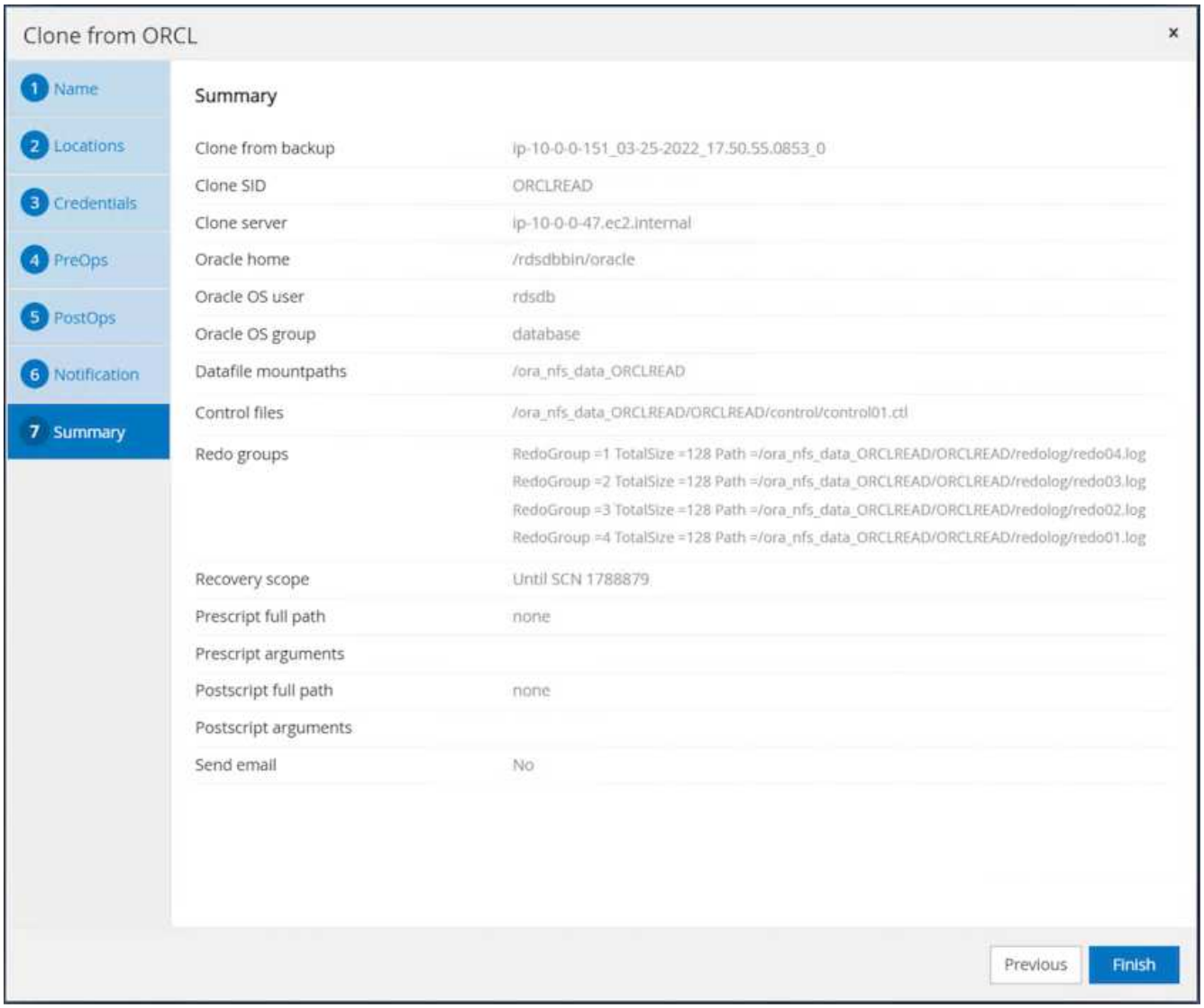
8. Specify a recovery point using either time or the SCN and mounted archive log path.



9. Send the SMTP email settings if needed.

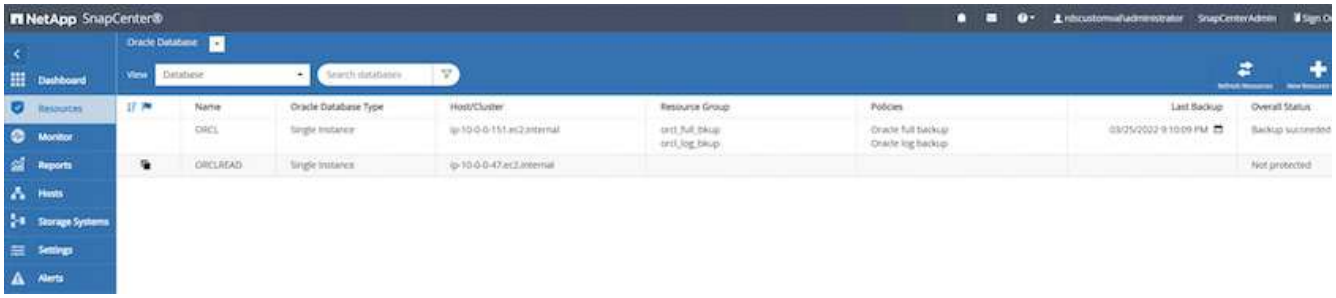


10. Clone the job summary, and click Finish to launch the clone job.



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

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;
```

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 SnapMirror technology that is built into AWS FSx ONTAP storage. The migration process can be orchestrated using NetApp BlueXP console.

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 the Oracle sqlplus CLI interface, execute an Oracle online log switch and allow SnapMirror sync to push the last archived log to the target volume.
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.

The following video demonstrates how to migrate an Oracle database from on-premises to AWS FSx/EC2 using the NetApp BlueXP console and SnapMirror replication.

Migrate on-prem Oracle DB to AWS

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.

Migrate on-premises Oracle databases to AWS FSx/EC2 using PDB relocation with maximum availability

This migration approach is best suited to Oracle databases that are already deployed in PDB/CDB multitenant model, and ONTAP storage is not available on-premises. The PDB relocation method utilizes Oracle PDB hot clone technology to move PDBs between a source CDB and a target CDB while minimizing service interruption.

First, create CDB in the AWS FSx/EC2 with sufficient storage to host PDBs to be migrated from on-premises. Multiple on-premises PDBs can be relocated one at a time.

1. If the on-premises database is deployed in a single instance rather than in the multitenant PDB/CDB model, follow the instructions in Converting a single instance non-CDB to a PDB in a multitenant CDB to convert the single instance to multitenant PDB/CDB. Then follow the next step to migrate the converted PDB to CDB in AWS FSx/EC2.
2. If the on-premises database is already deployed in the multitenant PDB/CDB model, follow the instructions in Migrate on-premises Oracle databases to cloud with PDB relocation to perform the migration.

The following video demonstrates how an Oracle database (PDB) can be migrated to FSx/EC2 using PDB relocation with maximum availability.

Migrate on-prem Oracle PDB to AWS CDB with max availability

Although the instructions in step 1 and 2 are illustrated in the context of Azure public cloud, the procedures are applicable to AWS cloud without any changes.

The NetApp Solutions Automation team provides a migration toolkit that can facilitate Oracle database migration from on-premises to the AWS cloud. Use following command to download the Oracle database migration toolkit for PDB relocation.

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

Copyright information

Trademark information