WHITE PAPER

Using FlashArray ActiveDR with Oracle Databases

Simplify Oracle Database recovery with continuous, storage-level replication

Contents

Executive Summary

Businesses rely on Oracle Database to drive performance, ensure customer satisfaction, and maintain operational continuity. Protecting this critical data while minimizing downtime and ensuring consistency is essential for business resilience.

Traditional disaster recovery solutions can be complex, expensive, and slow to respond, often falling short of modern recovery time and recovery point objectives. Pure Storage® FlashArray™ ActiveDR™ delivers a modern alternative: continuous, near synchronous replication with near zero recovery point objectives. Replication can occur within a single data center or across geographically distributed sites, supporting a wide range of architectural needs.

With simple, always-on continuous replication, ActiveDR enables non-disruptive failovers and failbacks, seamless failover testing, and real-time data protection. Organizations can confidently validate their disaster recovery strategies, strengthen their resilience, and ensure business continuity, all without interrupting daily operations.

Solution Overview

FlashArray ActiveDR is a continuous, near synchronous storage replication solution. It provides always-on storage replication that maintains up-to-date secondary copies with minimal impact on primary performance.

While not integrated directly with Oracle Database, ActiveDR offers seamless cooperation—enabling Oracle environments to benefit from fast, reliable disaster recovery without changes to the database stack. By replicating data continuously at the storage level, ActiveDR supports near zero recovery point objectives and rapid failover capabilities.

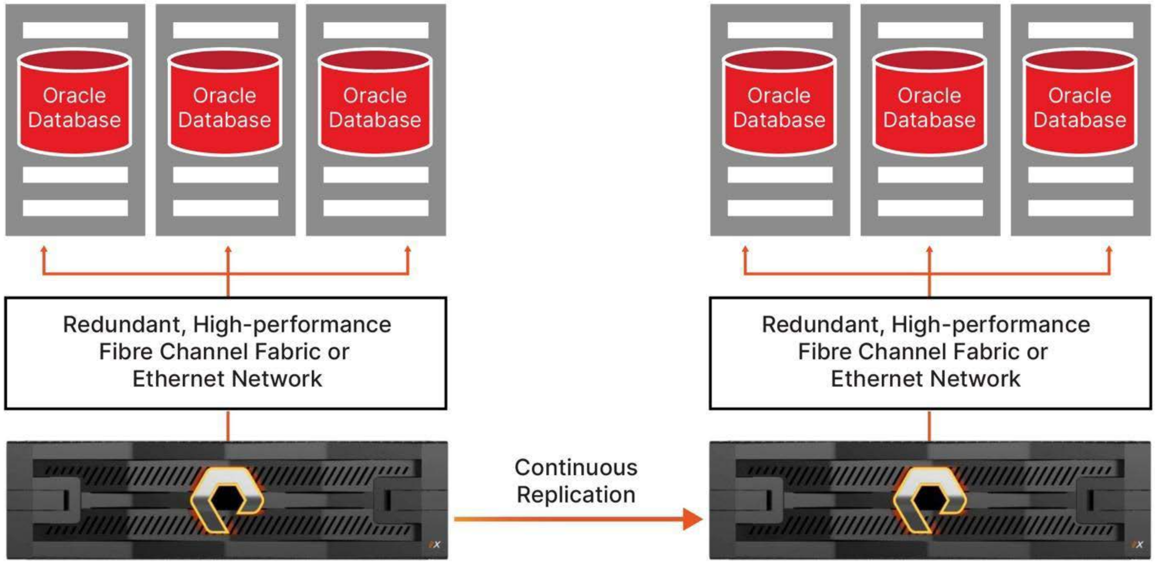

FIGURE 1 High-level overview of Oracle Database, FlashArray, and ActiveDR.

This solution includes the following:

- Oracle databases running on either bare metal or virtual machines
- FlashArray storage using block storage protocols
- A local or wide area network between arrays

Solution Benefits

ActiveDR offers unique advantages for Oracle customers looking to strengthen their disaster recovery strategy with storage-level replication that complements Oracle-native capabilities.

- **Included with FlashArray**: ActiveDR is available at no additional cost—offering immediate DR value without requiring extra licensing. Note: Oracle Data Guard does not require additional licensing, but Active Data Guard does.
- **Complementary to Oracle Database solutions**: ActiveDR operates at the storage level and can be used alongside Oracle Data Guard to offer an added layer of resilience. For example, while Data Guard replicates at the database layer, ActiveDR ensures storage-level protection across sites, providing options for broader failover and recovery scenarios.
- **No need for Oracle on the standby site**: Because ActiveDR replicates volumes directly, there's no requirement to run Oracle on the DR target until a failover is initiated. This reduces infrastructure and licensing overhead for organizations that prefer to defer Oracle instantiation until necessary.
- **Efficient data transfer**: Unlike Oracle Data Guard, which typically does not compress data during transport, ActiveDR uses built-in compression. In testing, ActiveDR data packages were smaller than uncompressed Oracle redo streams, reducing bandwidth usage and improving replication efficiency.
- **Minimal impact on production environments**: ActiveDR offloads replication to the FlashArray system, introducing no additional CPU load on the Oracle Database. In contrast, Data Guard processing has been observed to consume up to ~3% CPU in certain environments.
- **Simple and non-disruptive testing**: ActiveDR supports test failovers that do not interrupt replication or impact RPO/RTO metrics. This allows teams to validate recovery workflows without impacting production or backup SLAs.
- **Fast and flexible failovers**: With a single command, administrators can fail over Oracle-protected volumes, including all associated snapshots, enabling rapid recovery at the storage level.
- **Near-zero recovery point objectives**: Continuous asynchronous replication ensures that Oracle data is kept nearly up to date on the secondary array, enabling rapid and precise recovery after a failure.
- **Supports remote Oracle use cases**: Replicated volumes can be easily made available to Oracle instances or RAC clusters at secondary sites, simplifying recovery operations without rigid networking or distance limitations.

Technology Overview

The following sections provide an overview of the technologies that are used in an ActiveDR environment for Oracle Database.

Pure Storage FlashArray

FlashArray is an all-flash storage solution that provides storage and database administrators with a fast, scalable, unified block- and file-storage platform that is ideal for high-performance Oracle databases.

By providing a unified interface and simple-to-use tools for storage administrators, FlashArray enables administrators to quickly and seamlessly replicate, move, and manage data. FlashArray also deduplicates and compresses all data before it is written, efficiently reducing its size without impacting performance. Storage and database administrators can further increase storage capacity by using the FlashArray snapshot capabilities to create snapshots of production databases. They can then use those snapshots in development or testing environments.

The FlashArray family consists of the following:

- **FlashArray//XL**™: Provides high-performance storage at scale that helps reduce the number of arrays needed to run large applications
- **FlashArray//X**™**:** Provides high-performance, high-capacity storage that is ideal for performance-oriented workloads
- **FlashArray//C**™: Provides low-latency storage for capacity-oriented workloads

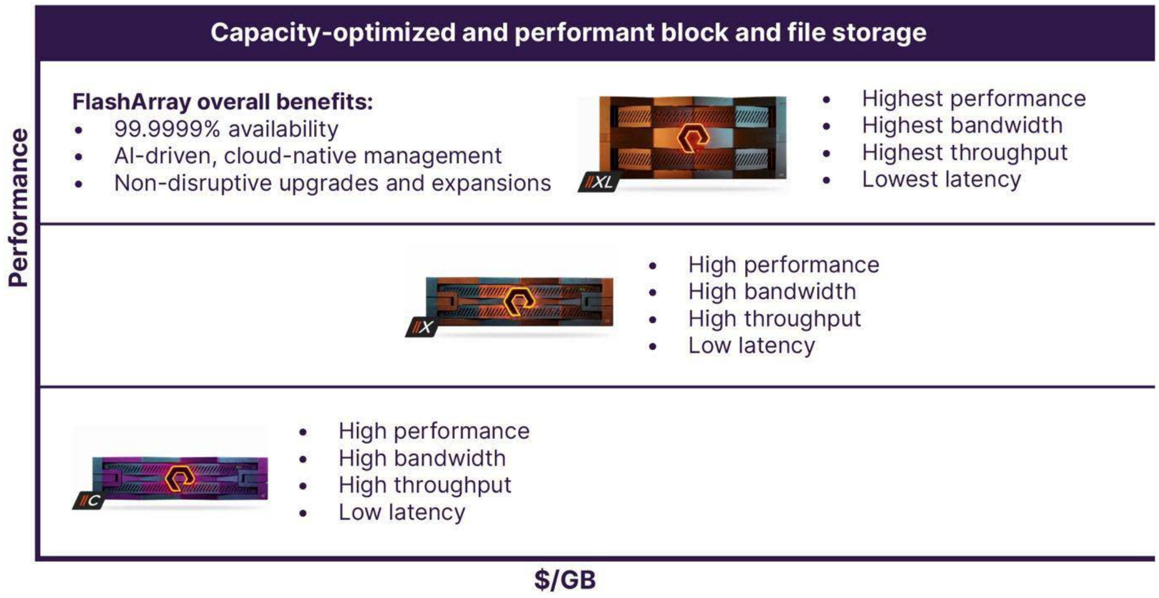

FIGURE 2 FlashArray is tailored to meet diverse business needs and workloads through several distinct offerings.

ActiveDR

ActiveDR provides near synchronous storage replication between two FlashArray systems within or across disparate data centers to protect against threats such as hardware failures, ransomware attacks, and user errors. ActiveDR enables a near-zero recovery point objective that improves business resilience for critical application infrastructure compared to traditional asynchronous replication.

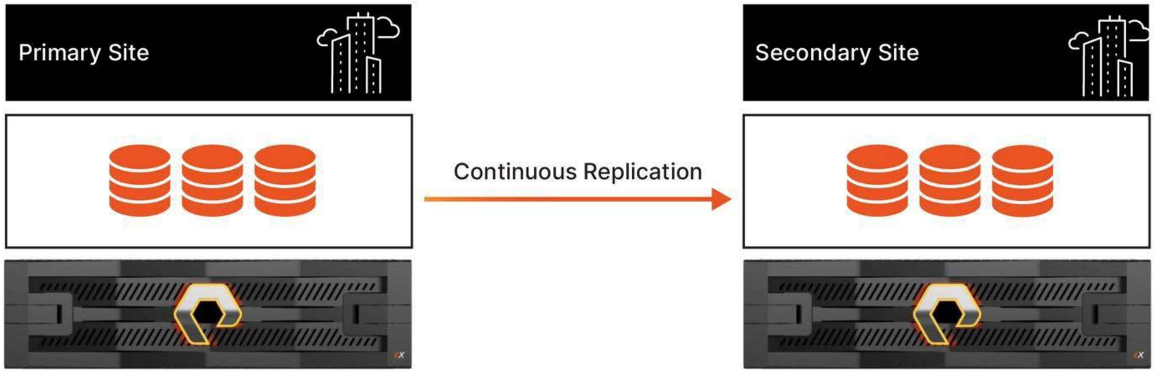

FIGURE 3 Near-synchronous storage replication between two FlashArray systems with ActiveDR.

ActiveDR simplifies disaster recovery by providing continuous data protection and enabling non-disruptive testing of disaster recovery workflows. Organizations can perform test failovers, live failovers, resynchronization, and failbacks without disrupting production or halting replication, which helps ensure operational continuity while validating recovery readiness.

ActiveDR uses near-synchronous replication, making it ideal for latency-sensitive Oracle workloads and geographically distributed environments. Unlike synchronous replication, it doesn't require remote acknowledgment of writes, allowing for efficient use of existing wide area networks without performance trade-offs.

ActiveDR consists of three components:

- **Pods**: These are storage-management containers that organize storage objects and configuration settings into groups that are failed over and failed back as a unit. A pod can contain volumes, volume snapshots, and protection groups. Additionally, a pod can contain configuration settings such as protection group snapshot schedules, snapshot retention policies, and quality-of-service volume limits.
- **Replica links**: These provide replication between pods and provide directional and auto-reversing capabilities. Once a replica link is created, ActiveDR is automatically enabled.
- **Connected FlashArray systems**: ActiveDR requires a minimum of two FlashArray systems connected over a network to replicate data between. With no latency requirements, these systems can be within the same data center or as far apart as on different continents.

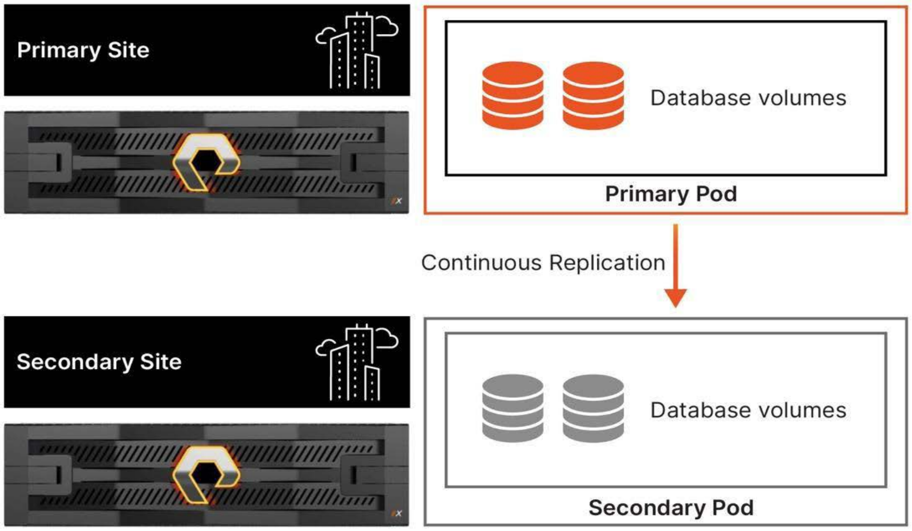

FIGURE 4 A typical ActiveDR deployment that includes pods and replica links between FlashArray systems.

Pod-based replication helps simplify storage management across sites by replicating all configuration changes made on a primary site array to the secondary site array, which helps simplify management and disaster recovery storage failovers. ActiveDR also supports multi-directional replication for different pods. For example, a database administrator might have a pod at their primary site that they want to replicate to their secondary site, while the secondary site might contain a pod that they want to replicate back to their primary site. ActiveDR lets them easily configure the pods to replicate in either direction between sites.

Differences Between ActiveCluster, Asynchronous Replication, and ActiveDR

FlashArray has several storage replication functions that can be used to protect an organization's data. Among them are synchronous replication, which is used by the ActiveCluster™ solution, asynchronous replication storage array volume snapshots, and near-synchronous replication with ActiveDR.

ActiveCluster

ActiveCluster uses synchronous replication to maintain copies of data between two FlashArray systems. When data is written to a primary site FlashArray, it is simultaneously copied to a secondary site FlashArray. Once the data is written to both arrays, the write is acknowledged to the host system. This method of replication is recommended when the latency between arrays is 11ms or less, which means that ActiveCluster should be used between arrays in the same data center, or between data centers that have very-low-latency wide area network capabilities. For more information about ActiveCluster, see ActiveCluster Solution Overview.

Asynchronous Replication

Asynchronous replication is a snapshot-based solution that uses space-efficient snapshots to replicate data between FlashArray storage devices, while ActiveDR is a streaming-based solution that continuously replicates volume data between FlashArrays at different sites. When asynchronous replication is enabled on a volume at the primary site FlashArray, a snapshot of the volume is created on the primary site array and then replicated to the secondary site array. The first snapshot transfer is a baseline, which is a complete copy of the entire contents of the volume. All subsequent transfers are incremental transfers that result by comparing existing data on the storage array with the newly created snapshot to determine what data to send to the secondary site array. For more information about asynchronous replication, see FlashArray Asynchronous Replication Configuration and Best Practices Guide.

ActiveDR

ActiveDR offers near-synchronous, pod-based replication, making it ideal for disaster recovery scenarios where data must be replicated across long-distance or high-latency sites. Unlike ActiveCluster (which requires low latency for synchronous replication) or traditional asynchronous replication (which has higher recovery point objectives), ActiveDR provides a balance—delivering near real-time replication without strict latency dependencies or host-level confirmation. However, its performance can still be impacted by extreme latency, and it does continuously replicate data in pod-based consistency groups.

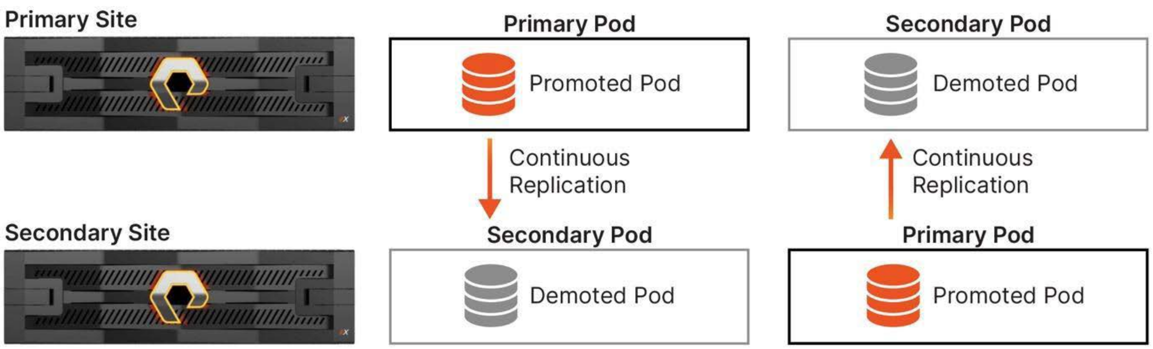

FIGURE 5 Pods in either primary or secondary sites can be replicated to either site.

For more information about ActiveDR, visit the Pure Storage support site.

Oracle Database

Oracle Database is a relational database management system ideal for businesses to efficiently store, retrieve, and manage large amounts of data. It uses SQL for querying and managing data, and it supports a wide variety of applications and integrations, from small to enterprise-level systems. Oracle databases use multitenant architecture to become multitenant container databases, which can contain user-created pluggable databases and application containers.

Within this white paper, Oracle Database is considered the core database platform that supports an organization's business applications and services and that will contain the data being replicated by ActiveDR.

ActiveDR Disaster Recovery Use Cases for Oracle Database

This section covers additional features and use cases administrators can use to protect Oracle Database data with ActiveDR.

ActiveDR and Oracle Database

With Oracle Database, users have the option to store data in several formats, including Oracle Automatic Storage Management (ASM) volumes and a standard file system. Regardless of how the data is stored, ActiveDR can replicate those volumes to a secondary location to support business continuity and disaster recovery strategies. Importantly, changes to ASM disk group configurations—which are not replicated by Oracle Data Guard—are fully captured by ActiveDR's storage-level replication.

For disaster recovery scenarios between data centers, ActiveDR replicates only the changed data, continuously and efficiently, from the primary to the secondary FlashArray system. This enables significantly faster recovery times for Oracle Database compared to traditional backup solutions. Because volumes are replicated at the storage level, businesses gain the flexibility to test failovers non-disruptively and remain fully operational—even in the face of outages or major site failures.

ActiveDR and Oracle Real Application Clusters

Oracle RAC is a database clustering solution that uses Oracle Clusterware to bind multiple servers together so they operate as a single system for high availability, scalability, and load balancing. Oracle Clusterware works with ASM to create a clustered pool of storage that can be used by any combination of non-cluster and Oracle RAC databases.

Database and storage administrators can use ActiveDR to replicate their Oracle RAC databases at the storage level to a secondary site, typically on a different subnet. Oracle RAC nodes can also be configured at the secondary site so the replication site can seamlessly take over for the primary, ensuring business continuity.

In this scenario, two or more Oracle RAC nodes utilize a shared ASM storage volume, where the Oracle datafiles, control files, parameter files, and redo log files are stored. The shared ASM storage volumes are placed into a FlashArray pod, which is then replicated to a remote FlashArray using ActiveDR.

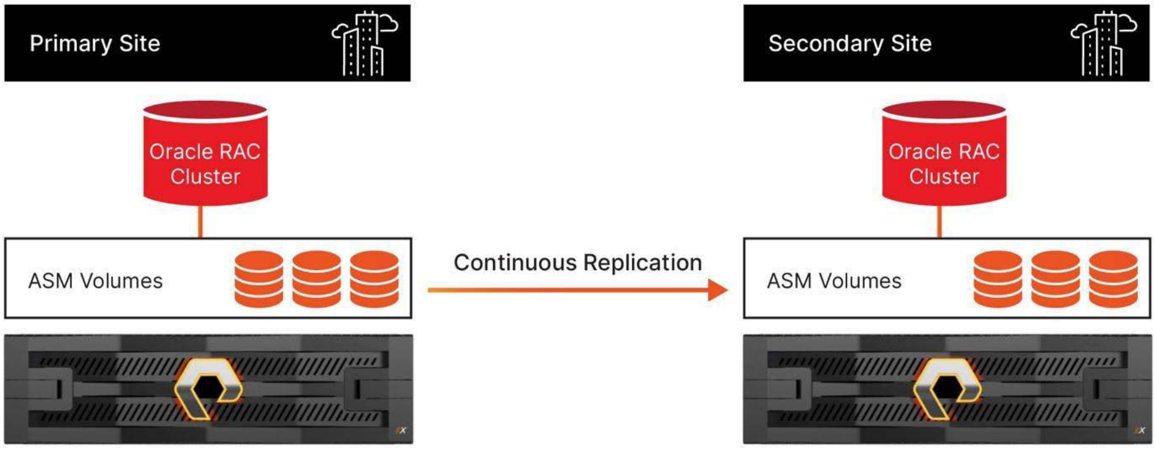

FIGURE 6 ActiveDR replication between two Oracle RAC environments.

If all instances of an Oracle RAC database fail, then Oracle Database automatically recovers the instances the next time one instance opens the database. To avoid a loss of data if the primary site goes offline, database administrators can also configure secondary Oracle RAC nodes at another location for failover via ActiveDR. In a disaster recovery event where the primary site is offline, such as during a natural disaster, the replicated ASM volumes at the remote site can be attached to a secondary Oracle RAC cluster, or the nodes replicated with ActiveDR can be brought online at the remote site. The remote Oracle RAC nodes must be preconfigured to recognize the replicated storage, allowing application requests to resume with minimal disruption to business continuity.

Because ActiveDR replicates everything on the volume—including Oracle binaries, configuration files (such as init.ora, spfile, listener files), and any other system-level or environment-specific changes—it provides a more complete recovery scenario compared to solutions that replicate only database-level activity.

Once the primary site is brought back online, storage administrators can reverse the ActiveDR replication from the secondary site back to the primary site and then fail back to the primary site at the time of their choosing.

VMware vSphere Integration with Oracle Database

FlashArray provides integration with VMware vSphere, which lets storage and database administrators replicate complete virtual machines to a remote location. Oracle also offers its own open-source virtualization solution, known as VirtualBox. In this scenario, FlashArray volumes are configured as VMware datastores (VMware vSphere Virtual Volumes [vVols] are not supported) that are attached to VMware ESXi hosts. These volumes contain files for virtual machines and any attached virtual disks.

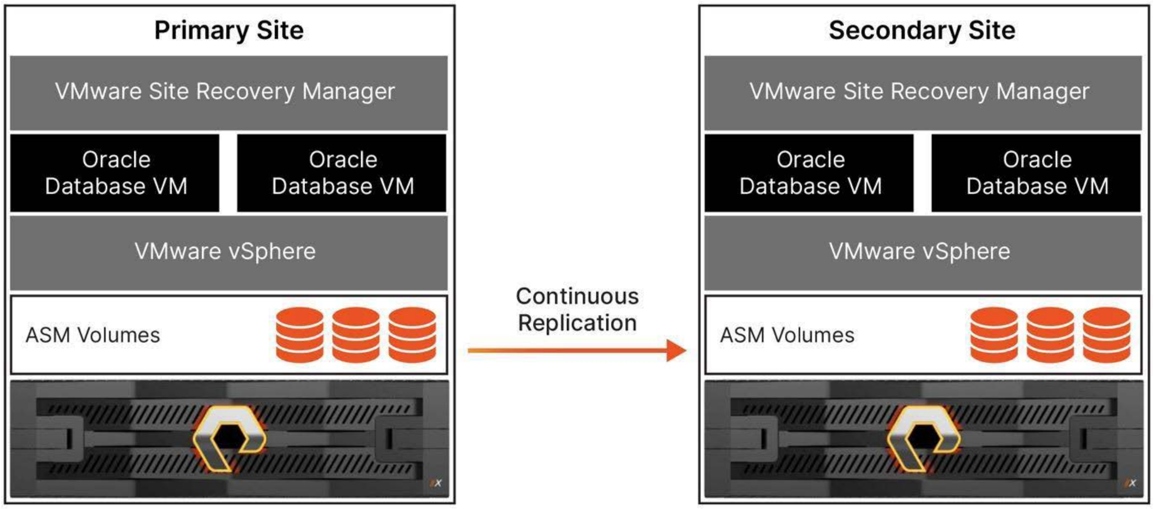

FIGURE 7 A virtualized Oracle Database stack that uses FlashArray.

Database and storage administrators can configure Oracle Database in virtual machines by creating separate virtual disks on FlashArray volumes. These volumes will contain the virtual machine, ASM volumes, and container databases/pluggable databases. If desired, ActiveDR can replicate a virtual machine and its contents to a remote site to be used with a remote VMware ESXi host.

For more information about ActiveDR and VMware, see ActiveDR with VMware User Guide.

Configuring ActiveDR with Oracle Database

The following sections describe how to configure and test ActiveDR with Oracle Database. These sections assume that the primary and secondary sites are already configured to work with FlashArray pods. For more information, see Setting Up Oracle Disaster Recovery Using Purity ActiveDR.

While these sections describe configuring and testing ActiveDR using the Pure Storage user interface, administrators can also use PowerShell scripts to achieve the same goals. ActiveDR and other Pure Storage scripts are located on GitHub.

Configuring ActiveDR

The following steps must be completed using the Pure Storage user interface to initiate ActiveDR replication. To ensure a seamless implementation, ensure the Oracle Database configuration matches the structure shown in Figure 8.

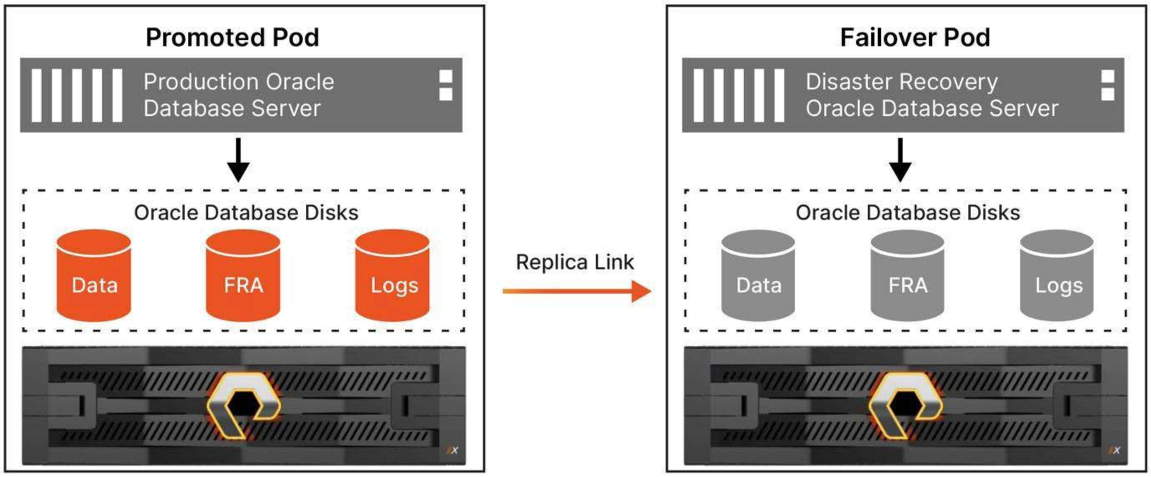

FIGURE 8 Oracle Database configuration for ActiveDR.

Note: In a pluggable database environment, data is stored under a container database structure. ActiveDR replicates the entire container database, including all associated pluggable databases. This means you cannot replicate individual pluggable databases directly unless they reside in separate container databases. ActiveDR replicates at the volume level, meaning all data stored within the replicated volume is included. Because Oracle's multitenant architecture stores multiple pluggable databases within the same container database's structure (and potentially the same volume), you must replicate the entire container databases.

Configuring ActiveDR for Oracle Database involves the following general steps:

1. Configure Oracle Database to store user database files on FlashArray volumes at the primary site.
2. Create a production pod.
3. Create a protection group within the pod.
4. Move volumes out of the current protection group or create new volumes.
5. Move volumes into the production pod.
6. Add volumes to the protection group in the production pod.
7. Create a remote pod.
8. Create a replica link.

Verify the Environment

Let's confirm the disks match the one we see on the primary Oracle Database server.

1. Confirm the Oracle Database volumes on the source primary pod. The two volumes used are **oraadr-data** and **oraadr-fra**, with serial numbers ending in **316A** and **316B**.

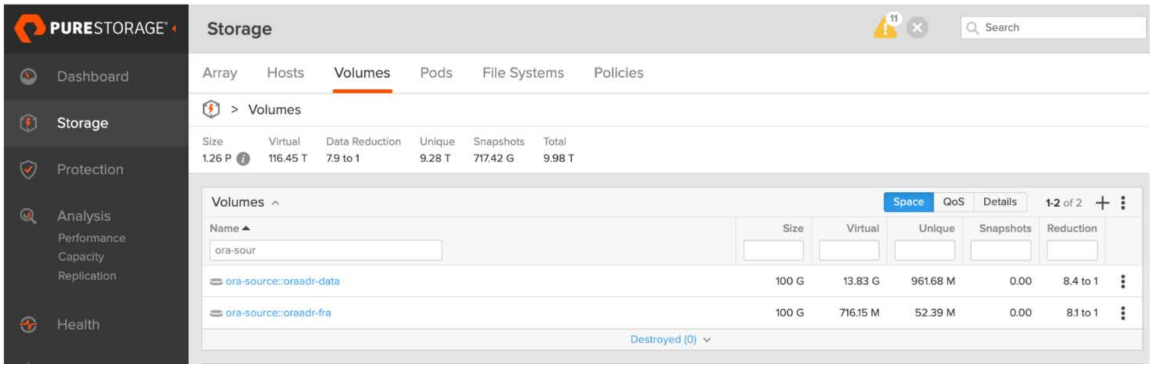

FIGURE 9 Verifying the volume names.

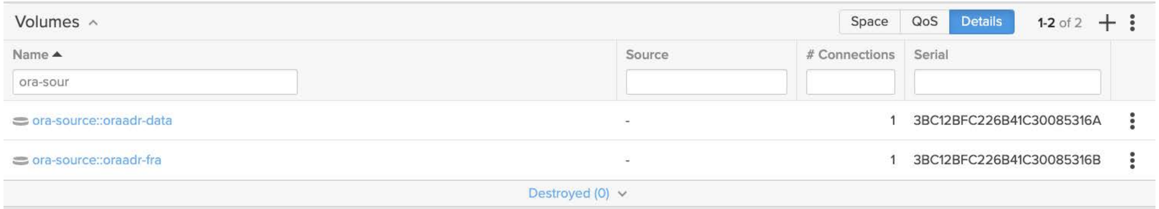

FIGURE 10 Verifying the serial numbers.

2. Next, using the **lsblk** command, confirm that the two drives, **sdb** and **sdc**, match the serial numbers from the user interface, with their disks mounted on **/u01** and **/u02.** In this example, we are using Oracle Database File System; however, it would work the same if we were using Oracle ASM for disk management.
3. We can see the correct disk path from inside the Oracle environment by using **SQL*PLUS** to query the names of the datafiles.

```
[root@oraadr1 ~]# lsblk -o name,serial
NAME           SERIAL
sda
sda1
sda2
sda3
ol-root
ol-swap
ol-home
sdb            3BC12BFC226B41C30085316A
sdb1
sdc            3BC12BFC226B41C30085316B
sdc1
sr0            00000000000000000001
```

```
[root@oraadr1 ~]# df -h
Filesystem           Size  Used Avail Use% Mounted on
devtmpfs             7.7G     0  7.7G   0% /dev
tmpfs                7.7G     0  7.7G   0% /dev/shm
tmpfs                7.7G   18M  7.7G   1% /run
tmpfs                7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/mapper/ol-root   61G  9.1G   52G  15% /
/dev/sda2           1014M  457M  558M  46% /boot
/dev/mapper/ol-home   30G  3.2G   27G  11% /home
/dev/sda1            599M  6.0M  593M   1% /boot/efi
tmpfs                1.6G   48K  1.6G   1% /run/user/54321
/dev/sdb1            100G   14G   87G  14% /u01
/dev/sdc1            100G  1.4G   99G   2% /u02
[root@oraadr1 ~]#
```

```
SQL> select name from v$datafile;
NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_system_mtodco38_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_sysaux_mtodfd7z_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_undotbs1_mtodg5d1_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_system_mtodgrxr_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_sysaux_mtodgrxt_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_users_mtodg6h5_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/datafile/o1_mf_undotbs1_mtodgrxw_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/2DD56AB3E54447A9E065025056971309/datafile/
o1_mf_system_mtodrh06_.dbf
/u01/app/oracle/oradata/oraadr/ORAADR/2DD56AB3E54447A9E065025056971309/datafile/
```

Now that we've verified the disk paths, we can take a look at the data. Let's create some tables on the source, generate data, and then confirm the data/tables that we created have been replicated to the target Oracle server.

4. To do this, we will use the SQL command **CREATE TABLES** to create **adr101**, **adr102**, and **adr103**, and to then populate them with data from **dba_objects.**

```
SQL> create table adr101  as select * from dba_objects;

Table created.

SQL> create table adr102 as select * from dba_objects;

Table created.

SQL>

SQL>  create table adr103  as select * from dba_objects;

Table created.

SQL>

SQL>

SQL> select count(*) from adr101;

  COUNT(*)

----------

     77010

SQL>

SQL>  select count(*) from adr102;

  COUNT(*)

----------

     77011

SQL>

SQL>  select count(*) from adr103;

  COUNT(*)

----------

     77012
```

Now we can create the pod on the source array.

Configure the ActiveDR Primary Site Pod in the Pure Storage User Interface

To configure the primary site pod:

1. From the primary site's Pure Storage user interface, select **Storage** in the navigation pane, and then select the **Pods** tab.

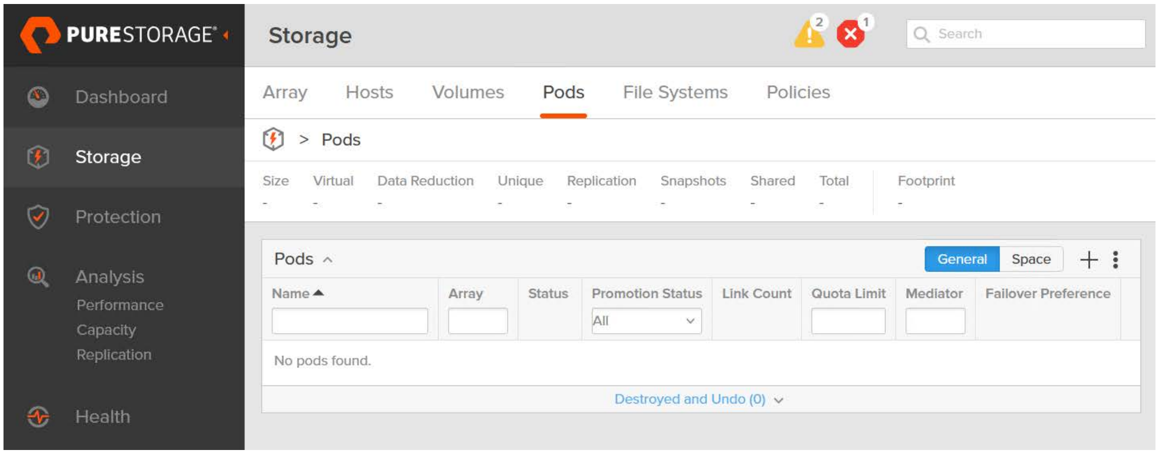

FIGURE 11 The Pods tab in the Pure Storage user interface.

2. Click the **+** icon in the **Pods** group to create a new pod.

3. In the **Name** field in the **Create Pod** dialog box, enter a name for the pod, and then click **Create**. The pod appears in the Pods group.

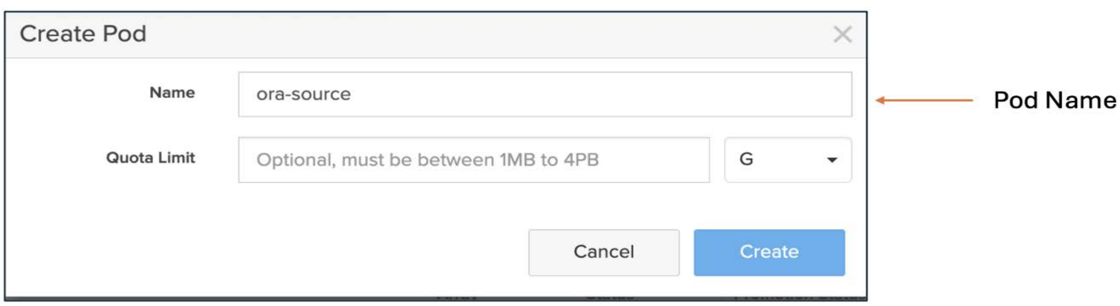

FIGURE 12 The Create Pod dialogue box.

4. In the **Pods** group, select the name of the pod that was created to display the **Pod management** view.
5. In the **Volumes** group, click the ellipsis, and then click **Move In** to display the **Move Volumes In** dialog box.

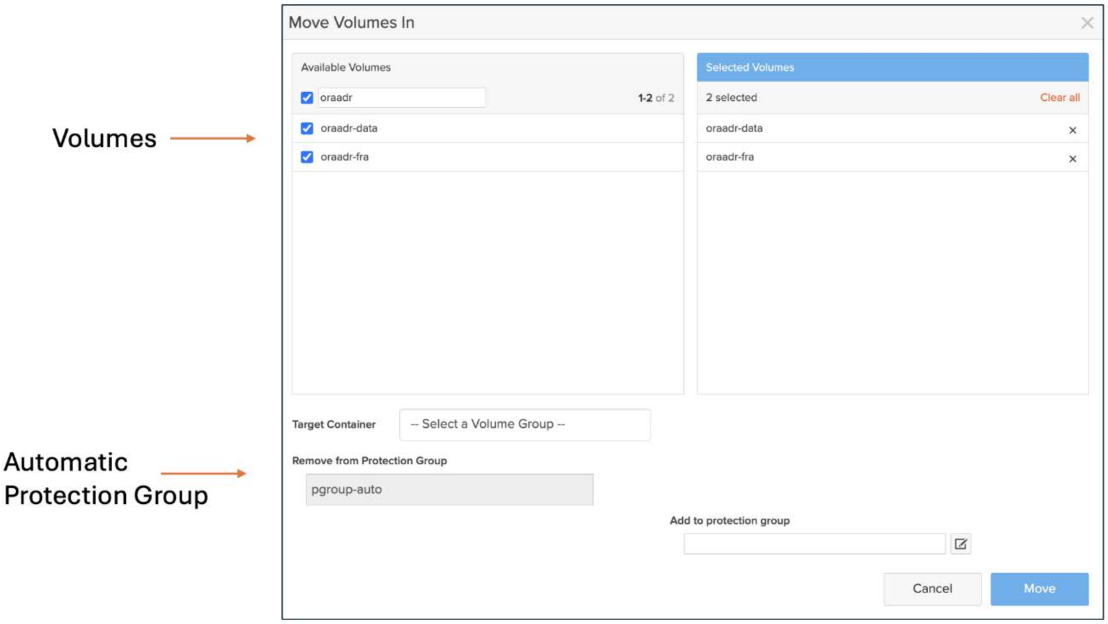

FIGURE 13 Moving volumes into a pod.

6. Select the Oracle Database ASM volumes that will be moved into the pod, and then click **Move**. The volumes appear in the **Volumes** group using the naming convention, **<pod name>::<volume name>**.

We can see from Figure 14 that the pod has been created, the volumes have been moved into the pod, and the default protection group has been created.

Note: Volumes can also be created by clicking the **+** icon under the **Volumes** section of the **Pod** view.

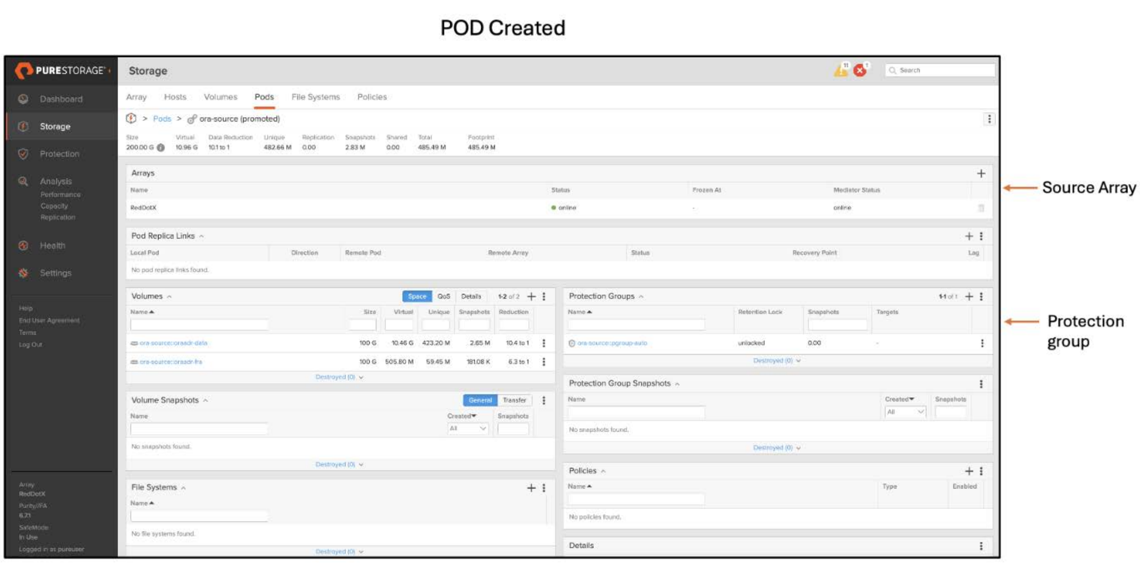

FIGURE 14 Confirming pod creation.

The primary site pod is now configured with the database files. ASM volumes that contain container databases, pluggable databases, database files, and redo log files should all be configured in an ActiveDR pod.

The next step is to create a replica link to the remote site's FlashArray, which enables ActiveDR replication.

Create a Replica Link in the Pure Storage User Interface

1. From the remote site's Pure Storage user interface, select **Protection** in the navigation pane.
2. In the **Array Connections** group, click the ellipsis, and then select **Get Connection Key**.

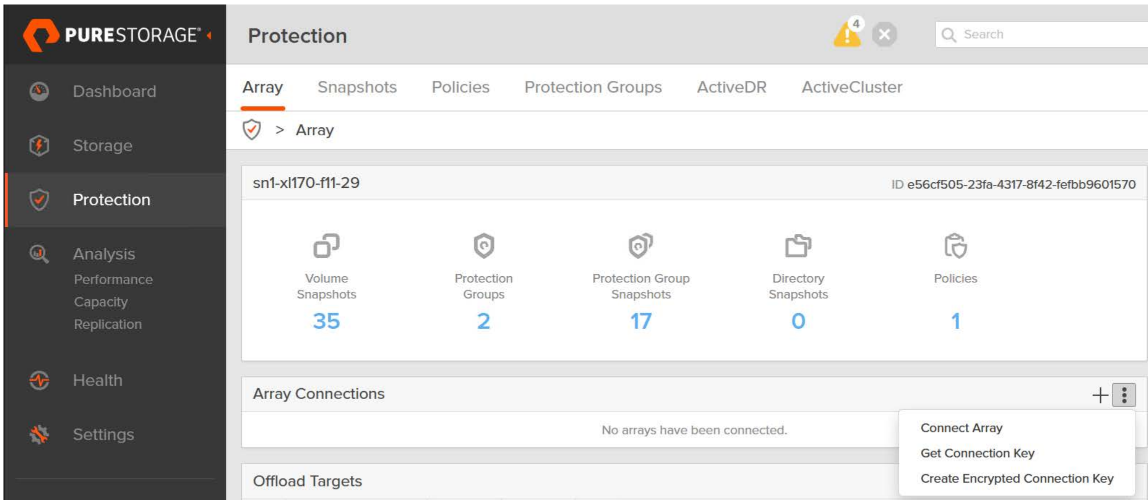

FIGURE 15 Getting a connection key on the Array tab.

3. In the **Connection Key** dialog, click **Copy** to copy the connection key to the clipboard.
4. From the primary site's Pure Storage user interface, click **Protection**.
5. In the **Array Connections** group, click the ellipsis, and then select **Connect Array**.
6. In the **Connect Array** dialog box, enter or select the following information:

a. **Management Address:** The IP address of the remote site's FlashArray.

b. **Type:** Select **Async Replication** from the drop-down list.

c. **Connection Key:** Paste the connection key that you copied from the remote site's Pure Storage user interface.

d. **Replication Transport:** Select **Ethernet (IP)** from the drop-down list.

e. **Encrypted:** Select the button to toggle replication encryption on or off.

7. Click **Connect.** The secondary site array appears in the **Array Connections** group.

FIGURE 16 Displaying the secondary site in the **Array Connection**s group.

8. In the **Pod Replica Links** group, click the ellipsis, and then select **Create**.

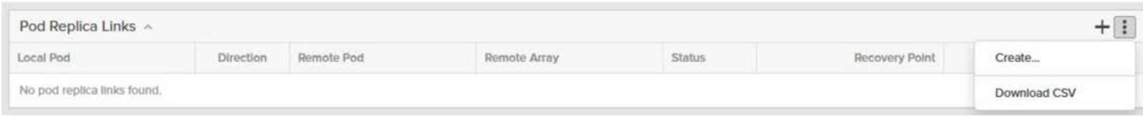

FIGURE 17 Creating a replica link.

9. In the **Create Replica Link** dialog box, select the remote site FlashArray from the **Remote Array** drop-down list, and then select a remote pod.

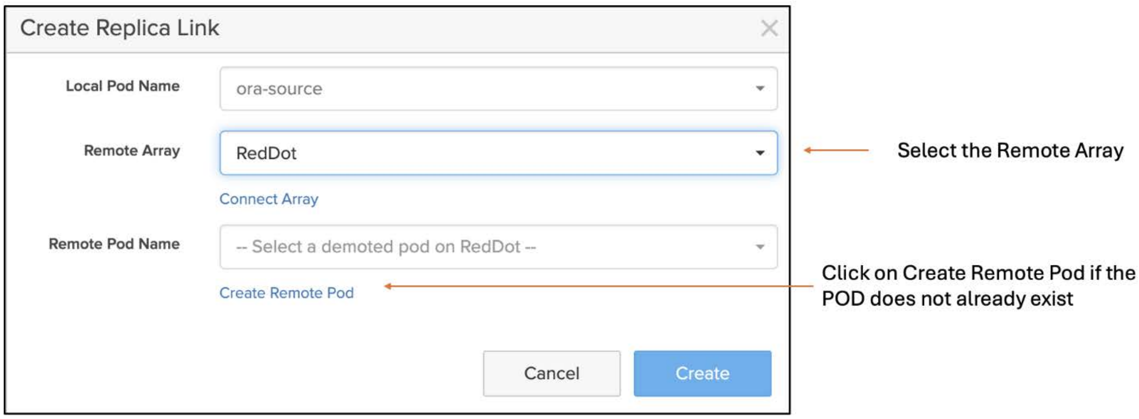

FIGURE 18 Creating a replica link.

Note: If a pod hasn't been created on the remote site's FlashArray, click **Create Remote Pod**, enter a pod name in the **Name** field, and then click **OK**.

10. Select **Create**.

Once a replica link has been created from the primary site's pod to the secondary site's pod, the primary site's pod begins a baseline replication of the volumes to the secondary site's pod. During a baseline replication operation, the primary site's FlashArray replicates a full copy of the volumes to the secondary site, which, depending on the size of the data, can take a significant amount of time. The status of the baselining progress can be monitored in the **Status** field in the **Pod Replica Links** group. When the baseline replication completes, the **Status** field changes from "baselining" to "replicating."

FIGURE 19 Monitoring pod replica link statuses on the primary site's FlashArray.

In addition to monitoring the pod replica link status at the primary site, the replica link status can also be monitored in the secondary site's Pure Storage user interface in the **Pod Replica Links** group.

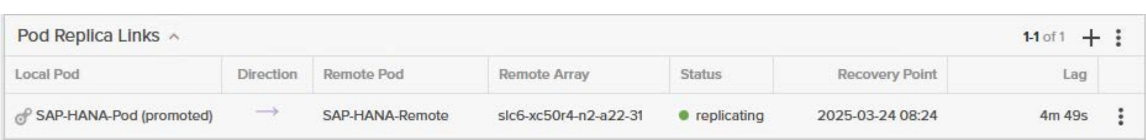

FIGURE 20 Monitoring the pod replica link status on the secondary site's FlashArray.

ActiveDR can now be used for the desired pods.

11. In the **Storage** section on the production/primary array, click the **Pods** tab, and then observe the state of the **Pod Replica Links** in a specific pod's details, which will show as **promoted**.

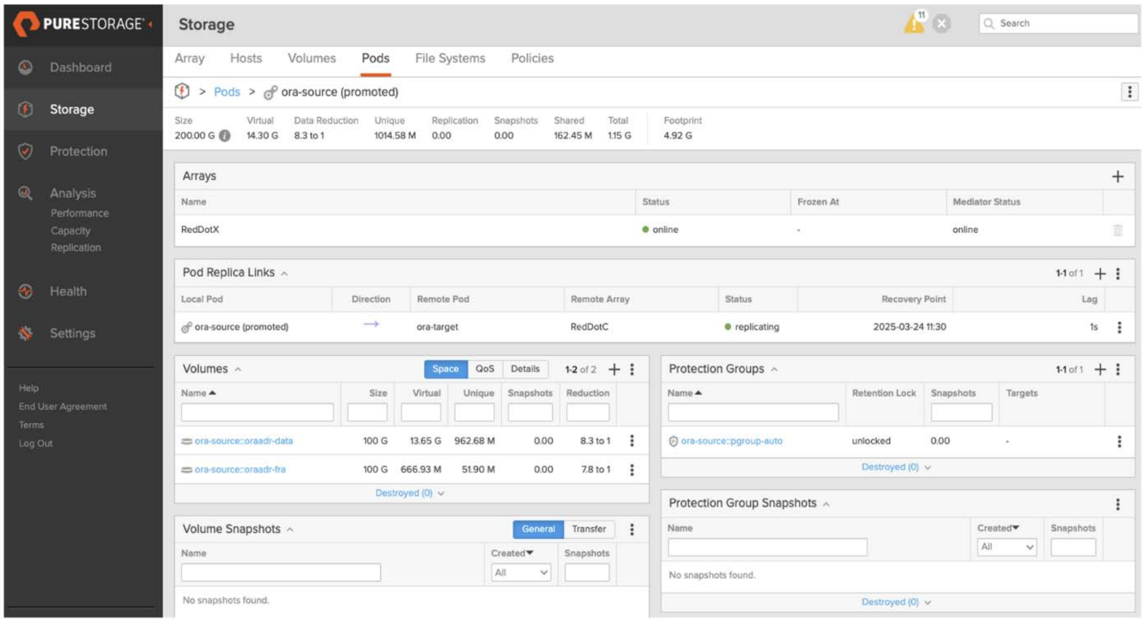

FIGURE 21 Checking the pod status on the primary array.

12. In the **Storage** section on the target/disaster recovery array, click the **Pods** tab, and then observe the state of the **Pod Replica Links**, which will show as **demoted**.

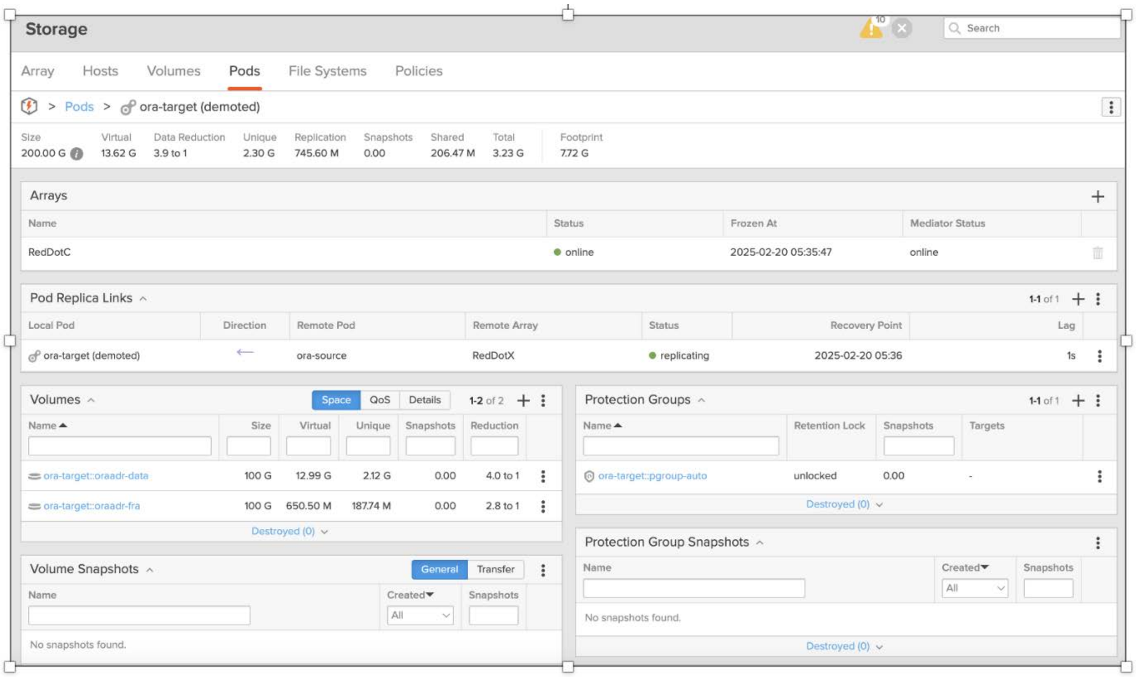

FIGURE 22 Checking the pod status on the target array.

Non-disruptive Failover Testing

Non-disruptive failover with ActiveDR allows for any one of the disaster recovery/target sites to bring a pod online without interruption to the production/primary pod. This is non-disruptive to the production instance database, and it allows for non-disruptive failover testing and the target database to be used for testing and development.

When using this process, the primary site will still queue changes, which will be applied to the target site once the process is undone.

To perform a non-disruptive failover, the state of the environment must be as follows:

- There is a functioning database residing on one or more volumes in a promoted ActiveDR pod.
- There is a demoted remote pod on a separate array in the same active pod replica link.
- The volumes in the demoted pod are connected to, and have been discovered by, a host with a running Oracle database.
- The demoted volumes are offline.

A non-disruptive failover consists of the following steps:

1. Promote the ActiveDR pod on the target array.
2. Set the disks online and, if required, provide a drive letter or mount to a folder.
3. Attach the database to the Oracle Database.

A failback after a non-disruptive failover only requires that the pod be demoted on the target system. Replication will automatically resume from the source pod to the target.

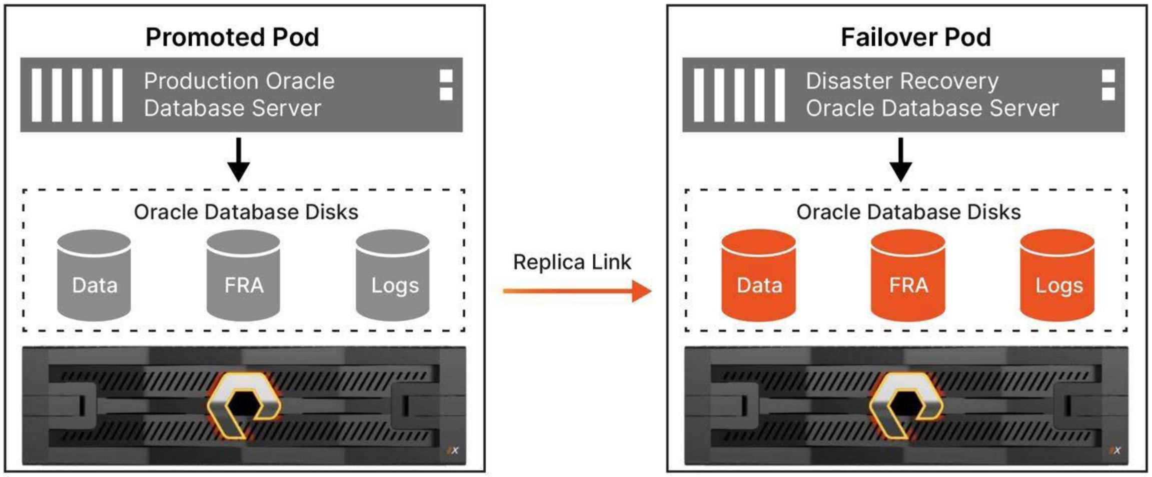

FIGURE 23 Configuration for a test failover with ActiveDR and Oracle Database.

Verification before a Test Failover

1. To complete the non-disruptive failover and failback between two arrays and different Oracle Database host systems, we must first create tables on the source and confirm they have been replicated.

```
SQL> create table adr101  as select * from dba_objects

Table created.

SQL> create table adr102 as select * from dba_objects;

Table created.

SQL>

SQL>  create table adr103  as select * from dba_objects;

Table created.

SQL>

SQL>

SQL> select count(*) from adr101;

  COUNT(*)

----------

     77010

SQL>

SQL>  select count(*) from adr102;

  COUNT(*)

----------

     77011

SQL>

SQL>  select count(*) from adr103;

  COUNT(*)

----------

     77012
```

2. Verify that the **ActiveDR replication link** is in a **healthy** state.

a. Use **purevol list --snap or purevol list --replica** to confirm replication health.

3. Confirm the target Oracle Database server volumes are not mounted and the Oracle Database is not running. For example, the **/u01** and **/u02** file systems are not mounted.

```
[root@oraadr2 ~]#
[root@oraadr2 ~]# df -h
Filesystem           Size  Used Avail Use% Mounted on
devtmpfs             7.7G     0  7.7G   0% /dev
tmpfs                7.7G     0  7.7G   0% /dev/shm
tmpfs                7.7G   18M  7.7G   1% /run
tmpfs                7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/mapper/ol-root   61G  7.9G   53G  13% /
/dev/sda2           1014M  457M  558M  46% /boot
/dev/mapper/ol-home   30G  3.2G   27G  11% /home
/dev/sda1            599M  6.0M  593M   1% /boot/efi
tmpfs                1.6G   12K  1.6G   1% /run/user/42
tmpfs                1.6G  4.0K  1.6G   1% /run/user/54321
[root@oraadr2 ~]#
[root@oraadr2 ~]#
```

```
[root@oraadr2 ~]# lsblk -o name,serial
NAME        SERIAL
sda
sda
 ├─sda1
 ├─sda2
 └─sda3
   ├─ol-root
   ├─ol-swap
   └─ol-home
sdb       21F018170267441100064061
 └─sdb1
sdc       21F018170267441100064062
 └─sdc1
sr0       00000000000000000001
[root@oraadr2 ~]#
```

4. Verify the disk serial numbers for the target server are correct. In the example shown in Figure 24, we can see the serial numbers on the target server volumes match the serial numbers in the target pod, ... **4061** and ... **4062**.

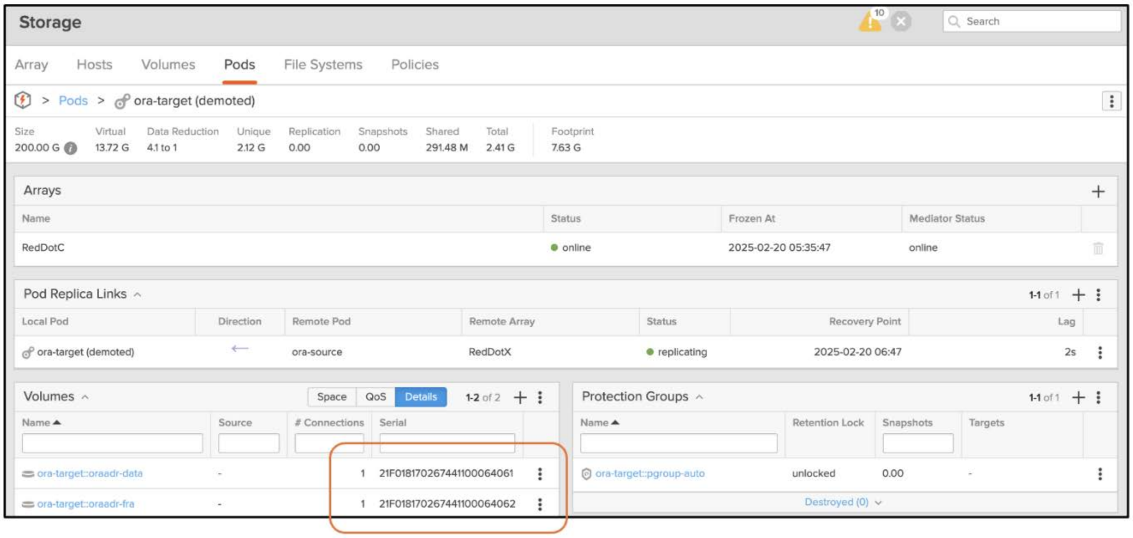

FIGURE 24 Confirming the serial numbers.

Now we can perform the test failover.

Execute a Test Failover

1. In the **Storage** section on the production/primary array, click the **Pods** tab, and then observe the state of the **Pod Replica Links** in a specific pod's details, which will show as **promoted**.

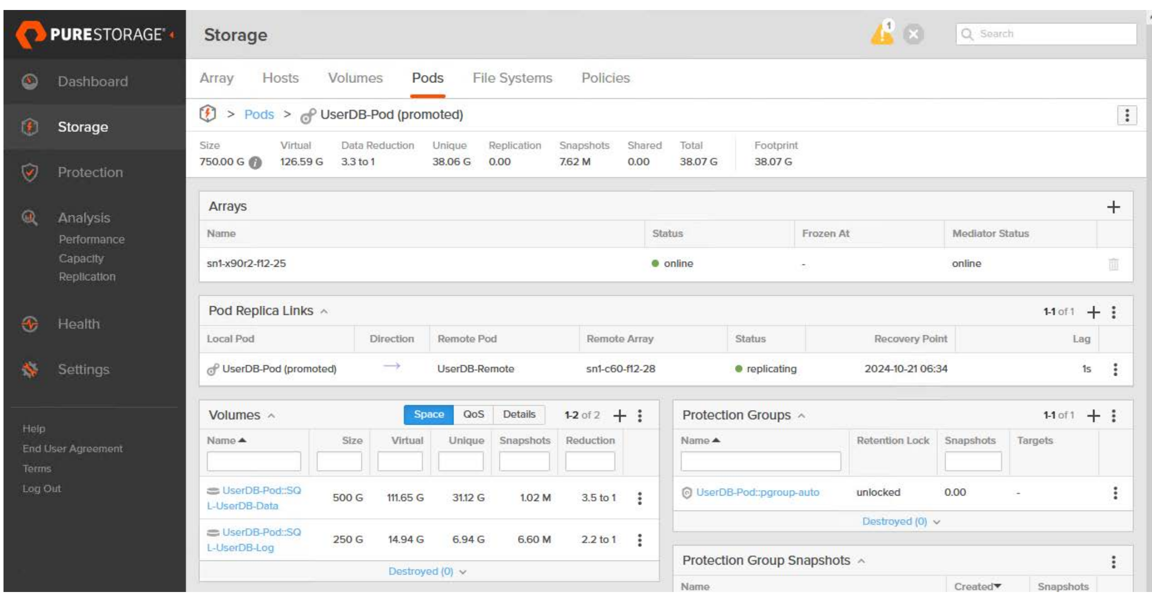

FIGURE 25 isplaying the state of the production/primary array pod.

2. Navigate to the specific pod in **Pods** under **Storage** on the target/disaster recovery array, click the ellipsis, and then select **Promote**.

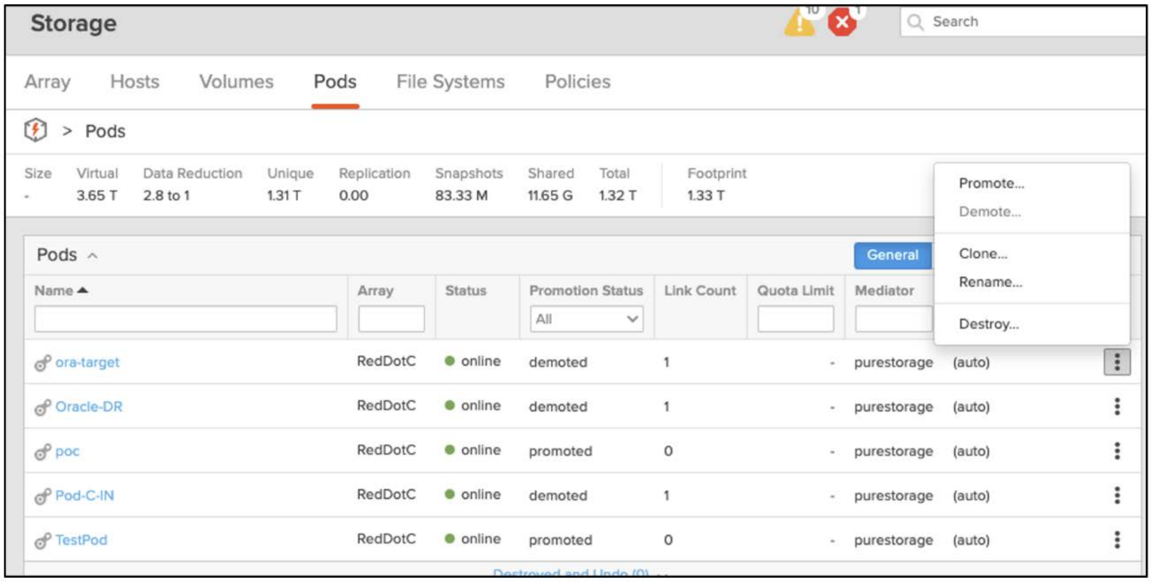

FIGURE 26 Promoting the pod.

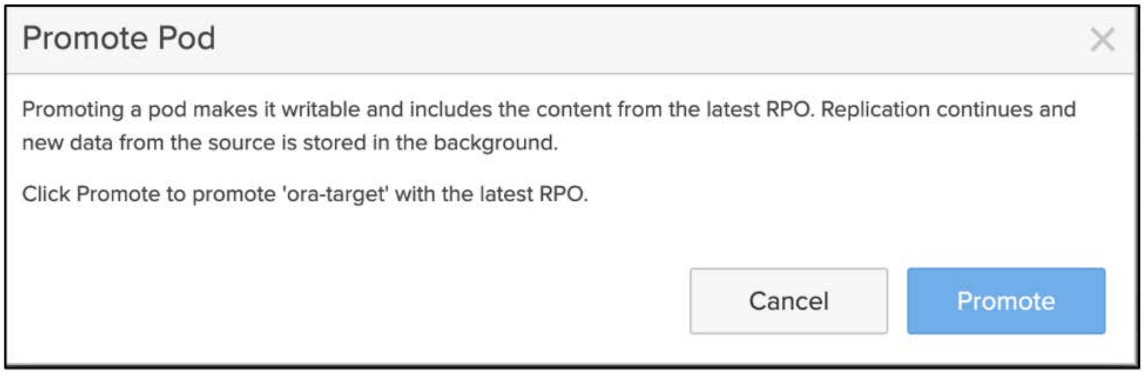

FIGURE 27 The Promote Pod dialogue.

When the pod has been promoted, it will have a promotion status of **promoted**.

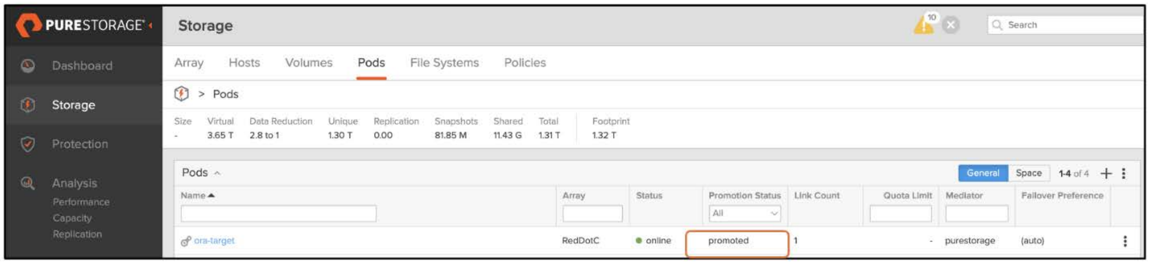

FIGURE 28 Displaying the state of the pod.

3. Mount the volumes on the target server.

```
[root@oraadr2 ~]#

[root@oraadr2 ~]# mount /dev/sdb1 /u01

[root@oraadr2 ~]#

[root@oraadr2 ~]#

[root@oraadr2 ~]# mount /dev/sdc1 /u02

[root@oraadr2 ~]#

[root@oraadr2 ~]#

[root@oraadr2 ~]# df -h

Filesystem           Size  Used Avail Use% Mounted on

devtmpfs             7.7G     0  7.7G   0% /dev

tmpfs                7.7G     0  7.7G   0% /dev/shm

tmpfs                7.7G   18M  7.7G   1% /run

tmpfs                7.7G     0  7.7G   0% /sys/fs/cgroup

/dev/mapper/ol-root   61G  8.1G   53G  14% /

/dev/sda2           1014M  457M  558M  46% /boot

/dev/mapper/ol-home   30G  3.2G   27G  11% /home

/dev/sda1            599M  6.0M  593M   1% /boot/efi

tmpfs                1.6G   12K  1.6G   1% /run/user/42

tmpfs                1.6G  4.0K  1.6G   1% /run/user/54321

/dev/sdb1            100G   14G   87G  14% /u01

/dev/sdc1            100G  1.4G   99G   2% /u02
```

4. Prepare Oracle Database by doing the following:

a. Switch user to **Oracle**.

b. Load and execute the configuration settings for the **Oracle Shell Profile.**

c. Use **SQL*PLUS** to connect to the database and run the startup command.

d. Confirm the correct host by querying the hostname from **v$instance.**

e. Confirm the tables **adr101**, **adr101**, and **adr103** have been replicated, as seen in the following example.

```
[root@oraadr2 ~]# su - oracle
[oracle@oraadr2 ~]$
[oracle@oraadr2 ~]$ . ./.bash_profile
[oracle@oraadr2 ~]$
[oracle@oraadr2 ~]$
[oracle@oraadr2 ~]$ sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Feb 20 10:10:42 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle.  All rights reserved.
Connected to an idle instance.
SQL>
SQL> startup
ORACLE instance started.
Total System Global Area 4949275712 bytes
Fixed Size                  9696320 bytes
Variable Size            1476395008 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7077888 bytes
Database mounted.
Database opened.
SQL>
SQL> select host_name from v$instance;
HOST_NAME
----------------------------------------------------------------
Oraadr2.localdomain
SQL> select count(*) from adr101;
  COUNT(*)
----------
     77010
SQL>  select count(*) from adr102;
  COUNT(*)
----------
     77011
SQL>  select count(*) from adr103;
  COUNT(*)
----------
     77012
```

5. Because the source production pod is still replicating the data from the target pod onto the replication space, create three more tables on the source pod.

```
SQL>  select host_name from v$instance;

HOST_NAME

----------------------------------------------------------------

oraadr1.localdomain

SQL>  create table adr104  as select * from dba_objects;

Table created.

SQL> create table adr105  as select * from dba_objects;

Table created.

SQL> create table adr106 as select * from dba_objects;

Table created.
```

6. Verify the new tables do not exist on the target pod, which should still be promoted and running.

```
SQL>  select count(*) from adr104;

 select count(*) from adr104

                      *

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> select count(*) from adr105;

select count(*) from adr105

                     *

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> select count(*) from adr106;

select count(*) from adr106

                     *

ERROR at line 1:

ORA-00942: table or view does not exist
```

The replication site is now verified to be functional and available for use.

Failback from a Planned Non-disruptive Failover

1. Shut down Oracle Database and unmount the file systems.

```
SQL> shut immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
[root@oraadr2 ~]# umount /u01
[root@oraadr2 ~]#
[root@oraadr2 ~]# umount /u02
[root@oraadr2 ~]#
[root@oraadr2 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
devtmpfs              7.7G     0  7.7G   0% /dev
tmpfs                 7.7G     0  7.7G   0% /dev/shm
tmpfs                 7.7G  9.2M  7.7G   1% /run
tmpfs                 7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/mapper/ol-root    61G  8.1G   53G  14% /
/dev/sda2            1014M  457M  558M  46% /boot
/dev/mapper/ol-home    30G  3.2G   27G  11% /home
/dev/sda1             599M  6.0M  593M   1% /boot/efi
tmpfs                 1.6G   12K  1.6G   1% /run/user/42
tmpfs                 1.6G  4.0K  1.6G   1% /run/user/54321
```

2. Navigate to the ActiveDR pod on the target/disaster recovery array, and then select **Demote Local Pod.** This will apply all changes from the production/primary instance and then set the pod as read-only.

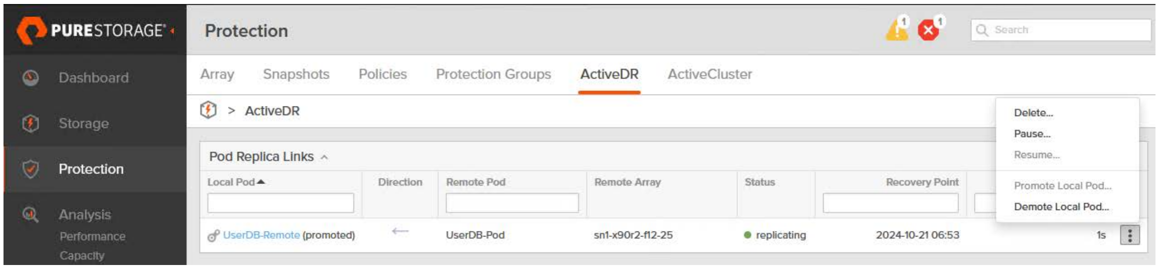

FIGURE 29 Demoting the local pod.

3. When selecting **Demote Local Pod**, a prompt to confirm demotion appears. Select **Demote** when ready.

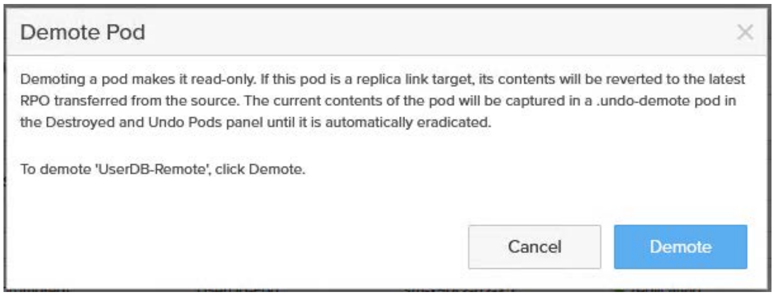

FIGURE 30 Confirming the pod demotion.

4. Once the pod has been demoted, it will show as **demoted** in the Pod Replica Links pane.

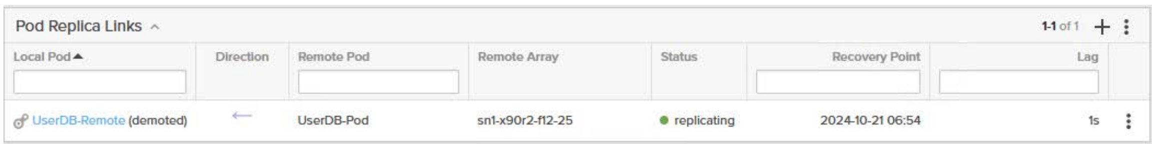

FIGURE 31 Displaying the pod status.

Next, we need to attach the database to the remote site's Oracle Database instance again to confirm the new tables we created (**ADR104**, **ADR105**, and **ADR106**) are there**.**

5. When ready to perform the failover, navigate to the specific pod in **Pods**, under **Storage**, on the target/disaster recovery array, click the ellipsis, and then select **Promote**.

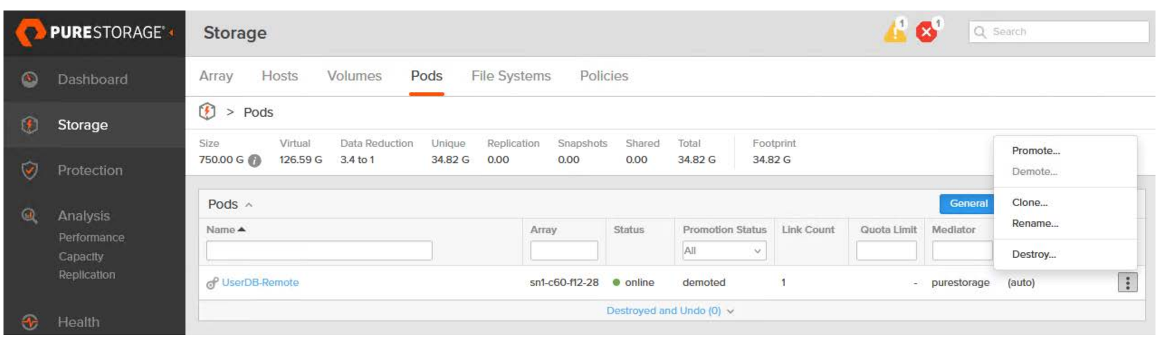

FIGURE 32 Promoting the target/disaster recovery array pod.

6. When the pod has been promoted, it will have a promotion status of **promoted**.

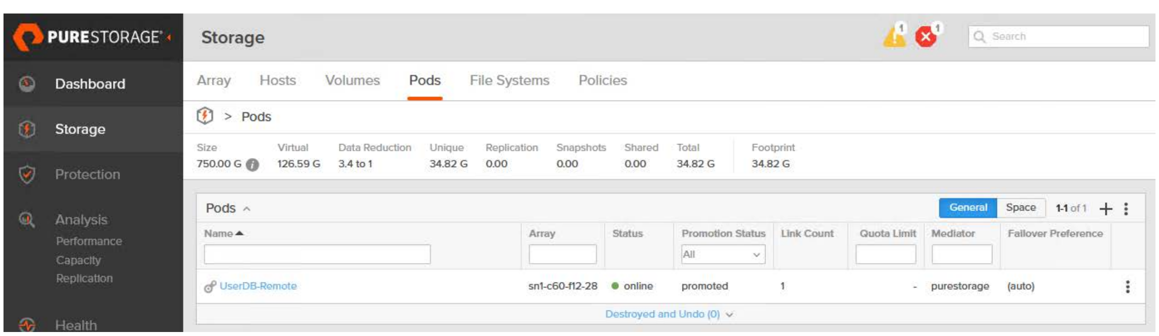

FIGURE 33 The promoted pod on the target/disaster recovery array.

7. Mount the file systems **/u01** and **/u02.**
8. Switch user to **Oracle**, and then start the database.
9. Confirm the new tables exist.

```
SQL>

SQL> select count(*) from adr104;

  COUNT(*)

----------

     77013

SQL>  select count(*) from adr105;

  COUNT(*)

----------

     77014

SQL>  select count(*) from adr106;

  COUNT(*)

----------

     77015
```

Once you've confirmed the new tables are present, operations can resume.

Conclusion

ActiveDR provides database and storage teams with a powerful, low-overhead disaster recovery solution for Oracle Database environments. With continuous, near-synchronous replication and support for non-disruptive failovers, failbacks, and disaster recovery testing, organizations can protect mission-critical data while maintaining uptime and performance.

By enabling seamless recovery without complex integration or added operational burden, ActiveDR helps businesses build resilient Oracle environments that are ready for the unexpected—whether caused by hardware failure, cyber threats, or site outages.

For more information, visit www.purestorage.com/solutions/databases-applications/oracle.html or try the functionality in a test drive at www.purestorage.com/products/unified-block-file-storage/flasharray-x/test-drive-dba.html.

CM422-01-en 04/25