Disaster recovery workflow

NetApp Solutions

NetApp
October 20, 2023

Table of Contents

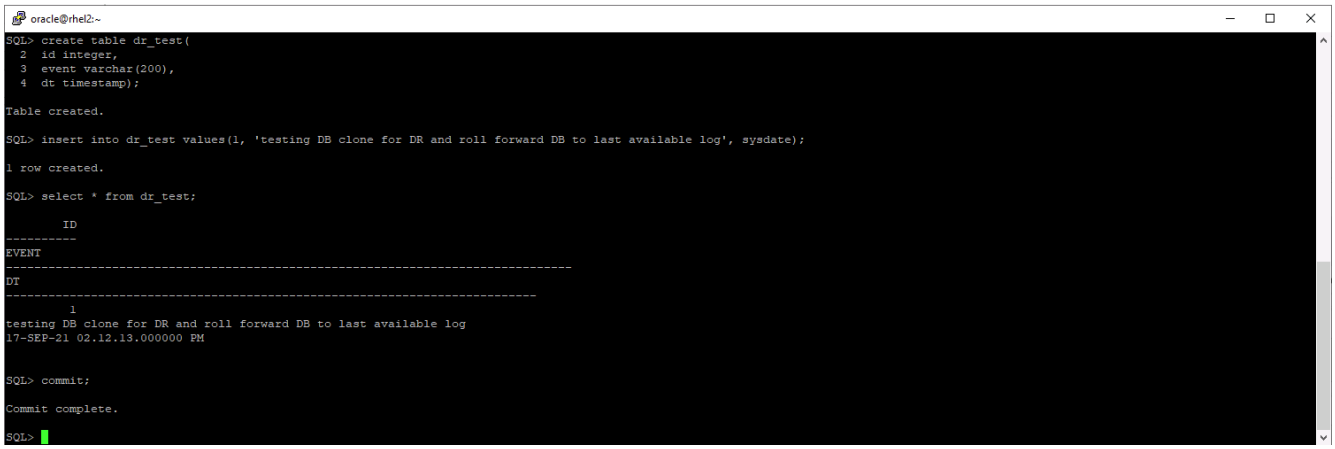
Disaster recovery workflow

Previous: Workflow for dev/test bursting to cloud.

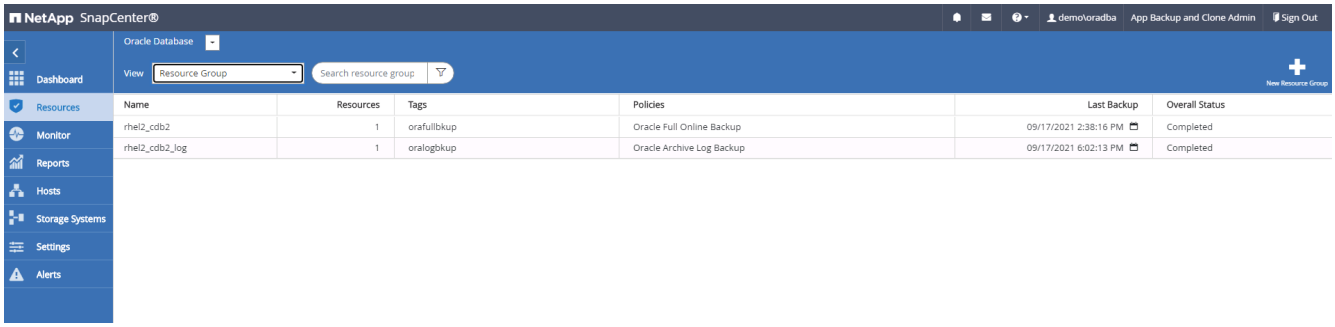
Enterprises have embraced the public cloud as a viable resource and destination for disaster recovery. SnapCenter makes this process as seamless as possible. This disaster recovery workflow is very similar to the clone workflow, but database recovery runs through the last available log that was replicated to cloud to recover all the business transactions possible. However, there are additional pre-configuration and post-configuration steps specific to disaster recovery.

Clone an on-premises Oracle production DB to cloud for DR

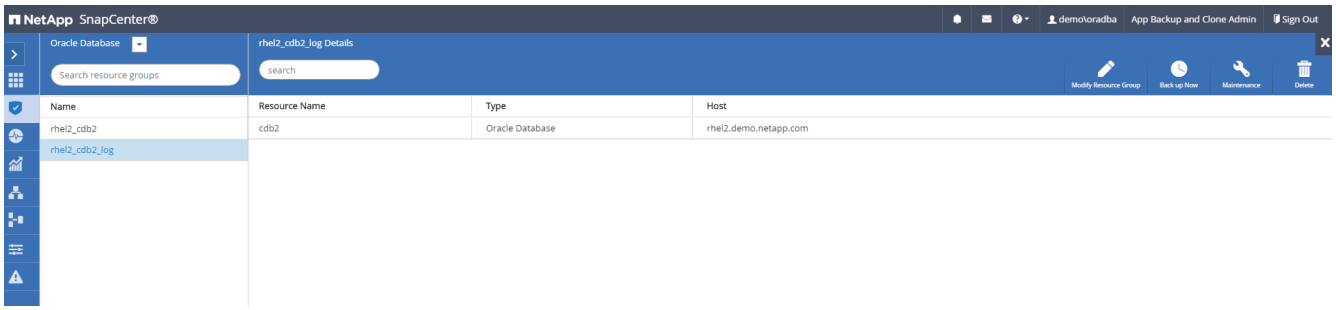
1. To validate that the clone recovery runs through last available log, we created a small test table and inserted a row. The test data would be recovered after a full recovery to last available log.

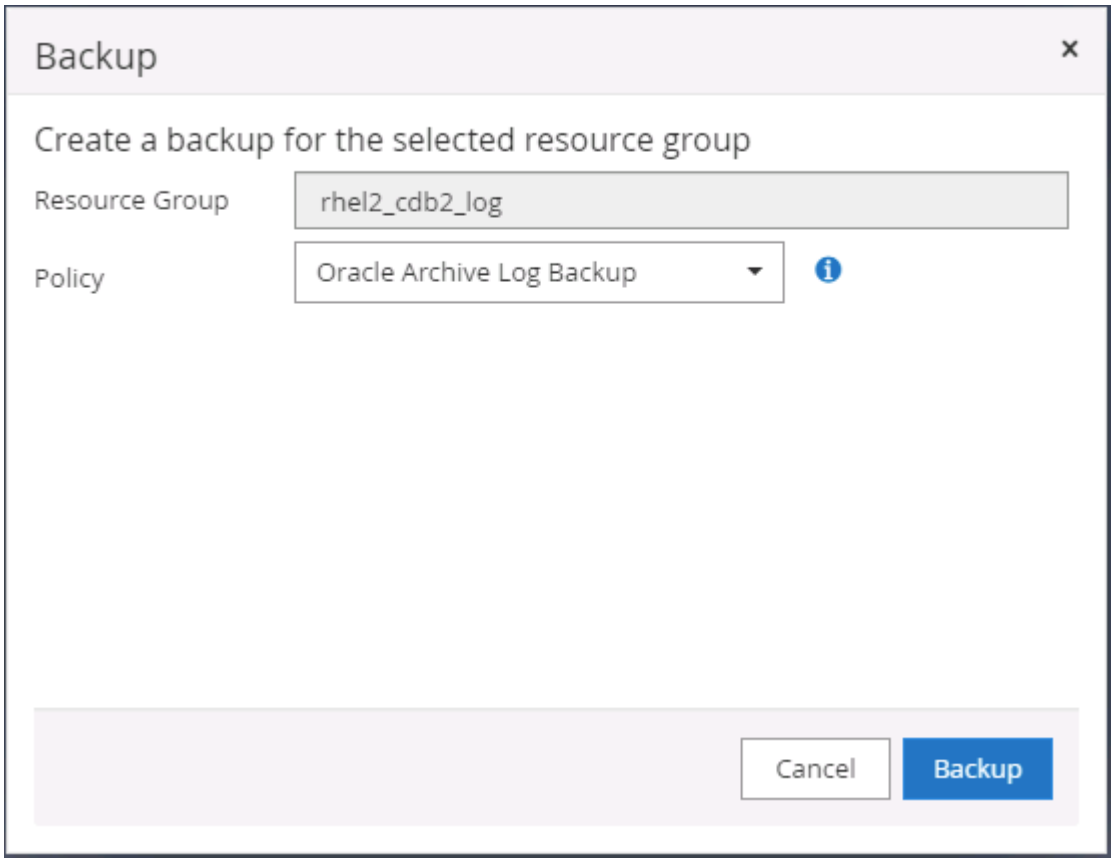


2. Log into SnapCenter as a database management user ID for Oracle. Navigate to the Resources tab, which shows the Oracle databases being protected by SnapCenter.



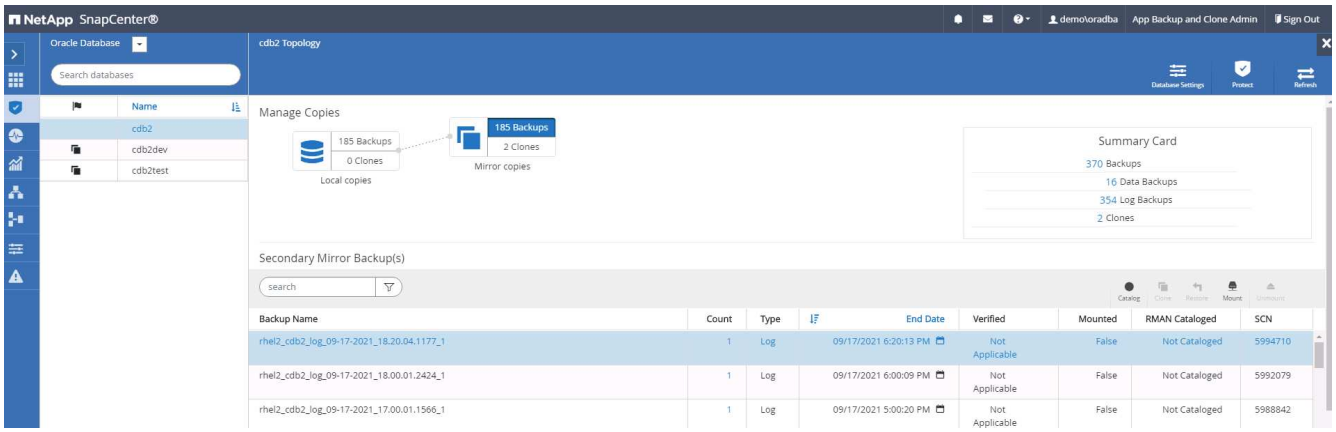
3. Select the Oracle log resource group and click Backup Now to manually run an Oracle log backup to flush the latest transaction to the destination in the cloud. In a real DR scenario, the last transaction recoverable depends on the database log volume replication frequency to the cloud, which in turn depends on the RTO or RPO policy of the company.

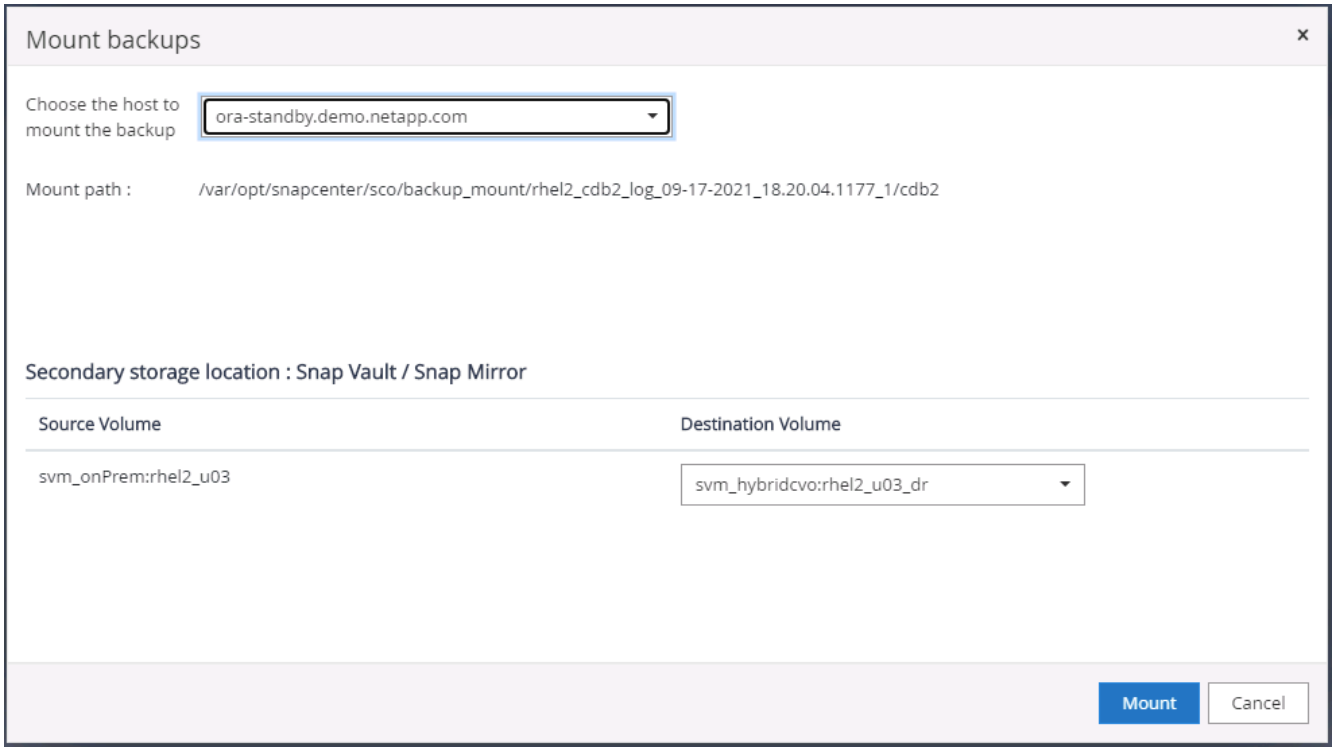




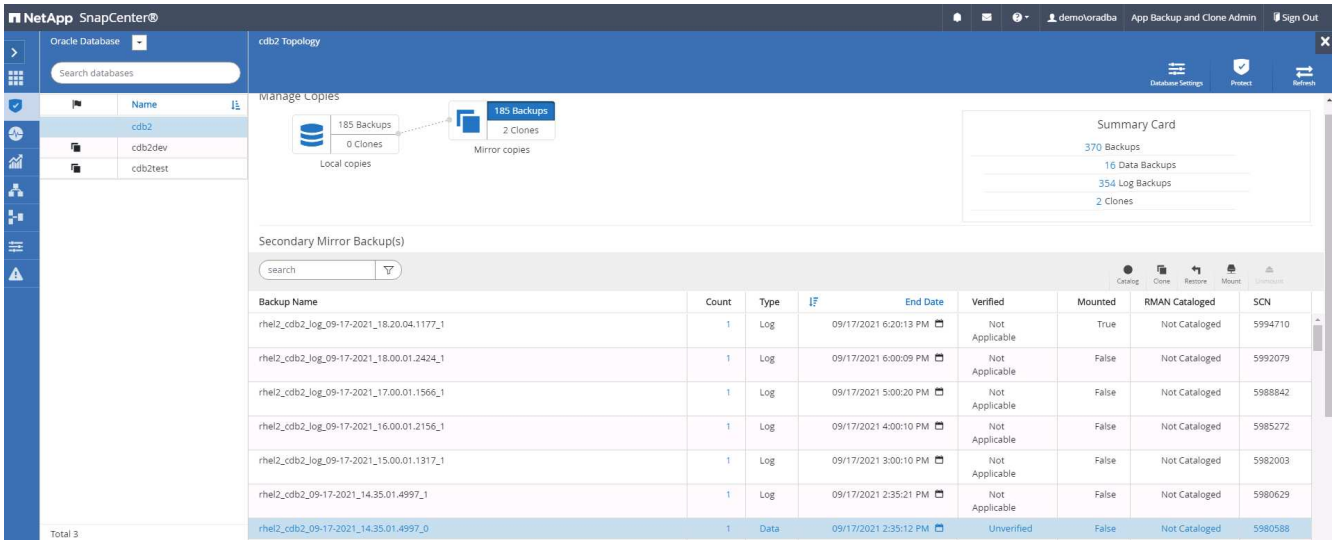
Asynchronous SnapMirror loses data that has not made it to the cloud destination in the database log backup interval in a disaster recovery scenario. To minimize data loss, more frequent log backup can be scheduled. However there is a limit to the log backup frequency that is technically achievable.

4. Select the last log backup on the Secondary Mirror Backup(s), and mount the log backup.

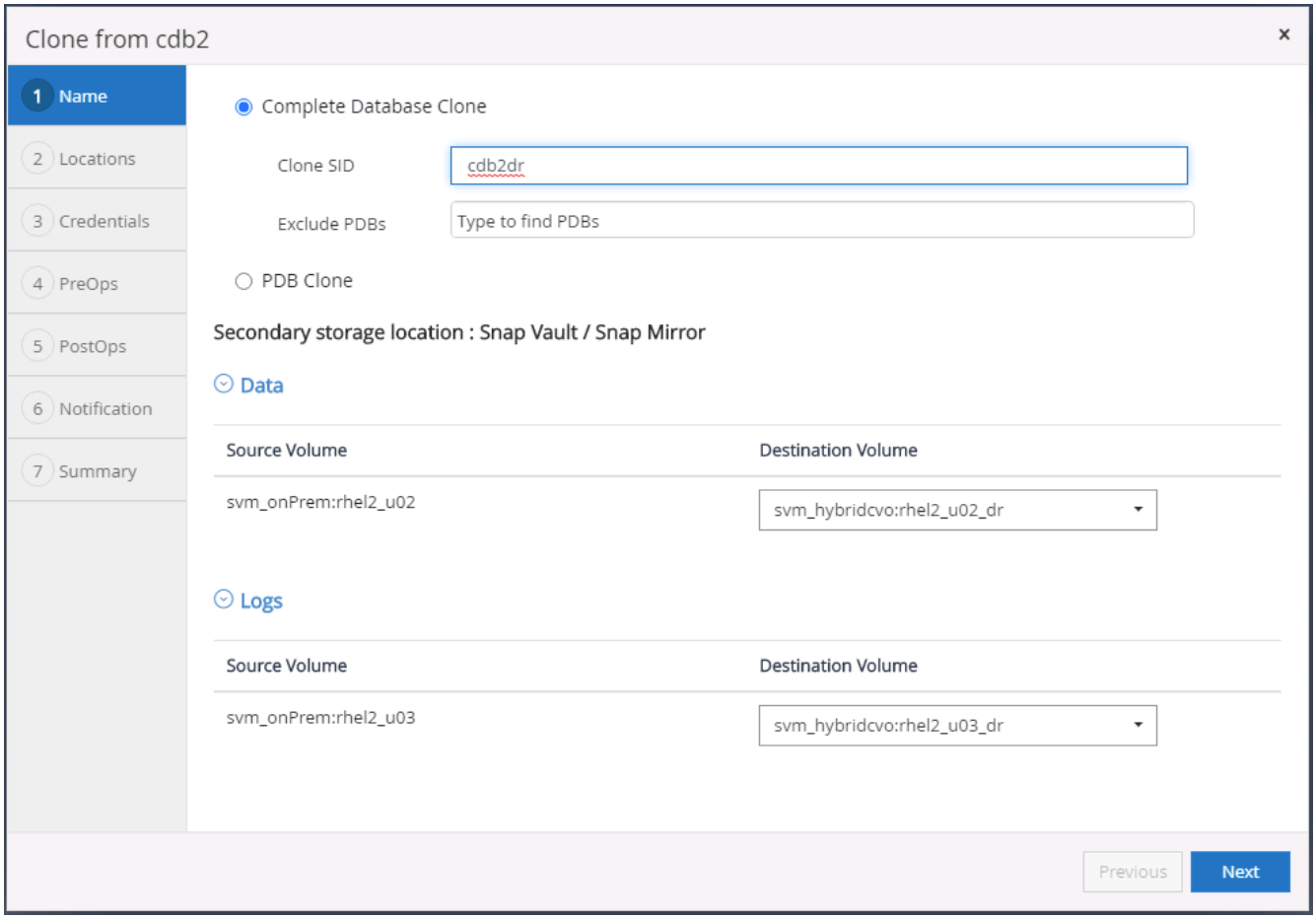




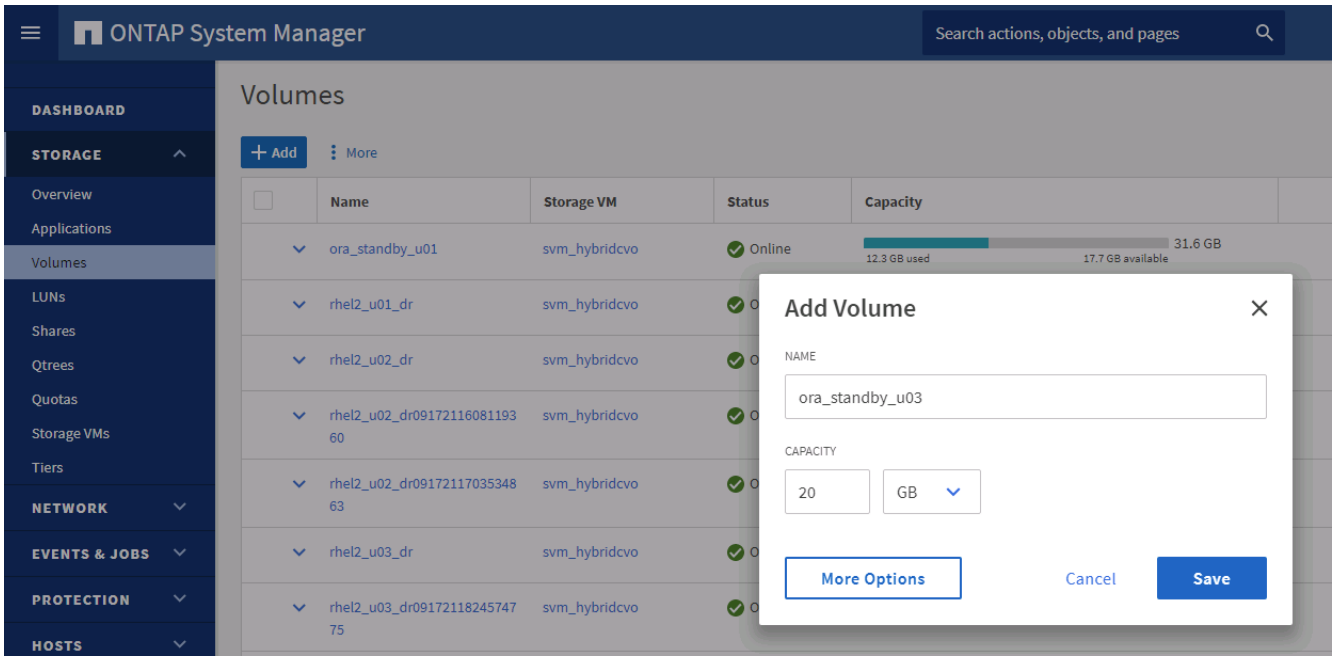
5. Select the last full database backup and click Clone to initiate the clone workflow.

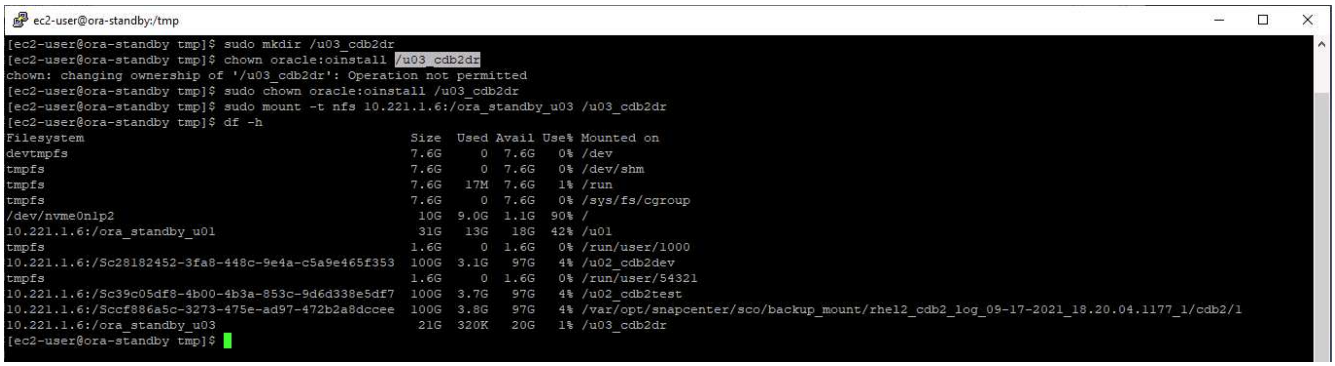


6. Select a unique clone DB ID on the host.



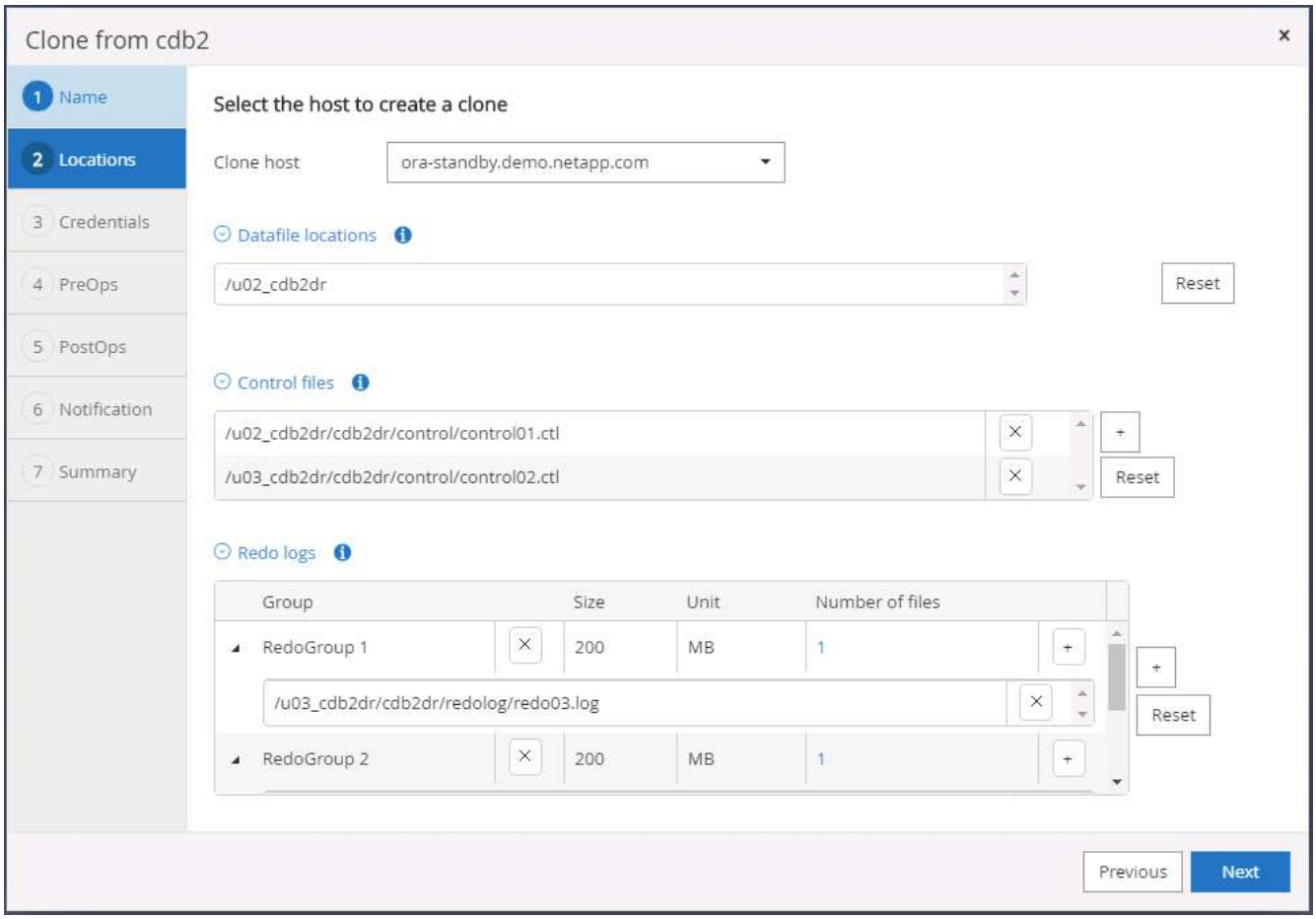
7. Provision a log volume and mount it to the target DR server for the Oracle flash recovery area and online logs.



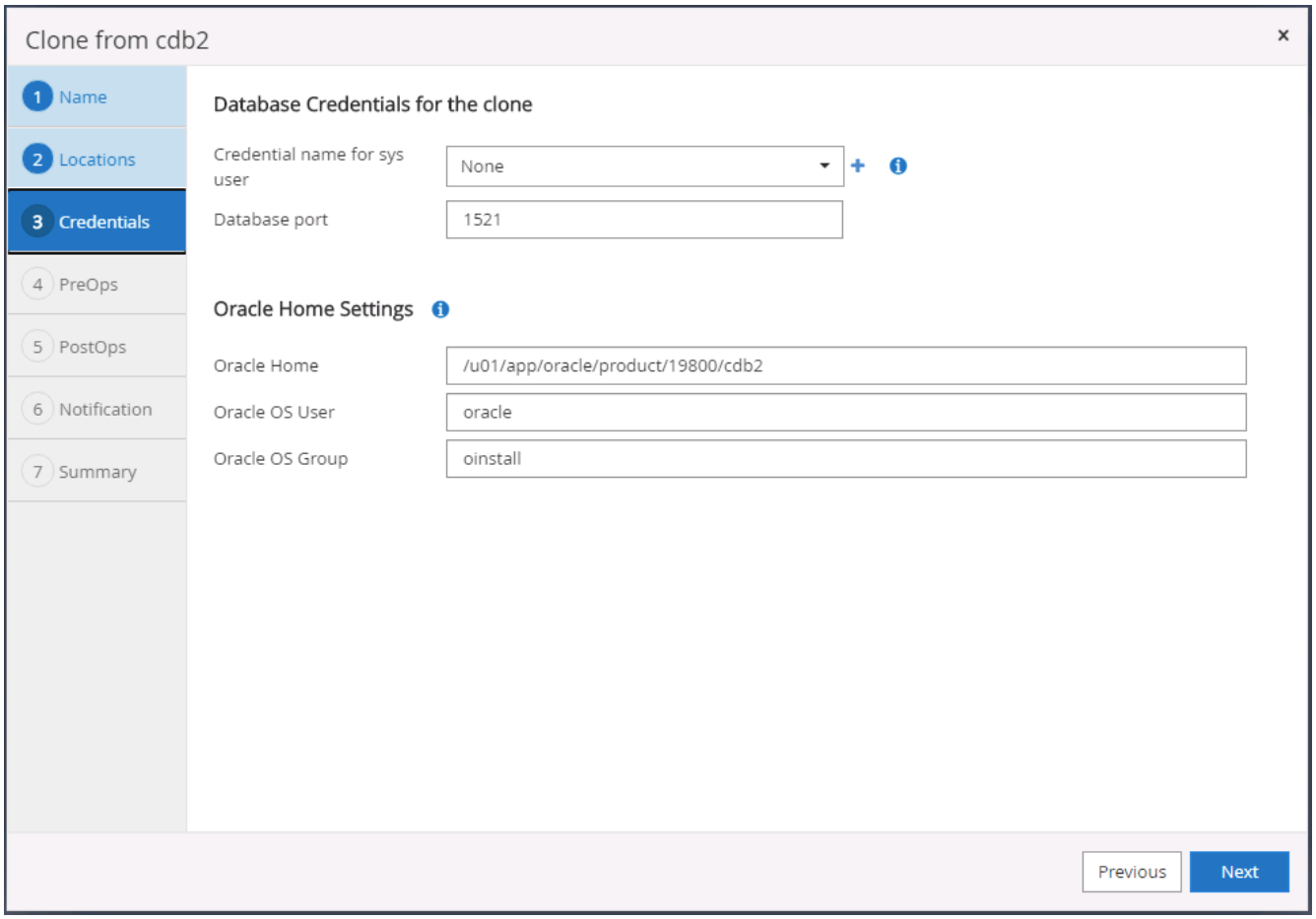


The Oracle clone procedure does not create a log volume, which needs to be provisioned on the DR server before cloning.

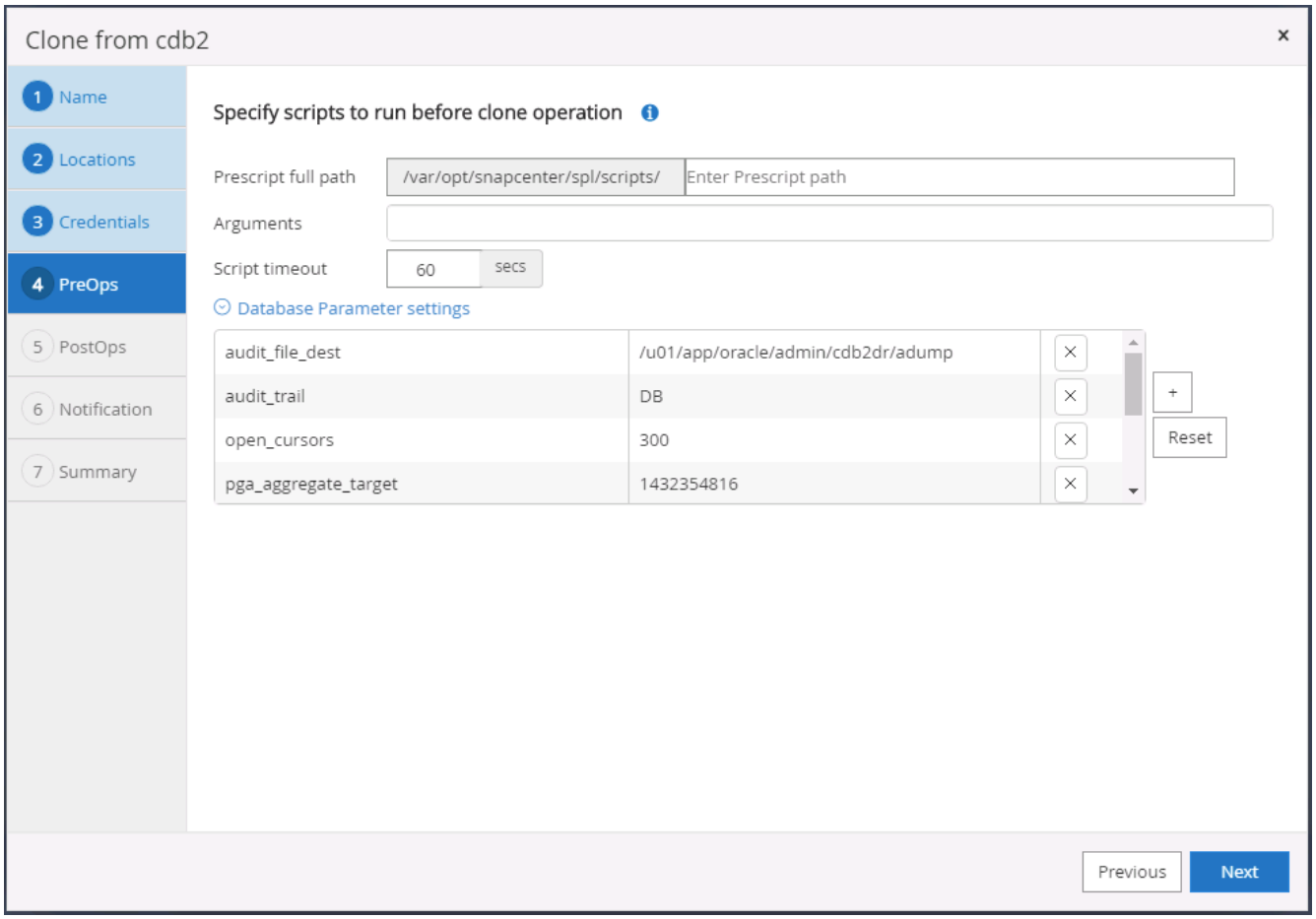
8. Select the target clone host and location to place the data files, control files, and redo logs.



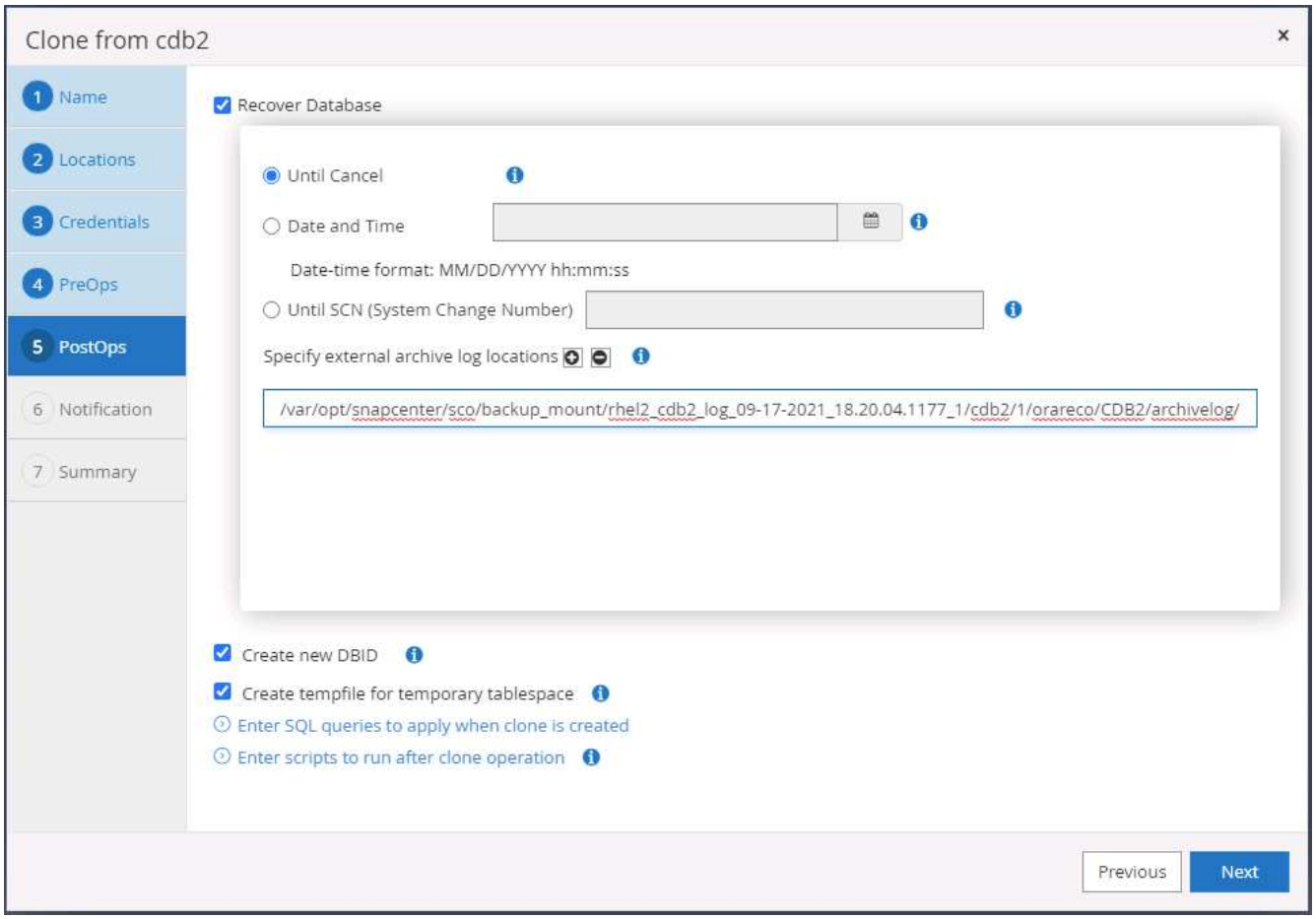
9. Select the credentials for the clone. Fill in the details of the Oracle home configuration on the target server.



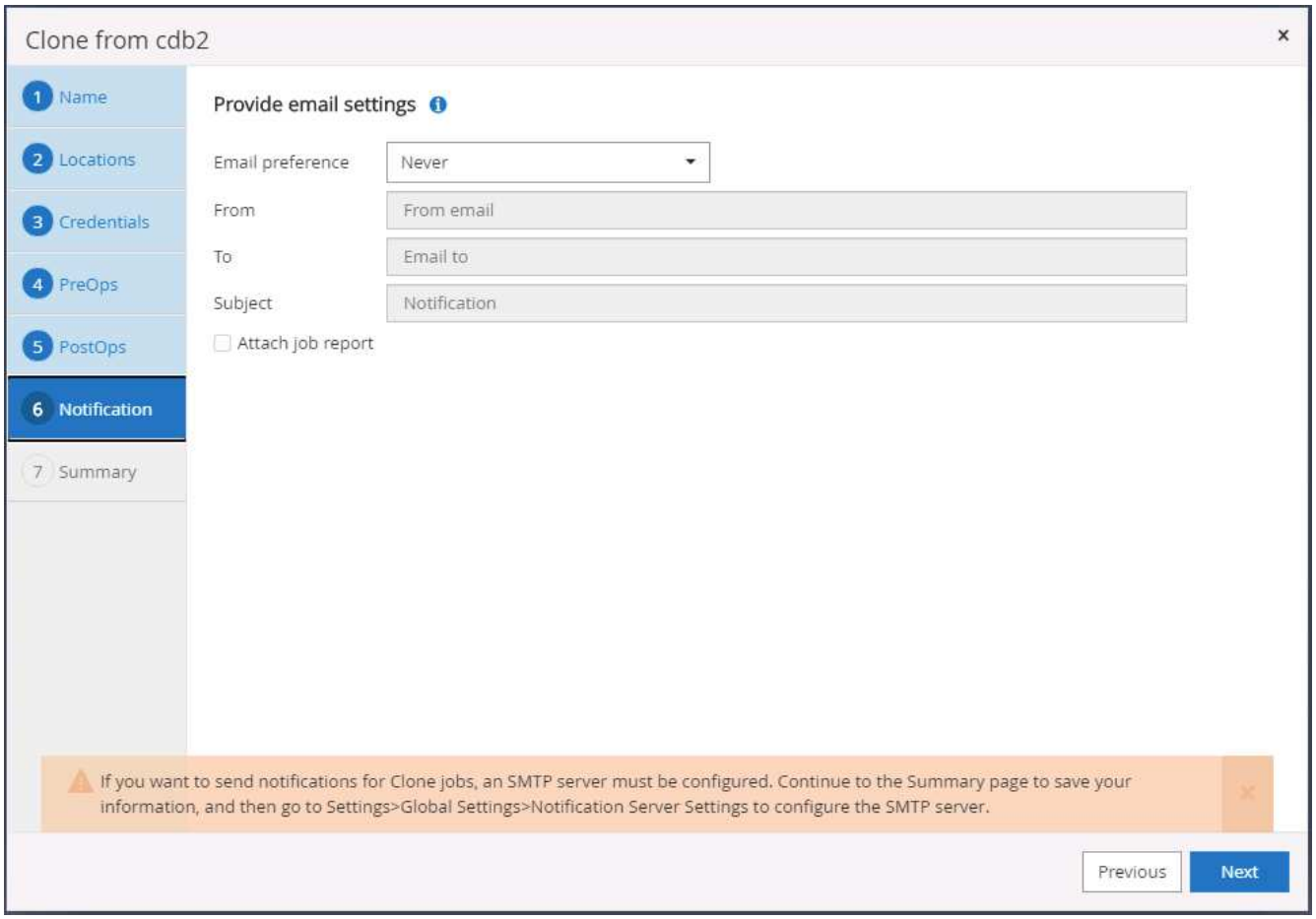
10. Specify the scripts to run before cloning. Database parameters can be adjusted if needed.



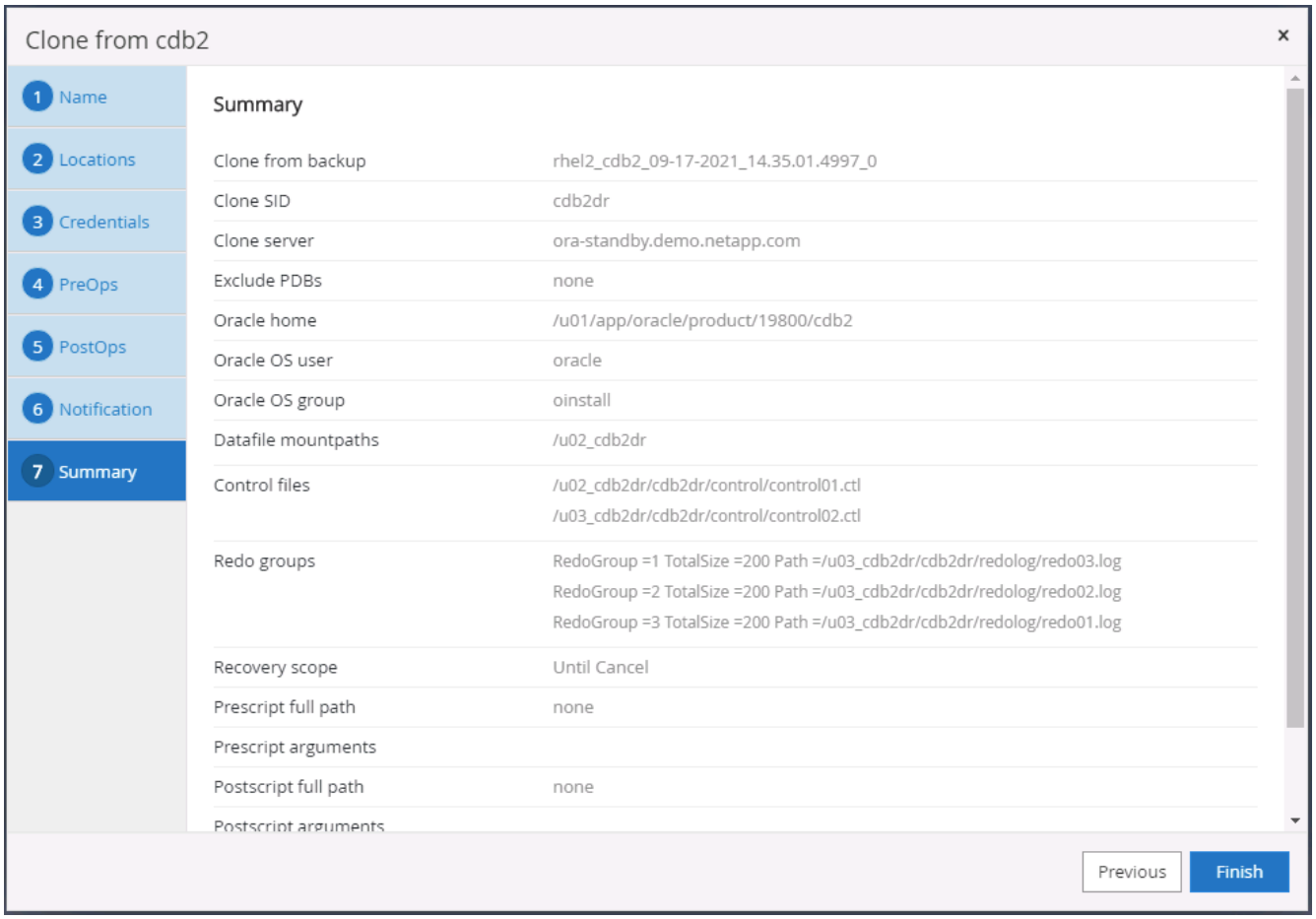
11. Select Until Cancel as the recovery option so that the recovery runs through all available archive logs to recoup the last transaction replicated to the secondary cloud location.



12. Configure the SMTP server for email notification if needed.



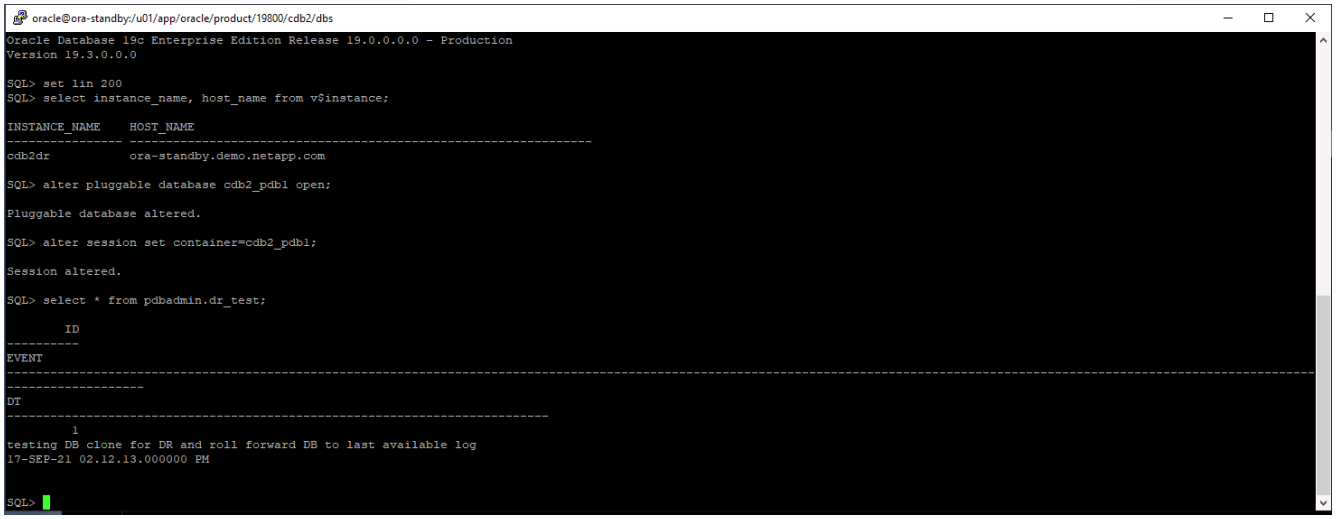
13. DR clone summary.



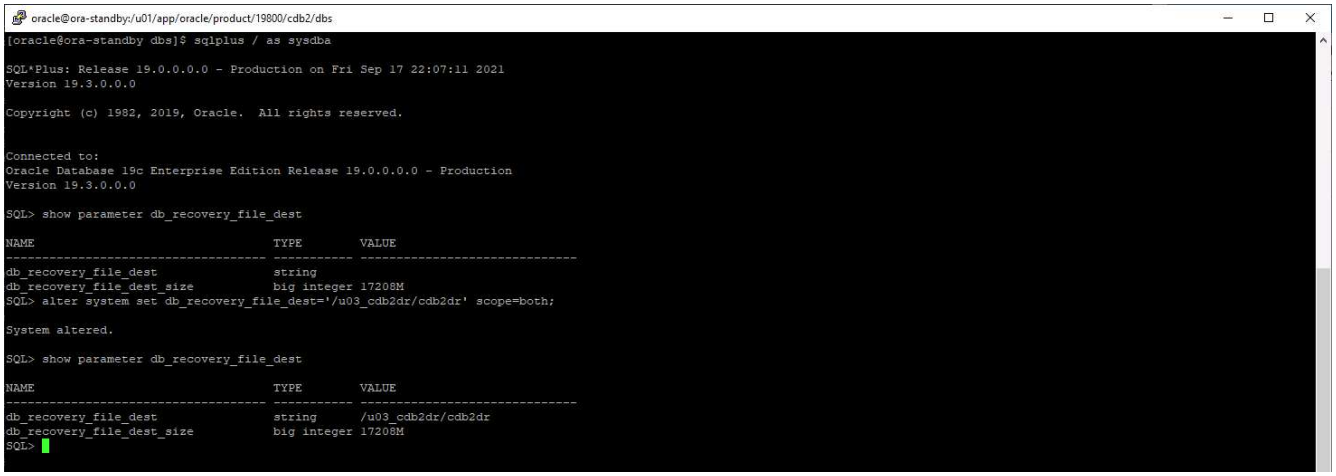
14. Cloned DBs are registered with SnapCenter immediately after clone completion and are then available for backup protection.

Post DR clone validation and configuration for Oracle

1. Validate the last test transaction that has been flushed, replicated, and recovered at the DR location in the cloud.



2. Configure the flash recovery area.

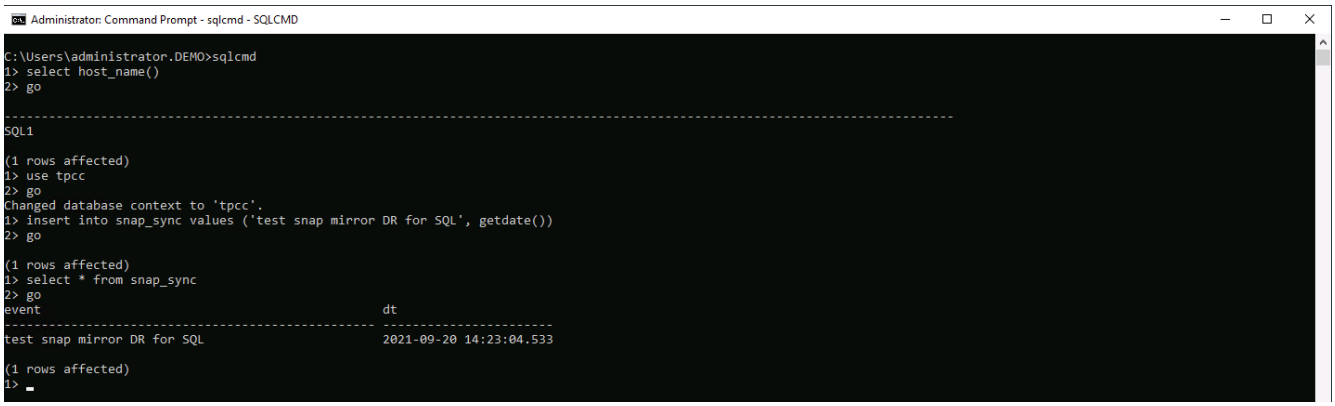


3. Configure the Oracle listener for user access.
4. Split the cloned volume off of the replicated source volume.
5. Reverse replication from the cloud to on-premises and rebuild the failed on-premises database server.

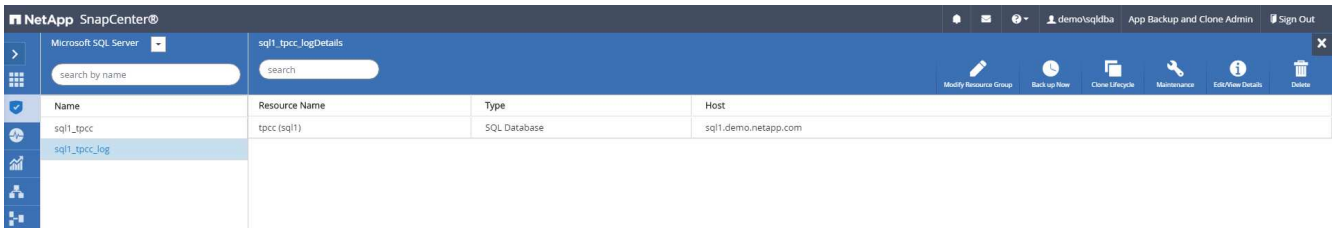
Clone split may incur temporary storage space utilization that is much higher than normal operation. However, after the on-premises DB server is rebuilt, extra space can be released.

Clone an on-premises SQL production DB to cloud for DR

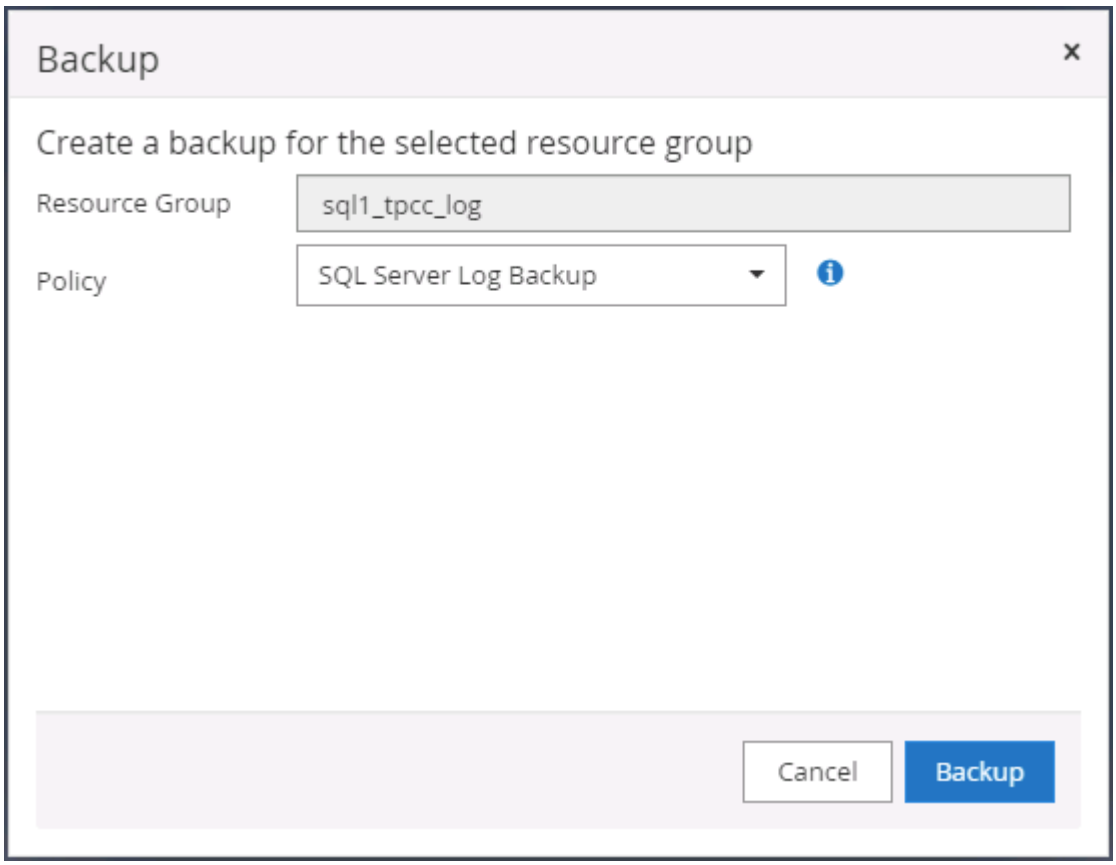
1. Similarly, to validate that the SQL clone recovery ran through last available log, we created a small test table and inserted a row. The test data would be recovered after a full recovery to the last available log.



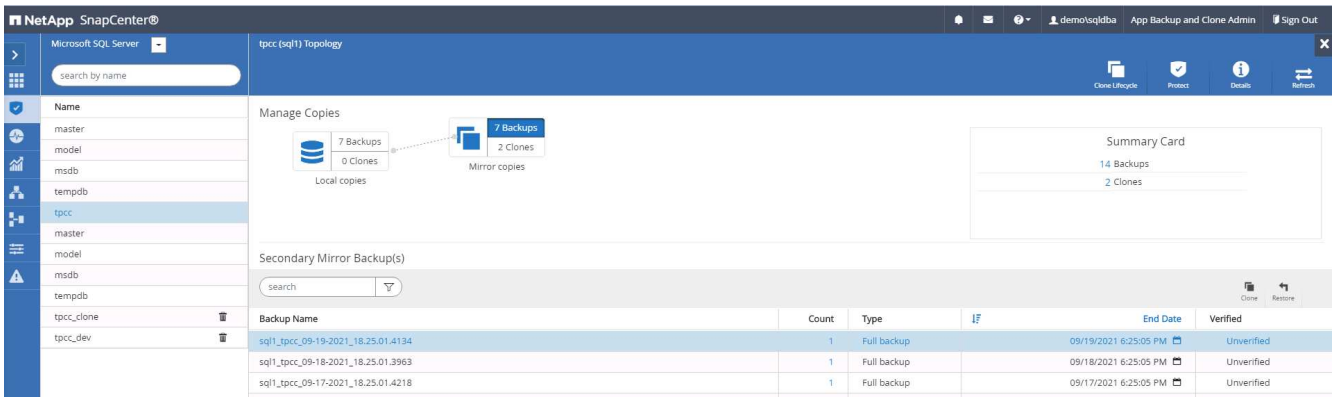
2. Log into SnapCenter with a database management user ID for SQL Server. Navigate to the Resources tab, which shows the SQL Server protection resources group.



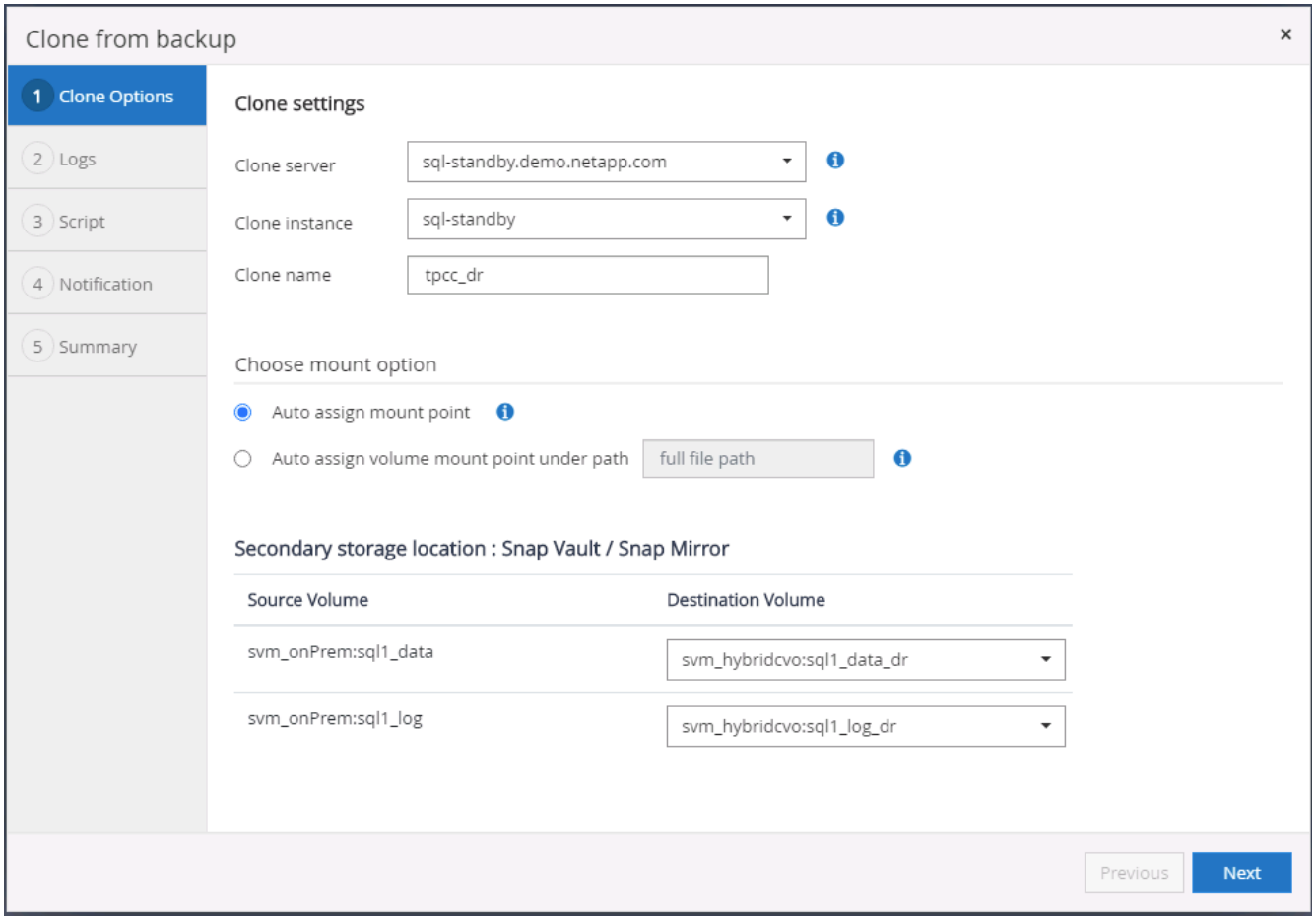
3. Manually run a log backup to flush the last transaction to be replicated to secondary storage in the public cloud.



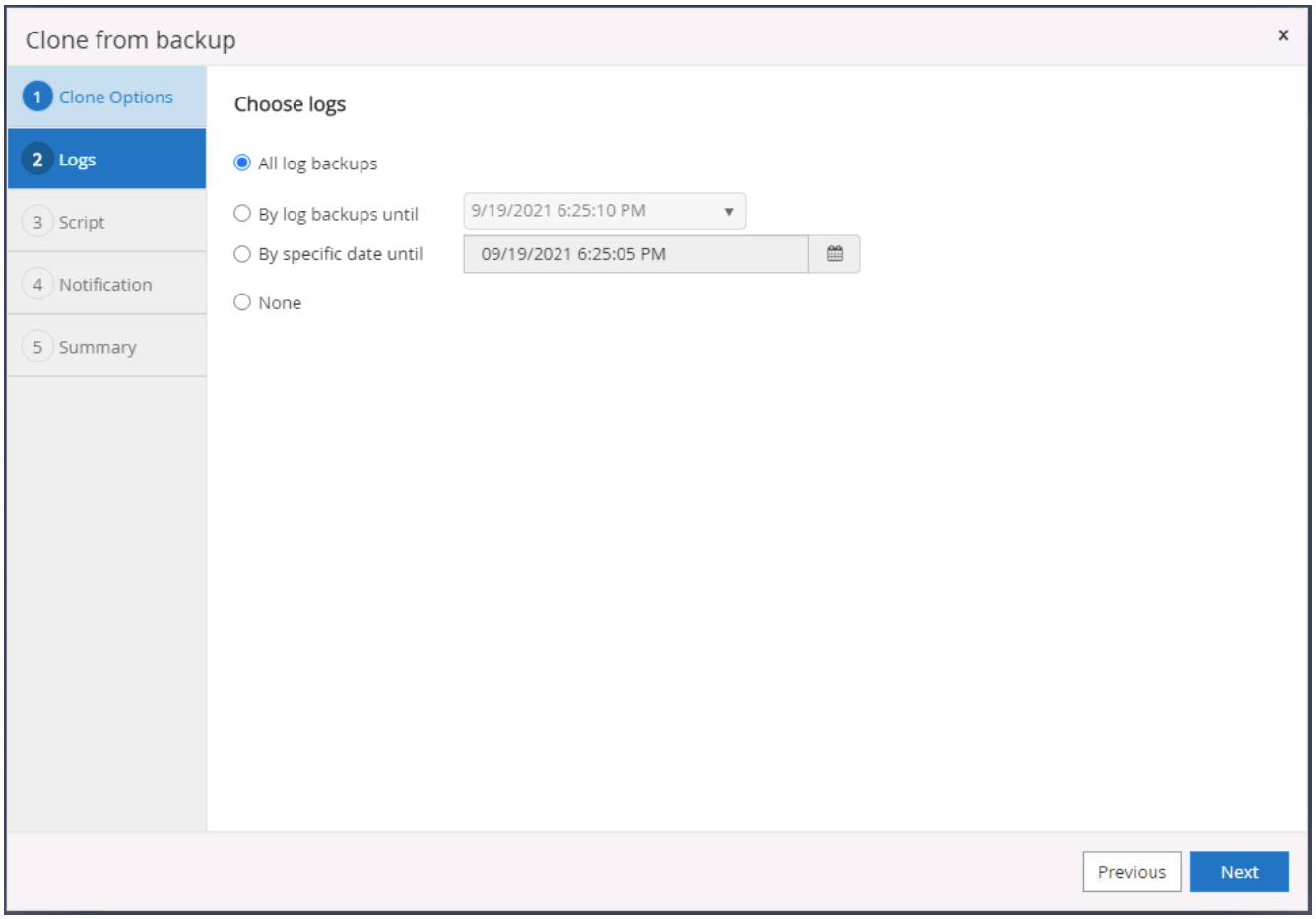
4. Select the last full SQL Server backup for the clone.



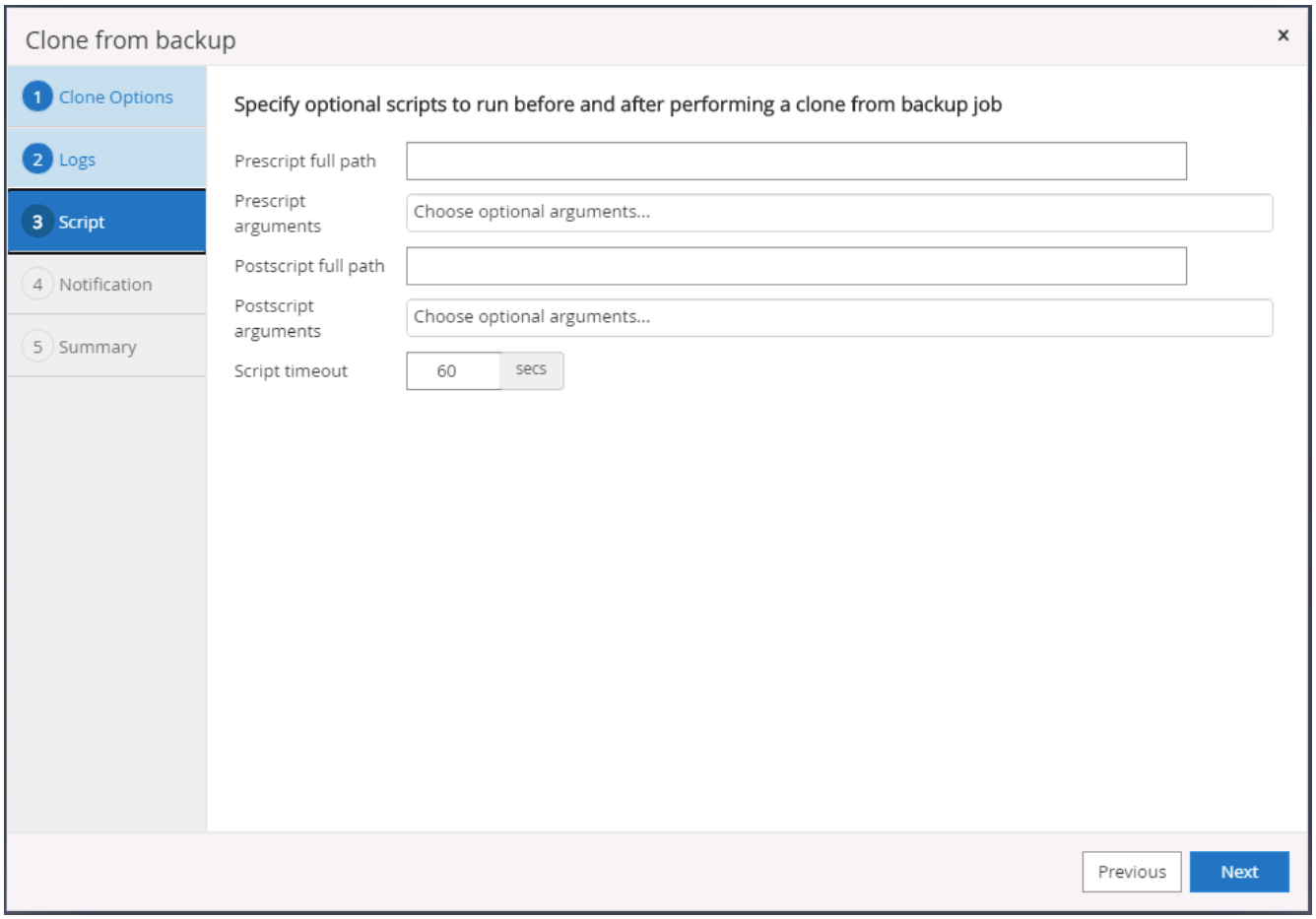
5. Set the clone setting such as the Clone Server, Clone Instance, Clone Name, and mount option. The secondary storage location where cloning is performed is auto-populated.



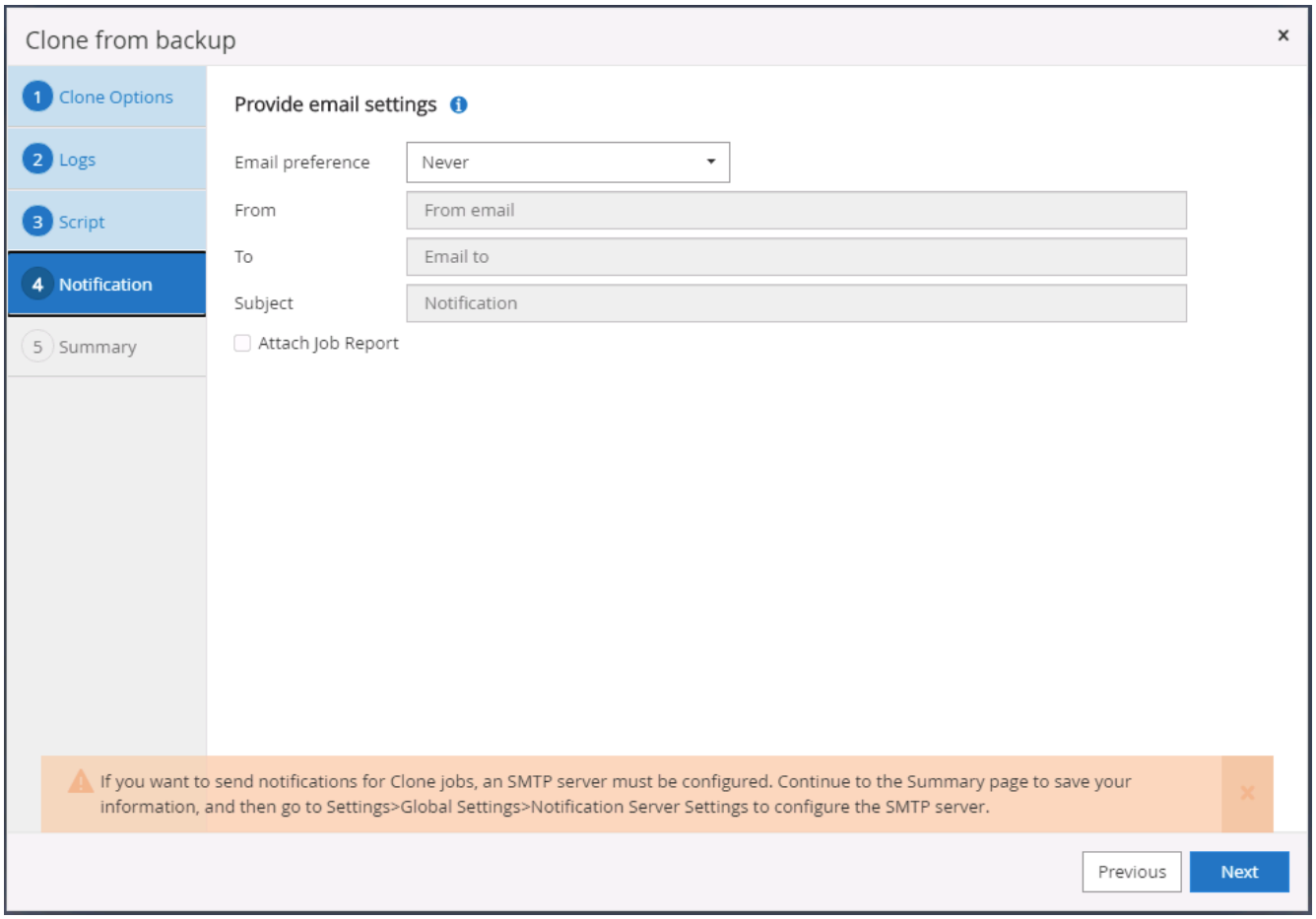
6. Select all log backups to be applied.



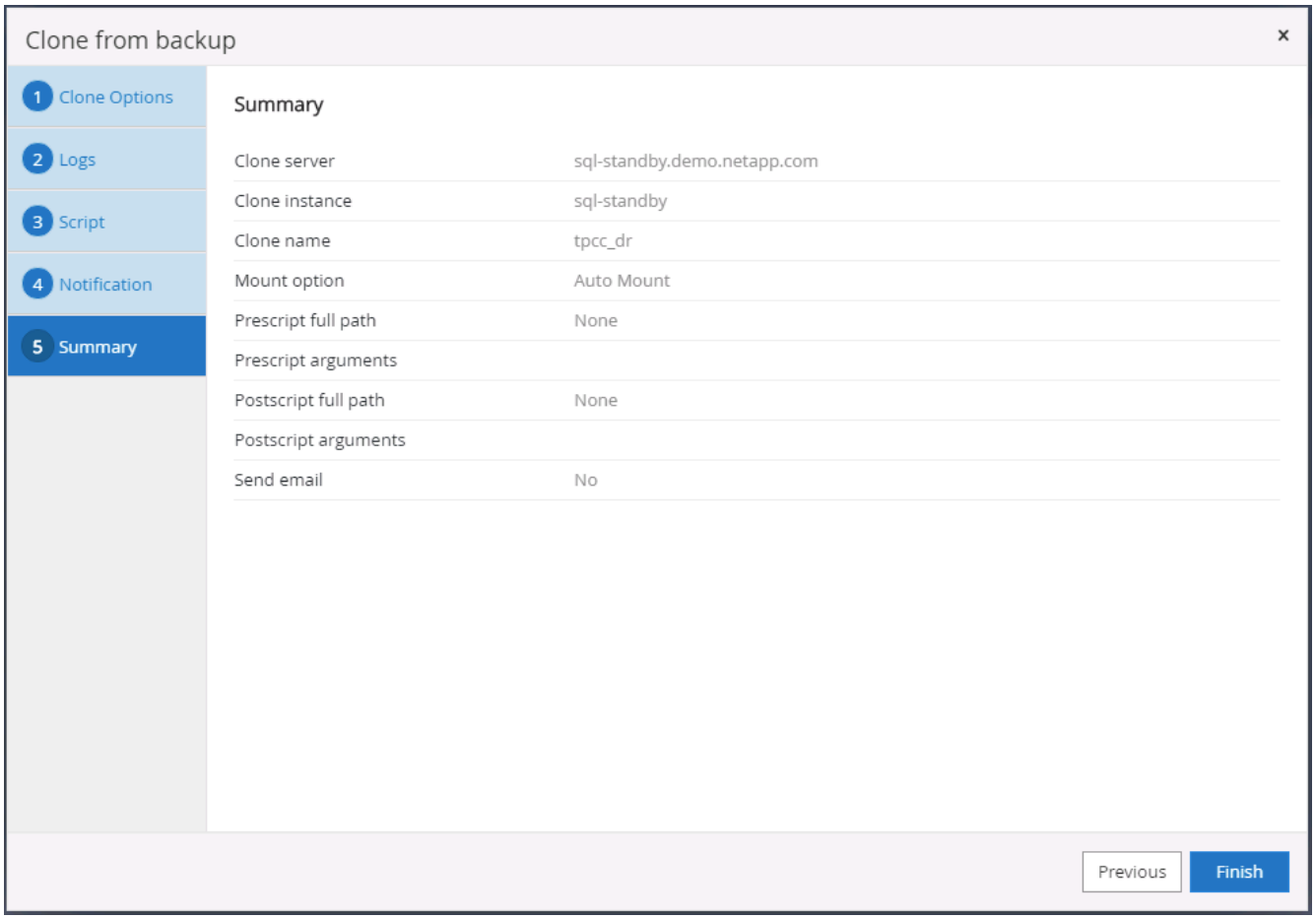
7. Specify any optional scripts to run before or after cloning.

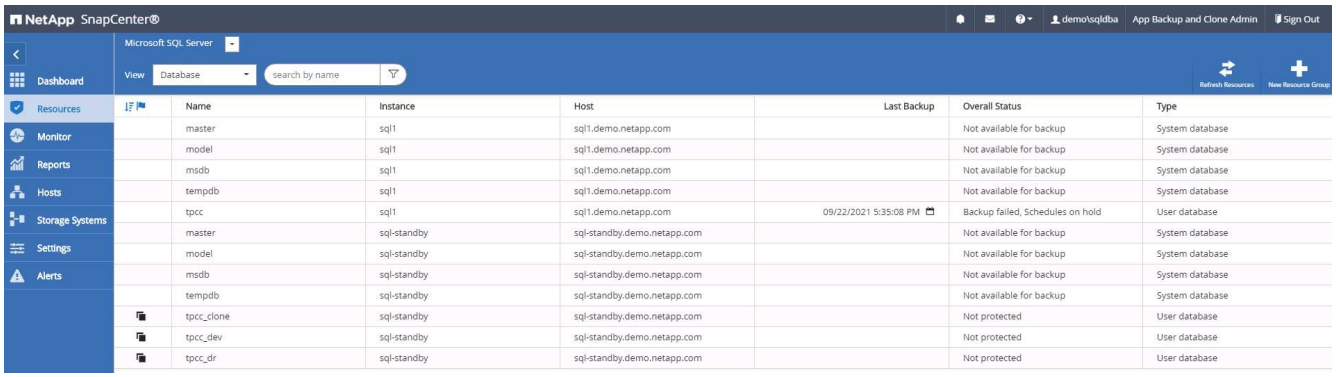


8. Specify an SMTP server if email notification is desired.



9. DR clone summary. Cloned databases are immediately registered with SnapCenter and available for backup protection.

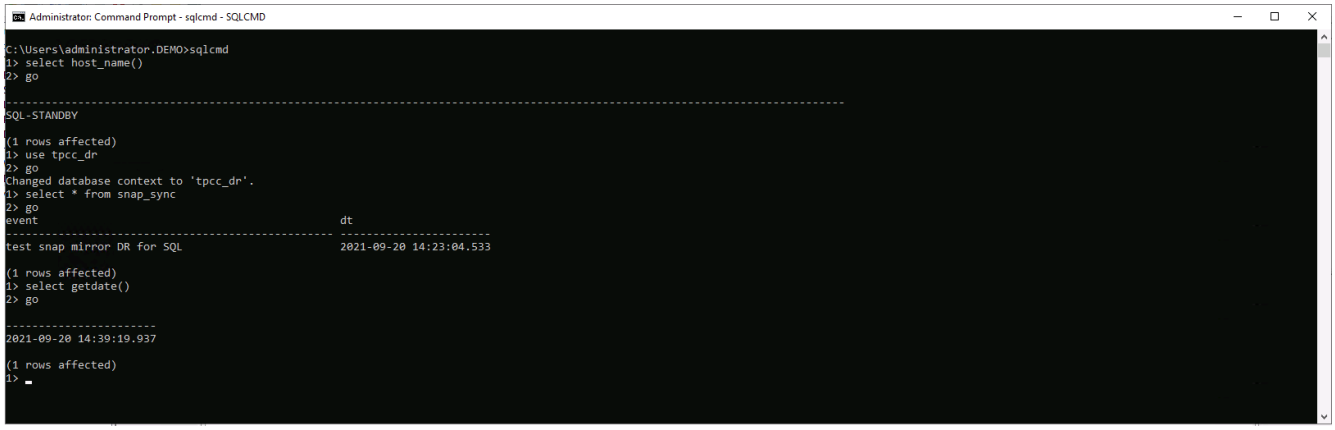




Post DR clone validation and configuration for SQL

1. Monitor clone job status.

2. Validate that last transaction has been replicated and recovered with all log file clones and recovery.



3. Configure a new SnapCenter log directory on the DR server for SQL Server log backup.
4. Split the cloned volume off of the replicated source volume.
5. Reverse replication from the cloud to on-premises and rebuild the failed on-premises database server.

Where to go for help?

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

Copyright information

Trademark information