ORACLE

Oracle WebLogic Server for Oracle Cloud Infrastructure Disaster Recovery

Production and DR in the Oracle Cloud Infrastructure (OCI)

February 2024 | Version 23
Copyright © 2024, Oracle and/or its affiliates
Public

PURPOSE STATEMENT

This document provides a description, a summary of requirements, and the setup procedure to configure a Disaster Recovery solution for Oracle WebLogic Server for Oracle Cloud Infrastructure stacks. This paper is oriented to a technical audience having knowledge of Oracle Cloud, Oracle WebLogic Server, Oracle Database, Data Guard and Oracle Database backup and recovery.

REVISION HISTORY

The following revisions have been made to this document:

Date	Revision	Comments
August, 2020	1	Initial publication
December, 2020	2	Added Best practices Added scale-out procedure for DR environment in Appendix C Added support for IDCS Authentication in DR scenarios
March, 2021	3	Added additional setup option (based in FSS with rsync method instead on DBFS). Several sections updated Added instructions for “Recreating the standby DB System” in the Appendix C
April, 2021	4	Added section "About having compute instances stopped in standby site"
May, 2021	5	Enhancements in wordings
July, 2021	6	Added additional setup option (based in Block Volume Cross-Region replication) in the Appendix E
September, 2021	7	Added footnote in page 5. Added OCI DNS switchover example in switchover operations.
October, 2021	8	Updated diagrams, additional info in Assumption > Database and note for recommended db connect strings
January, 2022	9	Updated manual Data Guard setup scripts and added section “RTO and RPO Overview”
April, 2022	10	Added point “Considerations for EXACS”
June, 2022	11	Added point “Patching” to “Appendix C – Additional Lifecycle Operations”
June, 2022	12	Updated links and removed references to Oracle Site Guard due to Site Guard deprecation (see Doc ID 2875372.1)
July, 2022	13	Added DNS views approach for the required host aliases Updated links
October, 2022	14	Added “Appendix F – Using additional standby Database in the primary” Added “End-to-End Validation of the Configuration Replication” in lifecycle procedures Moved “Open Secondary Site for Validation” to lifecycle procedues
November, 2022	15	Added clarifications for the OCI Policies in the “Provision WLS for OCI in Secondary Site” Updated role definition in crs services creation (physical standby role not mandatory)
February, 2023	16	Use tns alias in the datasources. Several sections updated.
May, 2023	17	Updated considerations for Block Volume DR method.
June, 2023	18	Improvements in “Custom Files” point
July, 2023	19	Added links to start/stop scripts
November, 2023	20	Highlight the usage of Volume Groups in Block Volume DR method.
December, 2023	21	Added post-failover actions
December, 2023	22	Reorg instructions to create CRS managed service for the PDB
February, 2024	23	Complete document reorganization. All the methods are in the body of the document.

Copyright © 2024, Oracle and/or its affiliates | Public

Contents

Purpose Statement 1

Revision History 1

Introduction 4

Disaster Recovery Solution for Oracle WLS for OCI 6

- Topology description 6
- Replication methods 7
- Assumptions 14
- Requirements 17

Download scripts 20

Disaster Recovery Setup Overview 21

Prepare for Disaster Recovery Setup 22

1. Choose a virtual front-end name 22
2. Prepare Primary mid-tier for the virtual front-end 22
3. Prepare primary mid-tier for using TNS alias 23
4. Setup the Database in Secondary Site 24
5. Provision WLS for OCI in Secondary Site 30
6. Prepare Secondary mid-tier for the virtual front-end 33
7. Prepare secondary mid-tier for using TNS alias 34
8. Configure required mid-tier host aliases 34

Complete the Disaster Recovery Setup 37

- Configure Using Block Volume Replica 37

1. Convert the standby DB into physical standby 37
2. Configure the Block Volume Cross-Region replication 37
3. Prepare the script for the environment specific replacements 39

- Configure Using DBFS and FSS with rsync methods 40

1. Configure the DBFS staging mount in DBFS method 40
2. Configure the FSS staging mount in FSS with rsync method 42
3. Run DR setup script in Primary WebLogic Domain 44
4. Run DR setup script in Secondary WebLogic Domain 45

- Complete DR configuration when IDCS authentication is used 48

Validate the DR Setup 54

Lifecycle Procedures for BV Replica Method 55

- Configuration Replication for BV Replica Method 55
- Open Secondary Site for Validation for BV Replica Method 55
- Switchover for BV Replica Method 58
- Failover for BV Replica Method 62
- Scale-out and Scale-in for BV Replica Method 63

Lifecycle Procedures for DBFS and FSS with rsync Methods 67

- Configuration Replication 67
- Open Secondary Site for Validation 70
- Switchover for DBFS and FSS with rsync Methods 71
- Failover for DBFS and FSS with rsync Methods 73
- Scale-out and scale-in for DBFS and FSS with rsync Methods 74
- Recreate the dbfs wallet 77

Copyright © 2024, Oracle and/or its affiliates | Public

Common Lifecycle Procedures **79**
About having compute instances stopped in standby site 79
About having different number of managed servers in primary and standby 79
Patching the WLS for OCI DR environment 80
Reassemble the WLS for OCI DR after recreating the standby DB System 81
RTO and RPO Overview **84**
Expected RTO 84
Expected RPO 85
Best practices **87**
Conclusion **88**
Appendix A – Considerations for RAC databases **89**
About the WebLogic on OCI Edition when using RAC 89
Network considerations when using RAC 89
Create and use a CRS database service 89
Apply other best practices 89
Appendix B – DB System Backups on manually configured Data Guard **91**
Appendix C – Using additional standby Database in PRIMARY **93**
Additional local standby Pre-Configuration Steps (DBFS and FSS with rsync Methods) 93
Additional local standby Cross-Region DR configuration (DBFS and FSS with rsync Methods) 94
Additional local standby Post-Configuration Steps (DBFS and FSS with rsync Methods 95
Additional local standby local database switchover (DBFS and FSS with rsync Methods) 98

Copyright © 2024, Oracle and/or its affiliates | Public

INTRODUCTION

Oracle's Maximum Availability Architecture (Oracle MAA) is the best practices blueprint for data protection and availability of Oracle products (Database, Fusion Middleware, Applications) deployed on on-premises, private, public or hybrid clouds. Implementing Oracle Maximum Availability Architecture best practices is one of the key requirements for any Oracle deployment. Oracle Fusion Middleware and Oracle Databases include an extensive set of high availability features which can protect application deployments from unplanned downtime and minimize planned downtime. These features include process death detection and restart, clustering, server migration, clusterware integration, GridLink datasources, load balancing, failover, backup and recovery, rolling upgrades, and rolling configuration changes.

Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WLS for OCI) is a **Platform as a Service (PaaS)** computing platform solution that allows you to create your Java Enterprise Edition (Java EE) application environment in Oracle Cloud Infrastructure very quickly. It is provisioned using the OCI Console Marketplace and is fully integrated with other OCI components (like OCI Load Balancer) and OCI infrastructure life cycle procedures (like backup and recovery). It uses Oracle Compute, Oracle Cloud Database, and Oracle WebLogic as its basic infrastructure. Oracle WLS for OCI can be configured as a JRF-enabled domain, to support the Oracle Application Development Framework (ADF), and to use Oracle Fusion Middleware Control console for the administration and monitoring. In this case, WebLogic uses the database to store the required schemas (like Oracle Platform Security Services information). For transactional consistency and simplified administration reasons, application data and WebLogic specific schemas are typically stored in the same database.

All critical WLS for OCI deployments need protection from unforeseen disasters and natural calamities, including within Oracle Cloud Infrastructure. This disaster recovery protection must address the middle tier (WLS for OCI), the data tier (Oracle Cloud Database) and Load Balancer (LBR) tier (OCI LBR or 3rd-party). The solution involves setting up a standby system at a Oracle cloud data center that is geographically remote to the primary production site. Although the standby system may have equal or fewer services and resources compared to the production site, Oracle recommends running **a mirror configuration with the same capacity**. The standby system is normally in a passive mode (it does not sustain the production workload), and is activated when the primary site becomes unavailable. This deployment model is sometimes referred to as an **active-passive model**.

Note that WLS for OCI stacks already provide High Availability in the scope of a single data center. WLS for OCI uses the Active High Availability (HA) policy for compute when it provisions compute instance nodes: virtual machines (VM) fail over automatically to another physical compute node in the same compute zone in case the primary compute node fails. On top of this, WLS for OCI uses different OCI Fault Domain by default for each compute instance of the cluster. In OCI regions with more than one availability domain, WLS for OCI provisioning places each compute instance in a different Availability Domain (AD). This happens by default whenever a regional subnet is used for the deployment. Similarly, the front-end LBR used by WLS for OCI is regional and provides failover across ADs by default (in regions with more than one AD). By protecting the Oracle Database used by WLS for OCI against AD failures with an Oracle Data Guard local standby (placed in a different availability domain), the WLS for OCI system will be completely resilient to any sort of outage in the scope of a compute instance, a fault domain or an availability domain. However, all this does not protect a WLS for OCI system against regional failures (affecting an entire region).

Copyright © 2024, Oracle and/or its affiliates | Public

This document has been particulary created to address **Disaster Recovery (DR)** for **Oracle WebLogic for OCI across regions**. This document does not apply to Java Cloud Service, which has been deprecated.

Oracle WLS for OCI Disaster Recovery can satisfy the most demanding **Recovery Time Objective (RTO) and Recovery Point Objective (RPO)** by utilizing high availability and disaster protection capabilities provided by Oracle Fusion Middleware and Oracle Database. While there are some unique considerations to a cloud disaster recovery configuration, it follows the same Oracle MAA best practices as any Oracle Fusion Middleware (FMW) and Oracle Database deployment. This Oracle MAA blueprint details Oracle MAA best practices and provides a procedural overview for deploying disaster protection for Oracle WLS for OCI. The Oracle WLS for OCI Disaster Recovery solution is achieved by replicating the configuration files required by the Oracle WebLogic Server domain. Custom applications deployed on the same WebLogic cluster may require additional files to be replicated. Options are provided in this document to suit different application paradigms. Disaster protection for the Oracle Cloud Database used by Oracle WLS for OCI is provided through Oracle Data Guard.

This document applies to **WLS for OCI stacks**. It is intended for a technical audience that has knowledge of **Oracle Cloud Infrastructure, Oracle WebLogic Server, Oracle Database, Data Guard, and Oracle Database backup and recovery.**

Copyright © 2024, Oracle and/or its affiliates | Public

Topology description

The Disaster Recovery solution for WLS for OCI is an **active-passive model**. There is a **primary system** that consists of a) a WLS for OCI stack, b) a load balancer, and c) an Oracle Cloud Infrastructure DB all deployed in in the same region and tenancy and a peer **standby system**, that consists of a WLS for OCI stack, load balancer, and Oracle Cloud Infrastucture DB system in a different region (same tenancy).

The terms "region", "data center" or "site" are used in this document indistinctly to refer to an Oracle OCI region. "Region", "data center" or "sites" are physical **location entities** that are **geographically separated**, **far enough** that should not be affected by the same disaster event. For example: Ashburn and Phoenix are two differnet data centers, sites, or regions in context of this document.

The primary and standby Oracle Cloud Infrastructure DB Systems are configured with Data Guard. By relying on **Data Guard** features, all the changes applied to primary database are replicated to secondary database (which acts as the "standby" database).

The standby WLS for OCI domain **is a replica of the primary domain**, that uses the same domain name, schemas, passwords, etc. but points to the secondary database. The listener addresses of the WebLogic Servers in the secondary data center are configured with the primary midtier host names, so in secondary, the pertinent aliases are created to resolve them with the secondary IPs. This document provides the steps to create and configure this standby system.

On the **front-end**, there is **a unique name** configured to access the applications that run in the system. This "virtual" front-end name will point to the IP of the OCI Load Balancer of the primary site. When a switchover takes place, this front-end name is updated to point to the IP of the OCI Load Balancer of the secondary site. The front-end hostname always resolves to the LBR's IP of the site that acts as primary at that point in time.

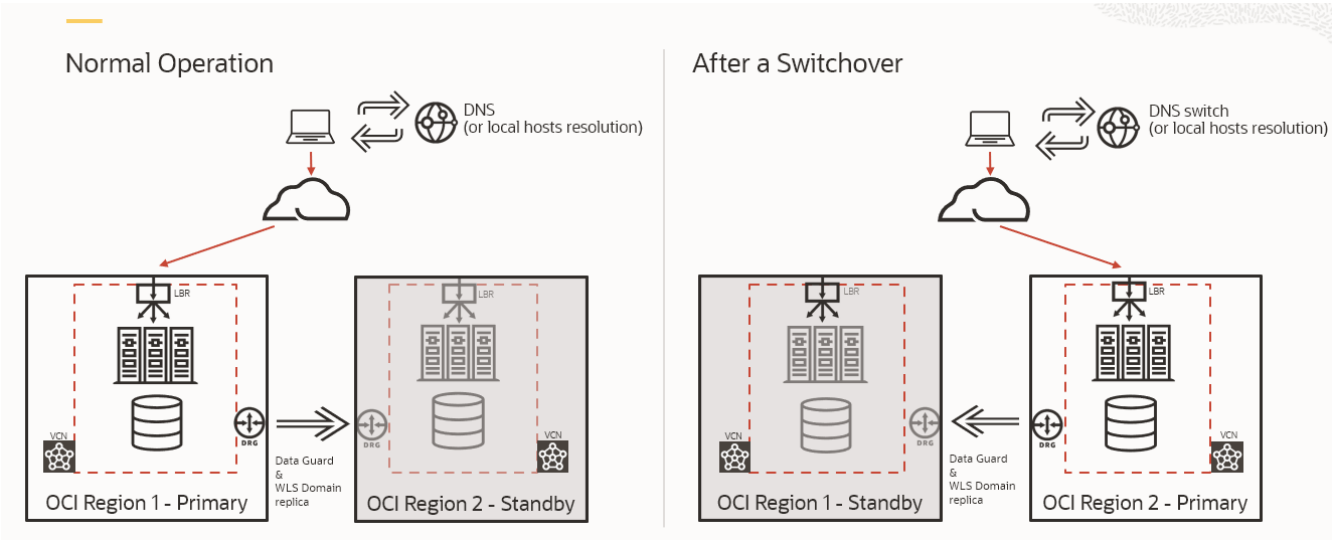


Figure 1 Oracle WebLogic for OCI disaster recovery topology

In normal business operation, the standby database is **a physical standby**. It is either in the mount state or opened in read-only mode when Active Data Guard is used. The standby database receives and applies redo from primary but cannot be opened in read-write mode. During some operations, like the initial DR setup and other lifecycle steps described in this document, the standby database is converted to a snapshot standby. A database in **snapshot standby** mode is a fully updateable database. It receives and archives, but does not apply, the redo data from a primary database. All the changes performed to a snapshot standby are discarded when it is converted again into a physical standby.

If the standby database is in shutdown status during normal business operation, it does not receive redo updates from primary and it will become out-of-sync. This can result in a data loss if a switchover needs to be performed. Thus, it is not recommended having the standby database stopped during normal business operation. The standby midtier hosts can be stopped to reduce incurred costs, however, the configuration changes that are replicated from the primary site will not be pushed to the secondary domain configuration if the standby site's admin server node is stopped. Depending on the configuration replication method used, this may affect differently the RPO of the system (see Replication methods below)

Copyright © 2024, Oracle and/or its affiliates | Public

Also, when a switchover event takes place, the RTO is increased if the standby midtier hosts need to be started and the domain needs to be synchronized with changes from primary. Oracle recommends having the secondary midtier hosts up (with WebLogic processes stopped). Check About having compute instances stopped in standby site for more details.

Replication methods

In this MAA solution, the information that resides in the database is automatically replicated to the secondary site by Oracle Data Guard. This includes MDS schemas, OPSS information, custom schemas, TLOGs, JDBC persistent stores, etc.

However, the WebLogic Domain configuration, located on the local filesystem, must also be replicated from the primary to the secondary. An initial replication is performed during the initial DR setup. It is also necessary to repeat this replication during the system' s lifecycle, ideally, whenever configuration changes are performed in the primary domain.

You can use two main approaches to maintain matching WebLogic domain configurations in both locations. The applicability of each depends on how frequently this "file-system-resident" configuration is modified:

a) For cases where the WebLogic domain configuration is **infrequently** altered it is **recommended to simply apply the configuration changes manually twice**: once in production and once in standby, by previously converting the secondary database to snapshot and starting the administration server. To maintain the WebLogic configuration synchronized by manually repeating the config change in the secondary site, follow these steps:

	STEP	DETAILS
1	Apply the configuration change normally in the primary site	Use the WLS Administration Console in the primary location to apply the configuration change. Activate the change, restart the required WLS servers if needed and verify that the change is working as expected.
2	Convert the standby database to a snapshot standby	Execute this as *oracle* user in the primary Database host: [oracle@drdba]$ dgmgrl sys/*your_sys_password@primary_db_unqname* DGMGRL> CONVERT DATABASE *secondary_db_unqname* to SNAPSHOT STANDBY; Converting database " *secondary_db_unqname*" to a Snapshot Standby database, please wait... Database " *secondary_db_unqname*" converted successfully
3	Start the WebLogic Administration Server on the secondary site	Follow the steps in the Oracle Cloud documentation to start the administration server. It is important to start ONLY the administration server and not the managed servers.
4	Repeat the configuration change in the secondary site	Use the WebLogic Administration Console in the secondary location to apply the configuration change. Activate the change and verify that the change is working as expected.
5	Stop WebLogic Administration server on the secondary site	Stop the WebLogic Administration server in secondary site.
6	Revert the database to physical standby	Execute this as oracle user in the primary Database host: [oracle@drdbaa ~]$ dgmgrl sys/*your_sys_password@primary_db_unqname* DGMGRL> CONVERT DATABASE *secondary_db_unqname* to PHYSICAL STANDBY; Converting database " *secondary_db_unqname*" to a Physical Standby database, please wait... Oracle Clusterware is restarting database "orclb" ... Continuing to convert database " *secondary_db_unqname*" ... Database " *secondary_db_unqname*" converted successfully

b) For those cases where the Weblogic domain configuration is modified **frequently**, Oracle provides three different methods to perform the WebLogic domain configuration replica: **DBFS (Oracle Database File System), OCI File Storage Service (FSS) with rsync, and Block Volume Cross-Region Replication**. The three methods use the same

Copyright © 2024, Oracle and/or its affiliates | Public

topology and similar mechanics. The difference between them is how the information is transferred from the primary site to the standby.

- In the **DBFS-based method**, a copy of the domain configuration is staged to a DBFS filesystem and replicated to the secondary site via Data Guard. A DBFS mount is a filesystem exposing information that resides in the database and that can be mounted like an NFS volume. The primary domain configuration is copied to that DBFS mount, and then, it is automatically replicated to the standby via the underlying Data Guard functionality. In the secondary site, the midtier hosts mount a DBFS mount point from the same DB tables as primary (replicated to the standby database). The replicated domain configuration data is now available and copied from the DBFS mount to the secondary domain. This document provides a script that automates this process in primary and in standby. Both sites run this script on a cron basis or by a schedule, to replicate the configuration at the desired frequency. See the section below for a comparison of the different replication approaches.

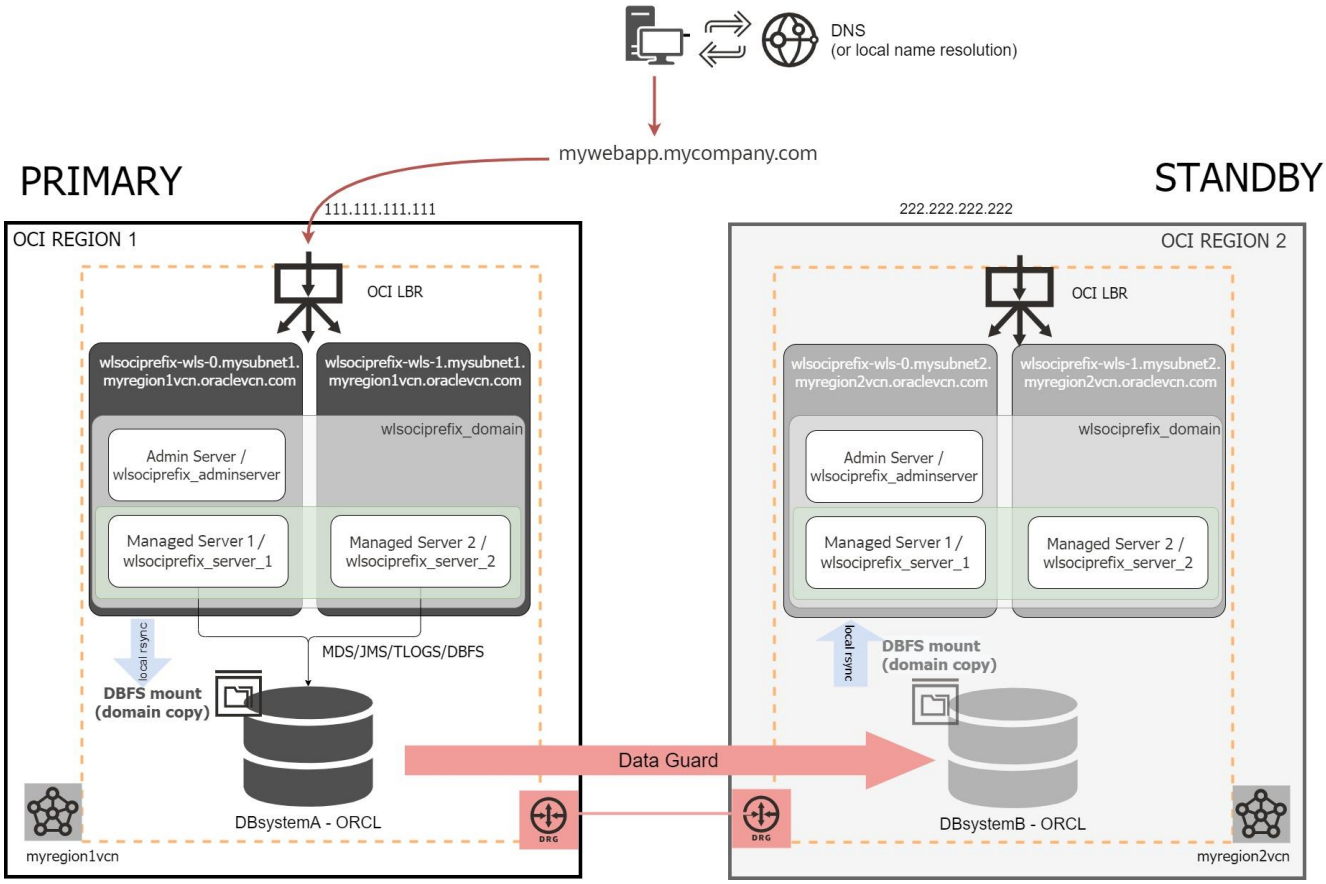


Figure 2 Complete WLS for OCI DR topology, using DBFS based method for WLS Domain config replication.

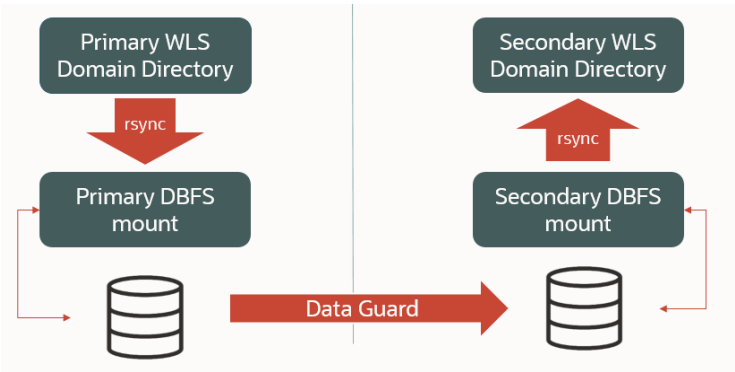


Figure 3 DBFS based method for WLS Domain config replication logical flow diagram.

Copyright © 2024, Oracle and/or its affiliates | Public

- In the **OCI File Storage Service (FSS) with rsync method**, the domain configuration is transferred to the secondary site by using direct rsync between two FSS file systems, one in primary and another in secondary. This approach, like the DBFS method, uses a shared filesystem as an intermediate "staging" point. For this, an OCI FSS is mounted in each region. To replicate the primary domain config, the WLS domain folder is copied first to the local staging OCI FSS mount, and then, via rsync, to the remote OCI FSS mount. Then, in secondary, the domain configuration is copied from the OCI FSS in the secondary environment's data center to the secondary domain directory. This document provides a script that automatizes this process in primary and in standby. Both sites run this script on a cron basis or by a schedule, to replicate the configuration at the desired frequency. Notice that, although FSS provides cross region replication, a WebLogic domain cannot reside directly on FSS. Hence, it is always needed to copy first the domain to a stage directory. Hence, it is this stage copy which drives the main replication RTO and RPO. See the section below for a comparison of the different replication approaches.

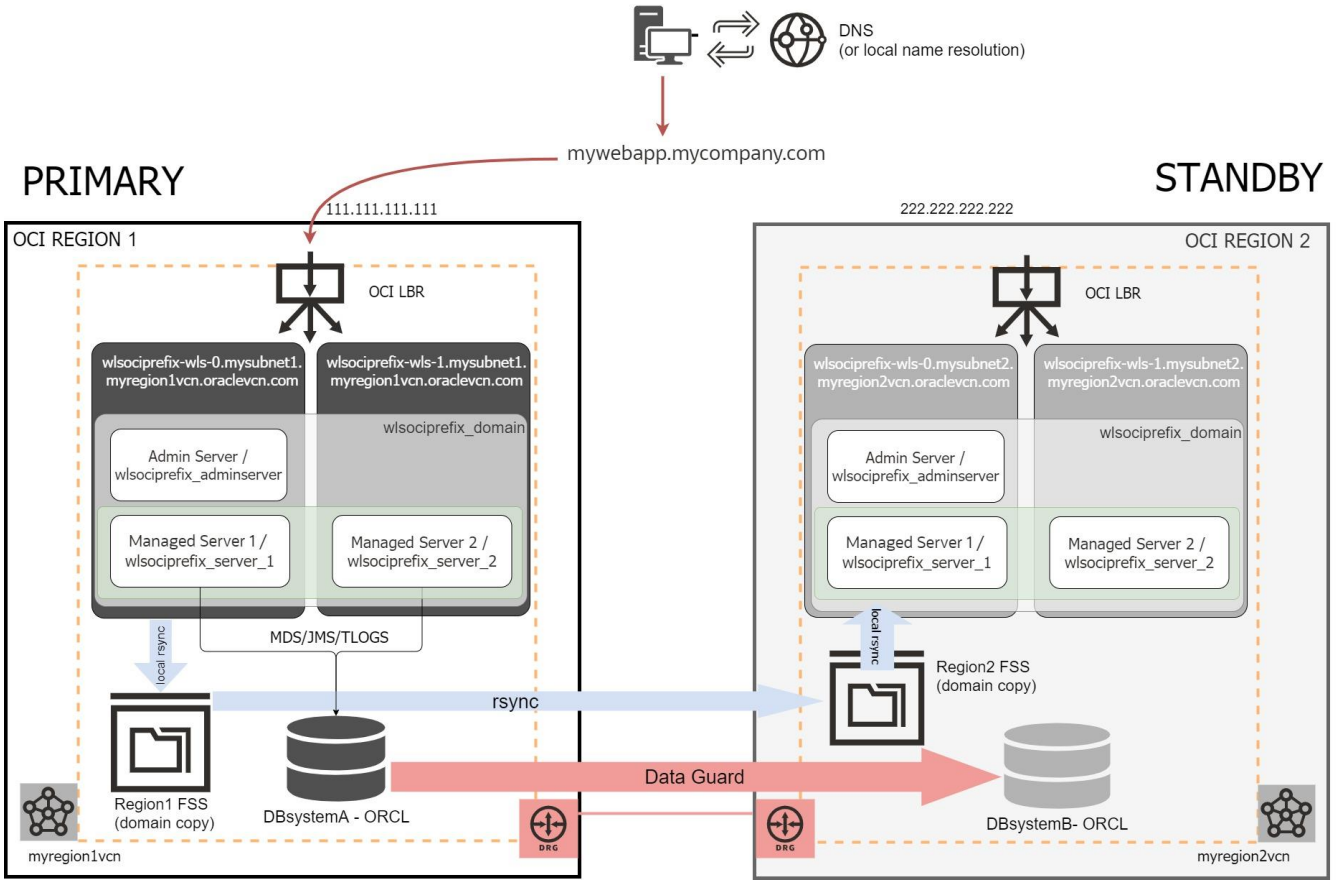


Figure 4 Complete WLS for OCI DR topology, that uses OCI FS with rsync method for WLS Domain config replication. The blue arrows just represent the logical flow of the configuration copy. The rsync commands run either in primary or standby site's WebLogic Administration hosts. I.e., for the remote copy, primary site's WebLogic Administration host connects to standby WebLogic Administration host with rsync.

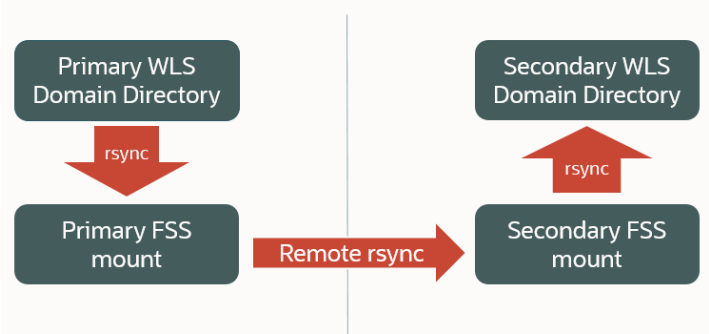


Figure 5 OCI FS with rsync method for WLS domain config replication logical flow diagram

Copyright © 2024, Oracle and/or its affiliates | Public

- In a **Block Volume cross-region replication** model, the entire block volume of the mid-tier hosts that contains the WebLogic Domain configuration is replicated to the secondary site using the OCI Cross-Region Volume Replication feature. Block Volume cross-region replication performs ongoing, automatic asynchronous replication of block storage volumes to other regions. This approach does not use a stage location for configuration replication.

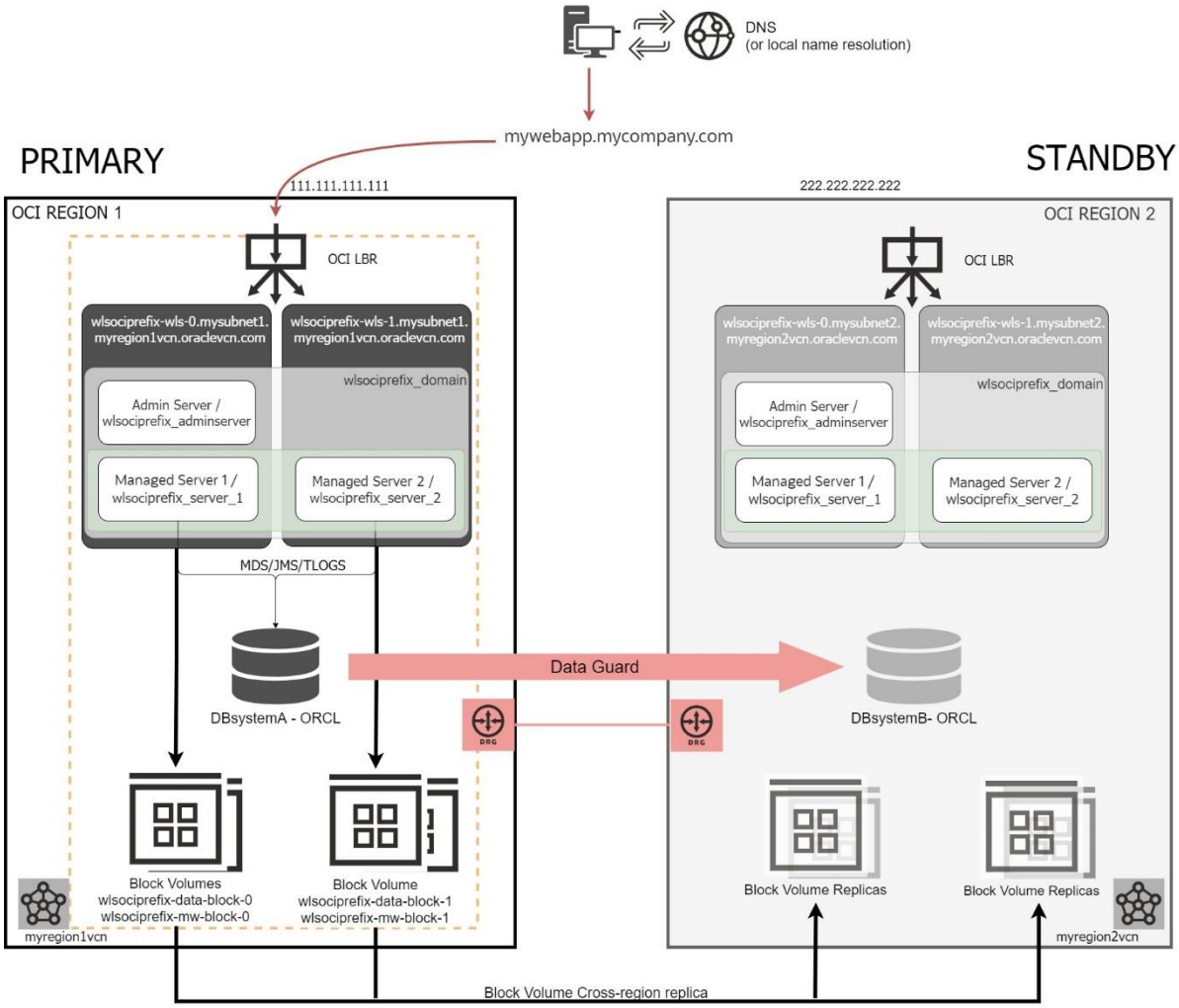


Figure 6 Detailed WLS for OCI Disaster Recovery topology based in Block Volume cross-region replication

Copyright © 2024, Oracle and/or its affiliates | Public

Replication Methods Comparison

- **Overall behavior:**
  - The **DBFS method** delivers higher local and remote availability through Oracle Driver's retry logic and provides a more resilient behavior than FSS with rsync or BV replication. When a RAC database is used to host the DBFS tables, the DBFS client can survive backend storage failures and can failover connections to the surviving database instance. This method takes advantage also of the **robustness of the Data Guard replica**. It can be used in any scenario, and it is recommended for DR cases that have high latencies between the regions since the Dataguard copy provides block and data verifications through Dataguard that are not available in other replication methods. However, the use of DBFS for configuration replication has also additional implications from the setup, database storage and lifecycle perspectives. It requires certain database maintenance (to clean, compress and reduce table storage) and a good understanding of how DBFS mount points behave.
  - The **FSS with rsync method** is easier to maintain and configure than the DBFS method. However, it is recommended only across Oracle OCI data centers with low latency between then and that have been configured to use a Dynamic Routing Gateway. Data centers that communicate over the public internet, for example, may not have a sufficiently low latency for a reliable behavior of FSS with rsync. Notice also that the FSS with rsync method can incur in additional costs due to the FSS usage and to the connectivity requirements between primary and standby middle tiers (customer billing conditions are out-of-scope of this document, contact your Oracle license team to get details on this).
  - In the **Block Volume replication** approach, the set up and ongoing replication process is simpler and differs significantly from the DBFS and FSS approaches. This model provides a continuous and automated replica. This model can be used as a general-purpose solution applicable not only to middleware-based PaaS services but also to all the data that may reside in block volumes attached to a compute instance. It requires a few additional steps (as compared to the DBFS and FSS with rsync methods) to prepare a switchover. However, these operations can be done in advance to the switchover to improve the total RTO[1]. Notice that it is not possible however to share a block volume across compute instances without a cluster file system on top. Thus, if shared storage is required by the middle tier, the simplicity and automatation benefits of the BV approach does not apply to the overall replication needs.
    The **Block Volume replication** is more agnostic to the specific system configuration. It is generally applicable to other systems, apart from WLS for OCI, and can be used to replicate any other compute instance's block volumes in the system. It provides an automatic and continuous replication process performed by OCI infrastructure (not by any manual or croned scripts as in the DBFS and FSS cases).

    With Block Volume replication, data synchronization is not limited to the Weblogic domain configuration. The block volume that contains the WebLogic domain folder in each compute instance is <wlsprefix>-data-block-N. It is mounted in /u01/data. This block volume needs to be replicated to the standby site on an ongoing basis. The information of the replicated block volumes is an exact copy from the primary block volumes. This means that other custom files that are located outside the WebLogic domain (as long as they are in the block volume that is replicated) can also be replicated automatically without additional intervention.
    For the first switchover, you need to replicate the data block volumes of all the nodes. For subsequents switchovers, you can use different approaches to replicate the WebLogic configuration when using Block Volume replication:
    1. Use BV replication to replicate only the Weblogic Administration Server's block volume and let the Weblogic Infrastructure propagate that configuration to the managed server nodes. All the configuration under the domain/config directory is copied by WLS to the other nodes that are members of the domain when they start. This approach is recommended if there are no customizations and artifacts that reside outside the Weblogic configuration directory.
    2. Replicate also the Weblogic managed server nodes' block volumes. This is useful when additional artifacts and customizations are used ni each Weblogic node but incurs in higher complexity and cost.

- **Management Complexity:**

[1] In failover situations, however, these steps can cause additional operational overhead.

Copyright © 2024, Oracle and/or its affiliates | Public

The configuration replication procedures in the **DBFS and FSS methods** are pretty similar, irrespectively of the number of nodes in the Weblogic Domain.
Contrary to this, **Block Volume cross-region replication** management gets more complicatedas the number of replicated block volumes increases. It requires a good lifecycle management of the block volumes and replicas. Switchover and failover operations are also more complex than in the other methods, and there are additional pre and post switchover/failover steps. This complexity increases when the Weblogic Managed server block volumes are replicated (besides the Administration server's).This complexity, however, can be reduced drastically if you use Full Stack DR to automate the switchovers and failovers of the system

- **Cost implications**
  In the **DBFS** approach the increased costs are related to the additional storage required in the Database to host the DBFS tables. Typical DB tablespace and storage maintenance operations are required for an efficient recovery of allocated space. The network overhead in the cross-region copy is typically neglectable in comparison to the overall Dataguard traffic requirements.
  In **FSS with rsync**, the storage requirement is low since file storage allocation for a weblogic domain is typically a few gigabytes and the copy over rsync does not have a big network impact either.
  With **Block Volume replication**, however, once you enable replication for a volume, the volume will be replicated in the specified region and availability domain. Your bill will include storage costs for the volume replica in the destination region. The volume replica in the destination region is billed using the Block Storage Lower Cost option price, regardless of the volume type in the source region. This cost increases when the Weblogic Managed server block volumes are replicated (besides the 12dministration server's). Your bill will also include any applicable network costs for the replication process between regions. As part of the replication process, all data being updated on the source volume is transferred to the volume replica, so volumes with continual updates incur higher network costs.
  See point "Cost Considerations for Cross-Region Replication" in the Oracle documentation Cross-Region Volume Replication.
  On the other hand, the copy to the stage directory in the DBFS and FSS scenarios does have an impact on the Weblogic Administration node and additional memory and cpu resources may be required in it depending on the frequency of the copy, how big the weblogic domain is and whether other files on shared storage need to be replicated across regions in the same replication cycle.

- **Recovery Time Objective (RTO)**
  The switchover RTO is similar in the DBFS, FSS with rsync, and Block Volume replication approaches. For failover operations, however, additional steps are required by block volume replication (activate replicas, attach block volumes, etc.). These increment the downtime during the failover. For a normal failover of a 2 nodes WebLogic cluster, the RTO is increased in around 10 minutes.
  There are also post-switchover and post-failover tasks (detach block volumes, etc.) that will require additional time. Although this last set of steps can be performed without incurring in additional recovery time and can be automated with FSDR.

- **Recovery Point Objective (RPO)** (referring to WebLogic Configuration. The RPO for the database is exactly the same in all the replication methods)
  The **Block Volume replication** process is continuous, with the typical Recovery Point Object (RPO) target rate being less than an hour. However, depending on the change rate of data on the source volume, the RPO can vary. For example, the RPO can be greater than an hour for volumes with a large number of I/O operations to the volume. Refer to https://docs.oracle.com/en-us/iaas/Content/Block/Concepts/volumereplication.htm#volumereplication for more deails.
  In the **DBFS and FSS with rsync** methods described in this document, the user can have a finer control on the RPO for the WebLogic Configuration because a) the information is replicated using a scheduled script and b) the amount of information replicated is lower than the entire block volume. On the other hand, in the DBFS and FSS with rsync methods, it is the administration server's compute instance the one that acts as "manager" of the domain configuration received, so this nodes availability and capacity drives the speed of the configuration copy. When using **Block Volume** replication, the replication keeps taking place regardless of the secondary nodes being up and running.

The following table summarices the aspects discussed above for each replication method:

Copyright © 2024, Oracle and/or its affiliates | Public

	DBFS method	FSS with rsync method	Block Volume cross-region replica method
Stage storage used	Yes	Yes	No
Storage Setup complexity	Medium	Medium	Low
Storage management complexity	High	Medium	Low (*)
DR setup complexity	Medium	Medium	Low
Switchover and Failover complexity	Low	Low	Low (*)
WLS config replica complexity	Medium	Medium	Low
RTO for Switchover	Typically, 15-30 min	Typically, 15-30 min	Typically, 15-30 min (+ around 10 min for pre and post switchover tasks that do not incur in downtime, can be automated with FSDR)
RTO for Failover	Failover time (typically, 15-30 mins) + unplanned outage time	Failover time (typically, 15-30 mins) + unplanned outage time	Pre-failover tasks time (typically 10 mins) + Failover time (typically, 15-30 mins) + unplanned outage time
RPO (runtime data)	Depends on Data Guard for DB persisted data. 1 hour for shared storage on FSS artifacts See "Expected RPO"	Depends on Data Guard for DB persisted data. 1 hour for shared storage on FSS artifacts See "Expected RPO"	Depends on Data Guard for DB persisted data. 1 hour for shared storage on FSS artifacts See "Expected RPO"
RPO (WLS config)	Depends on the replica script execution frequency. See "Expected RPO"	Depends on the replica script execution frequency. See "Expected RPO"	Typical RPO is significantly less than thirty minutes, but it can vary depending on the change rate of data on the source volume.
Infrastructure Cost	Low	Low	Medium
(*) When Full Stack Disaster Recovery Service (FSDR) is used			

As an **Oracle Maximum Availability Architecture** best practice, Oracle recommends using **block volume replication with OCI Full Stack Disaster Recovery Service**. OCI Full Stack Disaster Recovery Service and Block Volume replication provide the best combined benefits for recovery time objective (RTO), recovery point objective (RPO), total cost of ownership (TCO), and management automation.

Copyright © 2024, Oracle and/or its affiliates | Public

Assumptions

The following consideration are key aspects for the WebLogic Cloud Disaster Recovery setup:

WebLogic Editions

There are three different WebLogic on OCI editions:

- Oracle WebLogic Server Standard Edition
- Oracle WebLogic Server Enterprise Edition
- Oracle WebLogic Suite

See WebLogic Chapter in Licensing Information User Manual for more information about their features. The use of Oracle WebLogic Server GridLink data sources, known as Active GridLink for RAC, is an entitlement available only as part of licensing WebLogic Suite. The other editions do not include Active Gridlink, so they use Multi Datasources to connect to a RAC database. It is an MAA best practice to use GridLink datasources when RAC is used.

This document applies to Oracle WebLogic Suite Edition and to WebLogic Enterprise Edition, except when Oracle RAC DB is used. **When the database is a RAC**, **this document assumes that Oracle WebLogic Suite Edition** is used. This DR setup supports only that edition for RAC scenarios because it is the only edition that uses GridLink datasources.

Authentication

Oracle WLS for OCI can be provisioned with the option "*Enable Authentication using Identity Cloud Service*". In that case, it uses Oracle Identity Cloud Service (IDCS) to authenticate users in the domain, instead of using the default Oracle WebLogic Server Identity Store. This adds additional components to the system, like the Cloud App Gateway agents and Identity Cloud Services.

This document addresses WLS for OCI systems that use the **default Oracle WebLogic identity store**, and **also** the implications of **using IDCS authentication** for WLS for OCI in context of a DR event. Notice, however, that IDCS's own disaster protection is out of the scope of this document[2].

This does not preclude customers from customizing the WebLogic identity store to use another external LDAP (as long as WebLogic supports it for OCI). As long as the WebLogic provider's configuration resides in the domain config, it will be replicated to the standby domain. If the standby WebLogic system can reach to the configured external LDAP url, it can be used. Note that the DR solution for any external LDAP service is out of the scope from this WLS DR document and should be provided by the specific LDAP product.

Load Balancer

The Disaster Recovery solution **assumes that the WLS for OCI stack is configured with an OCI Load Balancer.** A load balancer is mandatory when the cluster has more than one server, so the incoming requests can be balanced between them.

The default Load Balancer that is created during the provisioning is an OCI Load Balancer. Depending on your network topology, it can be public or private.

A **public** OCI load balancer is regional in scope. It requires either a regional subnet (recommended) or two AD-specific subnets, each in a separate availability domain. If your region includes multiple availability domains, it creates a primary load balancer and a standby load balancer, both in the same region but each one in a different availability domain. If the primary load balancer fails, the public IP address used for incoming traffic switches to the standby load balancer that is in the same region. The service treats the two load balancers as equivalent, and you cannot specify which one is "primary". This way, the load balancer provides local (inside a region) high availability for the load balancer layer.

[2] Contact IDCS support to check the IDCS Disaster Recovery solution availability in your region and to get details on how IDCS itself is protected against disasters.

Copyright © 2024, Oracle and/or its affiliates | Public

The same topology will exist in the secondary region: the OCI LBR in the secondary domain will have one primary load balancer in one of the availability domains of the secondary region and another one in the other availability domain of the secondary region.

With a **private** OCI load balancer a similar topology is used, although it only uses one subnet to host primary and standby. OCI private load balancers are used to isolate a system from public internet access. Its subnet can be regional or AD-specific, depending on the scope of the subnet it uses. If the subnet is regional and the region contains multiple availability domains, primary and standby load balancers are placed in different ADs. If the subnet is AD-specific, primary, and standby are in the same AD (in this case it has no failover capability in response to an availability domain outage).

In summary your OCI Load Balancers can be public or private, depending on the client access requirements. For implicit High Availability features in multi-AD regions, place the OCI Load Balancer, whether public or private, in a regional subnet.

This configuration is sufficient for the disaster recovery configuration. No configuration replication is required between primary and standby site's Load Balancers, as each needs to route only to its local WebLogic cluster. Any configuration changes that are applied to the load balancer configuration in the primary site need to be applied also manually to the secondary site's load balancer.

See documentation for OCI Load Balancing for additional details.

Database

During the Oracle WLS for OCI provisioning, you can choose to associate it to an existing database. The usage of an Oracle Cloud Infrastructure Database System along with Oracle WebLogic Cloud is the most common scenario:

- Database is mandatory when customers want to use features like **Fusion Middleware Control Console** to monitor, configure and manage the domain, OPSS Security, ADF Framework, etc. (known as "JRF-enabled" domains.).
- It is also an **MAA best practice** to use a database **for any persistent information** stored by the WebLogic Server domain, like JMS persistent stores or JTA logs. Especially in Disaster Recovery topologies, where this information can be automatically available in the secondary site in case that secondary becomes the primary thanks to the Data Guard replication.
- In addition, application's runtime data is usually stored in custom schemas in the database.

The Disaster Recovery solution assumes that the WLS for OCI is configured **with an Oracle Cloud Infrastructure database**. This document precisely focuses and uses Database Systems VM on OCI for the examples and configuration provided.

Only one standby database per primary database is supported in the DR topology provided by this document. This is consistent with the OCI Console, which limits the DG configuration to only one standby database for each primary database.[3] If your system uses an additional standby in the primary region (which will need to be managed **manually** since OCI Console does not support such a configuration), or you plan to add it later, see Appendix C – Using additional standby Database in PRIMARY.

The DR Setup procedure described in this document is **certified** with **Oracle Base Database Service DB Systems (single instance and RAC),** and with **Oracle Exadata Database Service (aka EXACS).**

WLS for OCI with Oracle OCI Autonomous Database requires specific management of wallets and connect strings. Refer to the playbook *Configure Oracle Fusion Middleware DR on Oracle Cloud with an autonomous database* for setup details.

Block Volumes Replicated in BV Replica Method

In the BV Replication DR solution **only Block Volumes are replicated** to the other site. The **Boot Volumes are not replicated**.

[3] See "Using Oracle Data Guard" in https://docs.oracle.com/en-us/iaas/Content/Database/Tasks/usingdataguard.htm

Copyright © 2024, Oracle and/or its affiliates | Public

Each mid-tier host of a **WLS for OCI** domain has **two Block Volumes** attached: <wlsociprefix>-data-block-N (mounted in /u01/data and it contains the WebLogic domain) and <wlsociprefix>-mw-block-N (mounted in /u01/app and it contains the Oracle software homes and inventory).

Any other **content outside these folders (/u01/data and /u01/app)** resides in the Boot Volume and **will not be replicated**. The Operating System is stored in the Boot Volume; hence, it will not be replicated. If the OS is patched or modified in primary mid-tier hosts, the same patching procedure must be performed in the secondary WLS for OCI compute instances.

This **document assumes** that **only the data block volumes** (<wlsociprefix>-data-block-N) **are replicated on an ongoing basis,** and the steps for the mw block volumes (<wlsociprefix>-mw-block-N) are included as optional.

You can also replicate the block volumes that contain the Oracle software homes (<wlsociprefix>-mw-block-N), but **you don't need to replicate them on an ongoing basis**. Their content does not change unless the WebLogic software or the JDK is patched/altered. Including it in the same replication strategy as the configuration volume would unnecessarily increase the cost and the complexity of the solution. It is possible, however, to replicate this volume sparsely to propagate patches from primary to secondary. These are the different approaches you can use to manage these volumes:

- You can include the mw block volumes in the ongoing cross-region replication strategy (just like the configuration volumes); in that case, any patch applied to the primary Oracle homes will be replicated to the secondary. However, this will cause higher management overhead and higher cost.
- An optimized approach is to replicate these volumes sparsely. For example, when the primary peer has been patched. This will incur lower cost and fewer steps when performing a switchover. In this model, patching is performed only in one site and copied to the secondary.
- As an alternative approach, the "software" volumes can be completely skipped from the replication strategy. In this case, when primary is patched, the same patching procedure must be applied in the secondary mid-tier instances. This will provide fewer steps in switchover operations and lower price solution at the cost of not automating the replication of patches (you will need to patch the secondary system manually. But this can be useful also to test fixes in secondary first).

.

Copyright © 2024, Oracle and/or its affiliates | Public

Requirements

The following sections describe requirements for the setup and lifecycle of the disater rptoection system to work properly. Notice that some requirements are specific to the replication approach used in each case. This is indicated in the header of the section when applicable.

Front-end address

The access from clients to the system must be agnostic to the site that is being used as primary. To accomplish this**, the front-end address host name used to access the system must be unique** and always maps to the IP of the system that is the primary at that moment. This name is usually referred to as **"virtual front-end"** or "vanity url".

You can reuse the existent system's front-end host name address (if such exists already) as the virtual front-end for disaster protection. For example, if the original system was using "mywebapp.example.com" as the vanity url for primary, this same virtual hostname can be re-mapped to the second site's load balancer IP after a switchover or failover.

Use the appropriate DNS services (Oracle Cloud DNS, other commercial DNS, local DNS, or local hosts resolution) to map the virtual front-end name to either site. This document explains how to configure WebLogic for OCI to use the virtual front-end name.

WebLogic Resource Name Prefix

During the provisioning of a WLS for OCI stack, you provide the "Resource Name Prefix". This property is used to construct the names of many resources used by the stack, including: the WebLogic Server domain name, the cluster name, the Weblogic server names, the VM's hostnames, etc.

This property must be **the same value in the primary and secondary WLS for OCI systems**, so that both systems have the same name for the WebLogic resources. Using the same name guarantees consistency and is required for the recovery of JMS messages and TLogs. It also simplifies customizations and operations in both sites.

Note that there is no problem in using the same "Resource Name Prefix" in multiple instances in the same Cloud tenancy, as long as they are created in different regions and/or compartment. Each instance is shown only in its specific region and compartment.

Network communication between sites

The primary and standby databases need to communicate with each other over their listener port for redo transport. In the DBFS replica model, the secondary middle tier hosts need to communicate with the primary database for the initial setup also.

When you use the FSS with rsync method, the WebLogic Administration host at each site needs to communicate via ssh (TCP/22) with the remote peer WebLogic Administration host for the rsync copy.

Oracle recommends using OCI's internal network with Dynamic Routing Gateways for the communication **between primary and secondary sites** (refer to the Dynamic Routing Gateway documentation for additional details on the network configuration). Whenever possible, use private IPs for connections between nodes. The communication between sites can also happen over an Internet Gateway (Oracle Net's traffic is encrypted), but this is not a recommended approach.

Enable the appropriate ingress rules for your case. Security rules are configured in the Security Lists for each Virtual Cloud Network in the OCI console. More information about this is available in Security Rules section on the OCI documentation.

The amount of database data replicated across sites depends on the redo generated by the primary database, and this is directly related with application load, its transactionality, concurrency, etc. In the DBFS approach, the database overhead caused by the configuration replication is typically irrelevant compared to the runtime data that Data Guard synchronizes. To ensure a timely delivery of the redo log files to the standby database, a suitable network connection between the primary site and the secondary site must be provided. Oracle Cloud Infrastructure regions are interconnected with high-bandwidth, fault-tolerant networks achieving ≥ 99.95 percent reliability (≤5 packets lost in 10,000), which also provides a consistent latency. See Oracle Cloud Infrastructure Data Center Regions for more details.

Use TNS Alias in the WebLogic's datasources and JPS files

In a Disaster Recovery topology, you can use three approaches for configuring the database connection string in the WebLogic datasources:

Copyright © 2024, Oracle and/or its affiliates | Public

- Use a dataguard ready (also known as "dual") jdbc string. In this case, the database connect string includes both primary's and standby's database connect addresses, but only the database that has the primary role provides the service. However, this approach is recommended only for DR environments that are based on stretched clusters, where the standby database is in the same region than primary, and the midtier can connect to primary or standby database. This is not the case of this document.
- Use a non-dual jdbc string, different in each site, pointing to the local database only. In this case, a replacement is required everytime that the WebLogic domain configuration is copied from primary to standby. This approach was used in this document before document version 16.
- Use a Transparent Network Substrate **(TNS) alias in the datasources**. The TNS alias name is the same in primary and secondary, so the datasources have the same db connect string in both sites. The TNS alias is resolved with a tnsnames.ora file that is not replicated between sites. Hence, you can have a different tnsnames.ora content in each site, but same WebLogic configuration. Each site will resolve the TNS alias with the appropriate connect string, pointing to the local database only. No replacement is needed when the WebLogic domain configuration is copied from primary to standby. **This is the recommended approach in remote DR scenarios.**

You can find more details in the section “Setting Up DataSources in the Middle Tier” of the FMW Disaster Recovery Guide.

This document uses the **TNS alias approach**. The provided scripts are designed to work in environments that use TNS alias in the WebLogic Datasources and JPS config files. If your WebLogic system is not already using the TNS alias approach, you can configure it by following the steps provided later in this document.

Regions with Block Volume Cross-Region Replication for BV Replica Method

The primary and standby regions used in the DR topology must be different, and the Block Volume Cross-Region Replication **must be available** between them. Not all the regions are interconnected for Block Volume Cross-Region replica. The source region for the volume to replicate determines the target regions available to select as destination region. Check the table that lists the source region and target regions available for volume replication in Oracle Cloud documentation link Cross-Region Replication > Source and Destination region Mappings.

Staging filesystems for the WebLogic domain config replication for DBFS and FSS with rsync Methods

Two different methods that use a a stage directory to replicate the WebLogic domain configuration between sites are included in this document: DBFS, and FSS with rsync. Both methods use an assistance filesystem, DBFS or FSS respectively, that is mounted in the WebLogic hosts. In the next sections, this document provides specific instructions to configure the staging filesystem in each case.
The method based in Block Volume cross-region replication is the only one that does not require staging file system.

Custom files in DBFS and FSS with rsync Methods

The WebLogic Server domain configuration is synced initially across sites with the following considerations:

- Each WebLogic system will **connect to its local DB** after the DR set up has completed, pointing to the same schemas (primary schemas). The **tns admin folder** that contains the tnsnames.ora is **excluded from the copy**.
- During the initial sync across sites, the content of the WebLogic domain folder **of the first primary node** is copied **to all the secondary nodes**.
- Custom application deployments (ear/war files, deployment plans, etc.) and everything that resides **under the Administration Server WLS domain directory** (except temp data) is synced across sites with the procedures described in this document.
- All the **configuration under weblogic_domain_name/config** is automatically distributed to the other nodes in the same site by the WebLogic cluster features: when a managed server starts, it retrieves the configuration from the Administration Server.
- **Posterior updates** of artifacts **in other nodes** of the domain (except in the Admin Server’s) **outside** the **weblogic_domain_name/config** directory **are not replicated** by neither the WebLogic cluster features nor the procedures in this document.
- Oracle recommends using **central stores for keys and identities** irrespectively of DR. For example, in **non JRF-enabled domains** (e.g., WLS version 14.1.1), the node manager does not use Keystore Service (KSS) as the identity keystore. It uses the $DOMAIN_HOME/security/DemoIdentity.jks out-of-the-box. This file is different in each host. To avoid "*Hostname verification failed*" errors when connecting to the node managers in secondary, you will need to copy the DemoIdentity.jks file from primary node wls-1 to secondary node wls-1, from primary wls-2 to

Copyright © 2024, Oracle and/or its affiliates | Public

secondary wls-2, etc. after the DR setup. The same applies if you are not using KSS and you have customized the JKS identity keystore files. It is for this that it is recommended to use a central location for these files. Otherwise, if you store them in a folder that is not automatically distributed nor shared, manual intervention is required (you need to copy them to other nodes). If these file change and are not replicated error will arise in different scenarios, like listen address changes, DR, and others.

The next sections of this document provide more details about how the configuration replica is performed. In case that you have any other data that resides in other nodes or outside the domain directory of the WebLogic Administration Server, **you will have to manually copy it to the secondary location**.

SLA requirements

WLS for OCI is a user-managed environment. The user must determine service level expectations for availability, data protection, and performance that are practical for a given configuration and application. Service Levels must be established for each of three dimensions relevant to disaster recovery that are applicable to any Data Guard configuration:

- **Availability**: Recovery Time Objective (RTO) describes the maximum acceptable downtime should an outage occur. This includes time required to detect the outage and to failover the database, the Web tier and WebLogic servers so that service is resumed. More details about this in the section RTO and RPO Overview of this document.
- **Data Protection**: Recovery Point Objective (RPO) describes the maximum amount of data loss that can be tolerated. The actual achievable RPO depends upon:
  - Available network bandwidth.
  - Network reliability.
  - Data Guard transport method used: either *asynchronous* for near-zero data loss protection, or *synchronous* for zero data loss protection.

  More details about this in the section RTO and RPO Overview of this document.

- **Performance**: Database and Middle Tier response time may be different after failover if less capacity – compute, memory, I/O, etc., are provisioned at the standby system than in the primary system. This occurs when users purposefully under-configure standby resources to reduce cost (accepting reduced service level while in DR model). MAA best practices recommend **configuring symmetrical capacity at both primary and standby** in the web, application, and database tiers so there is no change in response time after failover. Rapid provisioning available with the cloud can enable a middle ground where less capacity is initially deployed, but where the new primary is rapidly scaled-up should a failover be required.

In addition, the status of the hosts and services in the secondary site can impact on the RTO and RPO. As explained in previous sections, if the standby database is in shutdown status during normal business operation, it will not receive updates from primary and it will become out-of-sync. This can result in a data loss (impact on RPO) in case a switchover needs to be performed, thus it is not recommended to have the standby database stopped during normal business operation.

The standby midtier hosts can be stopped. However, the configuration changes that are replicated from the primary site will not be pushed to the secondary domain hosts while they are stopped. In case of a switchover event, the RTO is increased because the midtier hosts need to be started and synchronized with primary. Thus, it is recommended to have the secondary midtier hosts up (with WebLogic servers stopped). See About having compute instances stopped in standby site for more details.

Note: Independent of the service levels related to DR, all database instances created in the Oracle cloud conform to the service descriptions defined by the applicable Database Cloud Service[4].

[4] http://www.oracle.com/us/corporate/contracts/paas-iaas-public-cloud-2140609.pdf

Copyright © 2024, Oracle and/or its affiliates | Public

DOWNLOAD SCRIPTS

The scripts you need for the setup and lifecycle of your WLS for OCI DR are available in the **MAA GitHub repository**.

a) Go to the MAA repository in GitHub https://github.com/oracle-samples/maa
b) Download all the scripts in the **wls_mp_dr** directory.
c) Download all the scripts in the **app_dr_common** directory.
d) The scripts make calls to each other. Despite the specific operation being performed at a point in time, download the entire directories and **place all the scripts of both directories in the same folder**. You will need the scripts in both the primary and secondary sites.

Do not run any script now. Follow the instructions in this document and and run the scripts when required.

NOTE: You can use a tool like https://download-directory.github.io/ to download a particular folder only.

Copyright © 2024, Oracle and/or its affiliates | Public

DISASTER RECOVERY SETUP OVERVIEW

As starting point, this document assumes that a primary site already exists and is “live”. The secondary DR configuration, that resides in a geographically remote site, will be created based on this existing primary system. Since the primary system may already be running in production, the DR configuration process is designed to cause minimum downtime (only the modification of the front-end address requires WebLogic server restarts).

As described in the Topology Description point, depending on how the WebLogic domain configuration is replicated to the secondary site, there are three different DR models: **DBFS, FSS with RSYNC, and Block Volume Cross-Region replica**. The following flow chart describes the steps of the Disaster Recovery setup process for the three models described in this document:

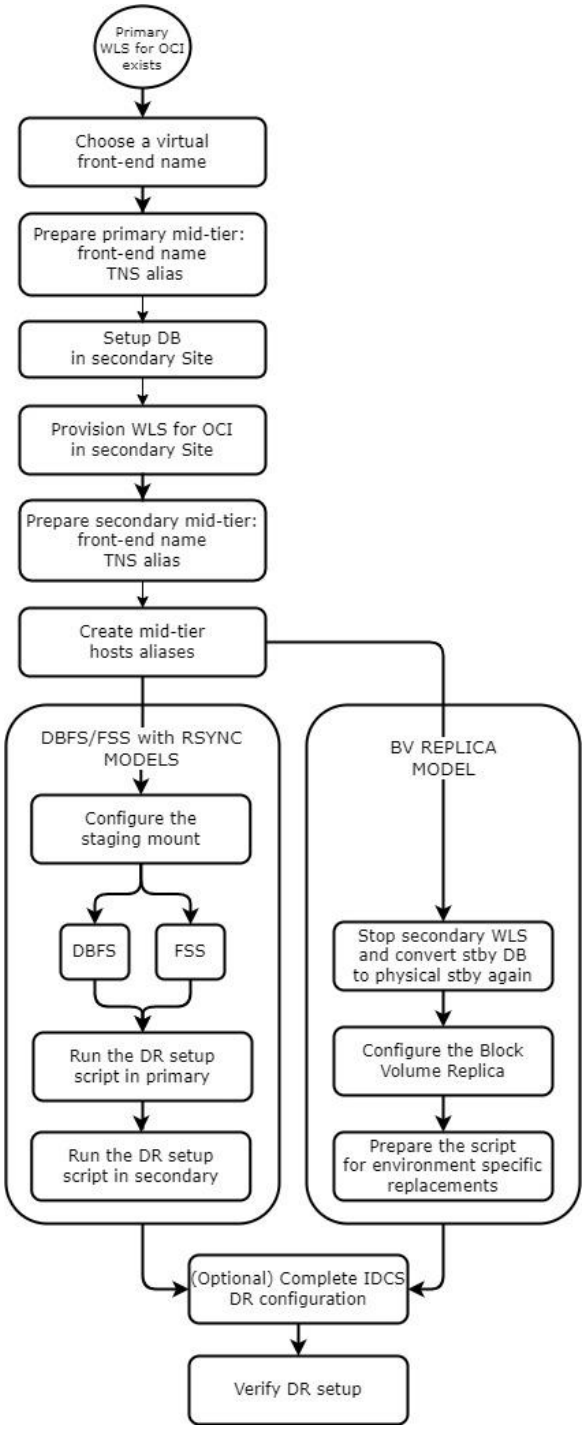


Figure 7 Flow chart of the DR setup steps for the models described in this document.

Copyright © 2024, Oracle and/or its affiliates | Public

PREPARE FOR DISASTER RECOVERY SETUP

NOTE: these steps apply to all the replication methods (DBFS, FSS with rsync and Block Volume cross-region replica)

1. Choose a virtual front-end name

When you create a WLS for OCI stack, the provisioned Load Balancer listens on a specific front-end IP. A Fully Qualified Domain Name (FQDN) front-end is not provided nor configured in the system. The primary site's LBR listens on one front-end IP address, and second site's LBR will listen on another front-end IP address. Each WLS cluster "Frontend Host" property is provisioned by default with the corresponding Load Balancer front-end IP.

In a disaster recovery topology, the clients must access the system using a URL with a front-end FQDN that is **agnostic** to the "Cloud region or data center", usually referred to as "**virtual" front-end name** or "vanity url". This virtual front-end name should resolve to the LBR IP address for the current active primary site. You must **choose a virtual front-end name** for the system within a DNS domain (for example, "mywebapp.mycompany.com") and **make it resolvable externally**. If you already have a virtual front-end name configured to access to the primary system, you can reuse it for the DR configuration.

To externally resolve this virtual front-end name, you must register it any formal public DNS (alternatively, you can add it to the client's local hosts file). To resolve the virtual front-end name in the scope of the WebLogic hosts locally, the system's hosts file should be manually configured, as explained in next points.

To determine the public IP address of the LBRs in your system, login into the OCI Console, select the correct region and compartment, navigate to Load Balancers section, click on your LBR, and look for the public IP address that the LBR listens on.

2. Prepare Primary mid-tier for the virtual front-end

Perform these actions in the primary mid-tier to prepare it for the DR configuration.

a) **Add the virtual front-end name and IP to the /etc/hosts file in all primary mid-tier hosts.**
Each mid-tier host should always resolve the front-end name to it's local LBR regardless of client-facing resolution via DNS. With root user, edit the /etc/hosts file and map the primary LBR public IP to the virtual front-end FQDN. Repeat in all primary mid-tier hosts. Example:

```
[oracle@wlsociprefix-wls-0 ~]$ more /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
…

# Front-end virtual name
111.111.111.111 mywebapp.mycompany.com
```

NOTE: the /etc/hosts file of the primary mid-tier hosts must not be altered when there is a switchover or failover. Primary hosts will always resolve the virtual front-end name with its front-end IP. The dns update that is needed during the switchover and failover procedures is performed in the DNS or host files used by the clients.

b) **Configure the front-end name as cluster front-end**.
Login in the WebLogic Console of your instance and:

- Navigate to Environment > Clusters and select the cluster.
- Go to Configuration > HTTP.
- Set the Fronted host to the virtual front-end FQDN (example "mywebapp.mycompany.com").
- Be sure that the Frontend ports for HTTP and HTTPS are correctly configured with values.
- Save and activate.
- A cluster restart is required for this change to be effective.

Copyright © 2024, Oracle and/or its affiliates | Public

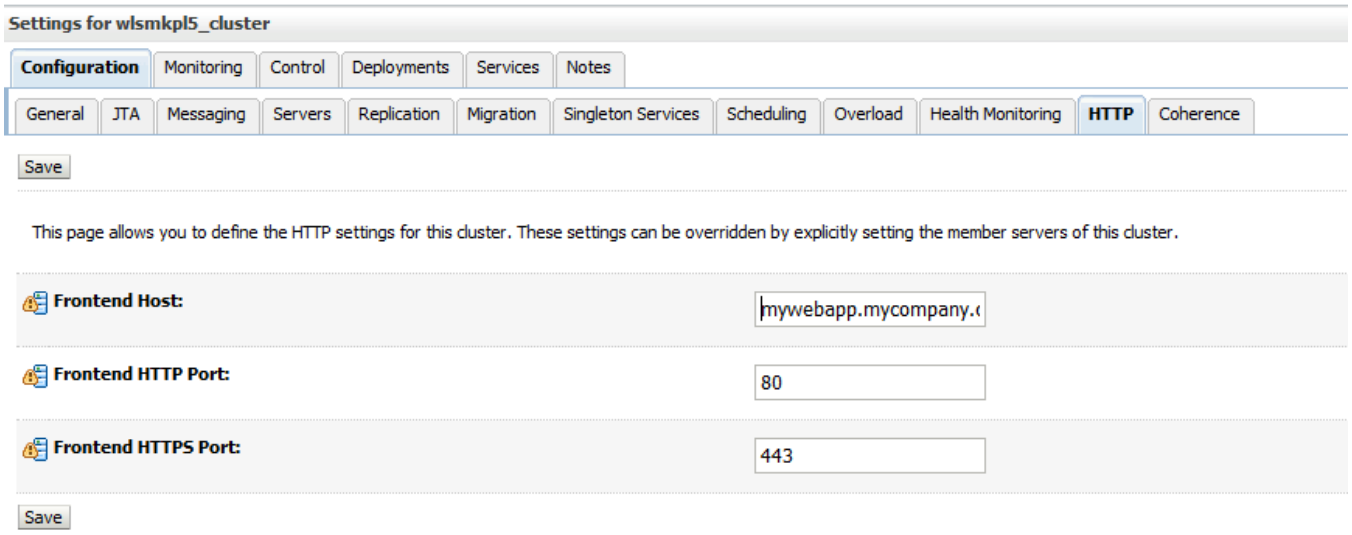


Figure 8 Cluster front-end host configuration

c) **Update t3/rmi urls (if used) with cluster syntax**
The urls used for RMI invocations in the WebLogic cluster need to be agnostic to the IPs or hostnames used by each site. Instead of using the host:port,host:portJNDI urls syntax, change them to use the cluster syntax. The cluster syntax is as follows: cluster:t3://cluster_name.[5]
Similarly, any other custom JNDI urls used in the system should also be updated so that when a switchover/failover occurs, the urls are valid also in the secondary site.

3. Prepare primary mid-tier for using TNS alias

Using a TNS alias in JDBC URLs facilitates the replica of the WebLogic configuration from primary to standby. If your system is not already using this approach in the datasources and JPS configuration files, follow these steps to configure it.

a) **If you are using RAC database**, use the script **fmw_change_to_tns_alias.sh** to perform the change automatically. The script assumes that the datasources are using the long connection string format (like it is expected in the GridLink datasources). This script takes the existing TNS string from the datasource, creates a tns_admin folder containing a tnsnames.ora, and writes the TNS entry into the tnsnames.ora. It replaces the connect string with an alias in each datasource and jps config file. It also adds the tns_admin property to each of these files, pointing to the tns_admin folder.

b) **Otherwise, if you are using single instance database**, perform the configuration manually as follows:

- **Create a tns folder in the mid-tier hosts.**
  Create the folder $DOMAIN_HOME/config/tnsadmin
  This folder will be excluded from the copy of the configuration that is performed from primary to secondary.
- **Create a tnsnames.ora file in the tns folder.**
  This file must contain a TNS entry for the database used in the datasources. Example:

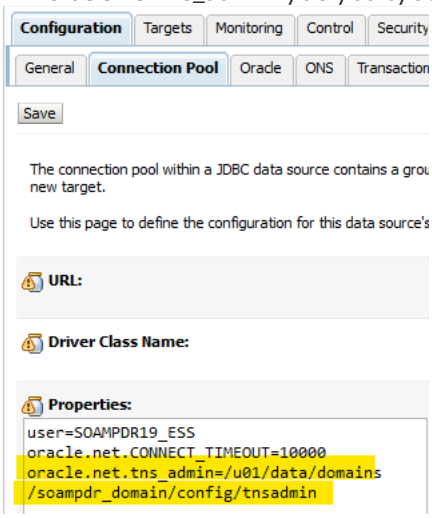
```
PDB1 = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.
dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=PDB1.dbsubnet.vcnash226.oraclev
cn.com)))
```

- **Use WLS Administration Console to set the tns alias in the URL of the datasources.**
  Two modifications are required in each datasource:

[5] Using the cluster name syntax in t3/RMI URLs is feasible only for intra-domain invocations. T3/rmi clients that are external to the domain will not be able to use this approach and will have to use the appropriate DNS mapping of the host:port list when switching to the secondary site. A TCP load balancer can be used for the JNDI InitialContext retrieval, but subsequent requests from JMS clients will connect to the host:port directly, so the DNS mapping to secondary site IPs is required also in this case.

Copyright © 2024, Oracle and/or its affiliates | Public

- Add the property “oracle.net.tns_admin” in each datasource > Configuration > Connection Pool > Properties. Example:
  oracle.net.tns_admin=/u01/data/domains/wlsociprefix_domain/config/tnsadmin



  - Modify the URL to this format: jdbc:oracle:thin:@tnsalias. Example:
    jdbc:oracle:thin:@PDB1
  - Save and activate.
- **Edit the JPS config files to set the tns alias.**
  You cannot use the WLS Administration Console to modify the JPS config files.
  - Edit the jps-config.xml and jps-config-jse.xml files in $DOMAIN_HOME/config/fmwconfig
  - Look for the *<property name="jdbc.url"*
  - Modify the property value to the alias format connect string. Example:
    <property name="jdbc.url" value="jdbc:oracle:thin:@PDB1"/>
  - Before this line, add the property oracle.net.tns_admin pointing to the tns admin folder. Example:
    <property name="oracle.net.tns_admin" value="/u01/data/domains/ wlsociprefix_domain/config/tnsadmin"/>

c) If you are using more than one database in your datasources, make sure **all of them have the appropriate alias** in the datasource **and the tns entry** is in the tnsnames.ora file.

d) **Restart the servers** so the changes are effective:
- Stop all the servers (admin and managed).
- Start the admin server first.
- Start the managed servers.

4. Setup the Database in Secondary Site

Create the database in the secondary site as a **Data Guard physical standby of the primary database**. There are two options for doing this: one is to use the OCI Console to enable Data Guard (referred in this document as “automated Data Guard”), and the other option is to manually create and configure the standby database with dgmgrl commands (referred in this document as “manual Data Guard”).

The **recommended** approach **is to configure the Data Guard with the OCI Console (option 1)**. This way, it is integrated with the OCI Console User Interface, and you can use the Console to manage Oracle Data Guard. It also provides out of the box configuration for backups in the Data Guard. Follow the point Option 1) Configuring Data Guard using OCI Console to enable the Data Guard using the OCI Console.

If for any reason the feature to enable Data Guard is not available for your case (refer to the DB System documentation to check the availability of the Data Guard across regions feature in each DB Systems flavor/edition), you can still configure the Data Guard manually using scripts provided in this document. Follow steps described inOption 2) Configuring Data Guard manually for this.

When using **database** in **Oracle Exadata Database Service**, setup secondary database as described in the Oracle Exadata Database Service documentation Using Oracle Data Guard with Exadata Cloud Infrastructure and then continue in the point 4.4 Considerations for EXACS.

Copyright © 2024, Oracle and/or its affiliates | Public

4.1 Option 1) Configuring Data Guard using OCI Console

When you enable Data Guard with the OCI Console**, the secondary DB system is automatically provisioned and configured as physical standby** when you click on **Enable Data Guard** in the primary DB System. There are some requirements for this, for example: both DB systems will be in the same compartment, both DB Systems will be the same shape type, if the DB Systems will be in different regions, they must be connected via remote VCN peering, etc. See Using Oracle Data Guard in Oracle Cloud Infrastructure Documentation for more details about these requirements.

To enable Data Guard to primary database, login to OCI Console, navigate to the primary DB System, and click in the primary database. Enable Data Guard in the section "Data Guard Associations". Most of the configuration properties of the secondary DB System (like version, DB name, etc.) are predefined because they are inherited from primary, but you need to provide some configuration properties. The following table provides examples and requirements for these properties:

DB SYSTEM CONFIGURATION PROPERTY	EXISTING PRIMARY DB SYSTEM / EXAMPLE	SECONDARY DB SYSTEM / EXAMPLE	REQUIREMENT FOR AUTOMATED DG
Oracle Cloud Tenancy	XXXX / paasmaa	YYYY / paasmaa	must be the same
Compartment	XXXX / wlsdr	XXXX / wlsdr	must be the same
Region	XXXX / Ashburn	YYYY / Phoenix	must be different (recommended different regions for DR)
Availability Domain	XXXX / efEXT:US-ASBURN-AD1	YYYY / efXT:PHX-AD-1	must be different
DB System Name	XXXX / drdba	YYYY / drdbb	must be different
Shape	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	must be the same
Virtual Cloud network	XXXX / wlsdrvcn1ash	YYYY / wlsdrvcn1pho	must be different (expected different regions, connected via remote VCN peering)
Client subnet	XXXX / ashsubnet1	XXXX / phosubnet1	must be different (expected different regions, connected via remote VCN peering)
Hostname Prefix	XXXX / drdba	YYYY / drdbb	must be different
Administrator password	XXXX / password	XXXX / password	must be the same To perform backups from the standby database, check Backup and Restore from a Standby Database in a Data Guard Association

4.2 Option 2) Configuring Data Guard manually

Use this approach only when it is not possible to use the same cloud tenancy for primary and standby or when the enable Data Guard option provided by OCI Console is not available for the DB flavor and/or locations involved in the DR configuration. In this case, you must provision the secondary database as a regular DB System, and then, manually configure the Data Guard. For this manual configuration, you can use the Data Guard setup scripts provided in this document, as explained in these steps:

Copyright © 2024, Oracle and/or its affiliates | Public

4.2.1 Provisioning Secondary Database

Note: In case that the Data Guard has been enabled using the OCI Console, these steps must be skipped and you can continue with 4.3 Configuring a custom PDB service in the DB Systems

When configuring the Data Guard manually, you first need to provision the secondary database, using the same Database name, PDB name, release, patch level, number of nodes and edition used in primary. This may require patching the primary system (especially if it has been running for a long time) before creating the standby. Oracle recommends using the same Compute Shape and Storage Size that are used for primary. Follow the steps in the Cloud DB System documentation to provision the required Database System for the standby datacenter.

The following table provides examples and requirements for the properties that need to be used in the standby DB System creation process:

DB SYSTEM CONFIGURATION PROPERTY	EXISTING PRIMARY DB SYSTEM / EXAMPLE VALUE	SECONDARY DB SYSTEM / EXAMPLE VALUE	REQUIREMENT FOR MANUAL DG
Oracle Cloud Tenancy	XXXX / paasmaa	YYYY / paasmaa	can be different
Compartment	XXXX / wlsdr	YYYY / wlsdr	can be different
Region	XXXX / Ashburn	YYYY / Phoenix	must be different (recommended different regions for DR)
Availability Domain	XXXX / efEXT:US-ASBURN-AD1	YYYY / efXT:PHX-AD-1	must be different
DB System Name	XXXX / drdba	YYYY / drdbb	must be different
Shape	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	must be the same
Total node count	N / 1	N / 1	must be the same
Oracle Database Software edition	EE, EE-HP or EE-EP / EE-EP	EE, EE-HP or EE-EP / EE-EP	must be the same
Available storage	XXXX / 256	XXXX / 256	must be the same
License type	LI, BYOL / BYOL	LI, BYOL / BYOL	can be different
SSH public key	XXXX	YYYY	must be the same
Virtual Cloud network	XXXX / wlsdrvcn1ash	YYYY / wlsdrvcn1pho	must be different (expected different regions, recommended connect via remote VCN peering)
Client subnet	XXXX / ashsubnet1	XXXX / wlsdrvcn1pho	must be different
Hostname Prefix	XXXX / drdba	YYYY / drdbb	must be different
Database Name	XXXX / ORCL	XXXX / ORCL	must be the same
Database Version	XXXX / 19c	XXXX / 19c	must be the same

Copyright © 2024, Oracle and/or its affiliates | Public

PDB name	XXXX / PDB1	XXXX / PDB1	must be the same
Administrator password	XXXX / password	XXXX / password	must be the same
Enable automatic backups	X / Checked	Y / unchecked	must be disabled in stby To perform backups from the standby database, check Backup and Restore from a Standby Database in a Data Guard Association

NOTE: The default database instance created on the secondary site will be deleted later as it cannot be used as a Data Guard standby database. It is created with the same name as primary to get the required lifecycle scripts seeded in the system with the same configuration as the primary DB

Make sure to apply the required patches to the DB in both locations (primary and secondary) so that to both are at the same patch level. More precisely, a Data Guard configuration requires a fix for bug 22611167 in 12c versions. Verify if the patch is applied in both the primary and secondary DB systems and apply it in case it is not. Latest OCI 12cR2 DB systems have the patch for this bug pre-installed.

4.2.2 Configuring Data Guard between primary and secondary

Note: In case that the Data Guard has been enabled using the OCI Console, skip these steps and continue with 4.3 Configuring a custom PDB service in the DB Systems

To configure the Data Guard manually between the primary and secondary databases, follow these steps.

a) The primary and standby databases in a Data Guard need to communicate each other on the listener's port. It is also needed that each database can reach its own IP on the appropriate listener port. Make sure that the appropriate ingress rules are defined in each VCN (primary and standby) to allow these connections.
For RAC databases, it is a **requirement** that primary and standby RAC communicate **via Dynamic Routing Gateway,** because the scan and VIP IP addresses must be reachable from one site to the other.

Verify communication using nc command (use the public/private IPs depending on your network topology). For example:

```
[opc@drdDBa ~]$ nc -vw 5 -z <secondary_db_ip> 1521
```

Example for correct output:

```
[opc@drdbb ~]$ nc -vw 5 -z 10.0.0.79 1521
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 10.0.0.79:1521.
Ncat: 0 bytes sent, 0 bytes received in 0.07 seconds.
```

NOTE: Use the public DB System's IPs only in case that primary and secondary sites use Internet Gateway to communicate each other. Use the internal DB System's IPs in case the communication between primary and secondary VCNs is done internally, using a Dynamic Routing Gateway (recommended approach).

Copyright © 2024, Oracle and/or its affiliates | Public

b) Download the set of scripts for manual Data Guard configuration[6], and follow the instructions described in the in the solution playbook Configure a standby database for disaster recovery. This set of scripts is valid to configure a Data Guard in single instance databases and in RAC scenarios.

c) After the DG setup is complete, enter the Data Guard Broker CLI from the primary system to check the configuration (redo apply may take some time to catch up):

```
DGMGRL> show configuration
Configuration - ORCL_lhr2bb_ORCL_fra22g
  Protection Mode: MaxPerformance
  Members:
  ORCL_lhr2bb - Primary database
  ORCL_fra22g – Physical  standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 33 seconds ago)
```

4.3 Configuring a custom PDB service in the DB Systems

When the DB system is a Real Application Cluster (RAC) Database, **Oracle WebLogic Server Suite** on OCI uses GridLink data sources to connect to the selected Oracle Database. GridLink provides dynamic load balancing and failover across the nodes in an Oracle Database RAC and receives notifications from the database when nodes are added or removed from the cluster. To fully take advantage of these capabilities, Oracle recommends that you create an Oracle Database service that supports Cluster Ready Services (CRS) and the Oracle Notification Service (ONS). These services monitor the status of resources in the database cluster and generate notifications when a status changes. Check About the WebLogic on OCI Edition when using RAC in Appendix A – Considerations for RAC databases for more details about supported WebLogic for OCI Edition with RAC.

Oracle Base Database services Systems don't create any CRS application services for the PDB out-of-the-box. There is only the PDB default service, with a name like *<pdbname>.<subnetdomain>.<vcndomain>.oraclevcn.com*. This is not a CSR managed service. To use a CRS service to connect to the PDB, you need to create a custom service manually. You need to create it in the primary and standby databases.

a) **Create a custom Oracle Database service for the PDB in primary.**
Connect to a db node of the primary RAC database and run the following with the user oracle to add, configure and start a service:

```
srvctl add service -db <PRIM_DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -preferred
<INSTANCE_NAME1>,<INSTANCE_NAME2> -pdb <PDB_NAME>  -role "PRIMARY,SNAPSHOT_STANDBY"

srvctl modify service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb
<PDB_NAME>

srvctl start service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME>
```

Note that it is important to provide the 2 roles to the "-role" parameter when creating the service, so the service is automatically started when the database is in primary or snapshot standby.

Example:

[6] https://github.com/oracle-samples/maa/raw/main/dg_setup_scripts/dg_setup_scripts.zip

Copyright © 2024, Oracle and/or its affiliates | Public

```
[oracle@priracnode1 ~]$ srvctl add service -db ORCL_lhr3jg -service mypdbservice.example.com  -preferred ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY "
[oracle priracnode1 ~]$ srvctl modify service -db ORCL_lhr3jg -service mypdbservice.example.com  -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb pdb1
[oracle@priracnode1 ~]$ srvctl start service -db ORCL_lhr3jg -service mypdbservice.example.com
[oracle@priracnode1 ~]$ srvctl config service -db ORCL_lhr3jg -service mypdbservice.example.com
Service name: mypdbservice.example.com
Server pool:
Cardinality: 2
Service role: PRIMARY,SNAPSHOT_STANDBY
Management policy: AUTOMATIC
…

Available instances:
CSS critical: no
```

b) Create the same Oracle Database service in the secondary RAC database.
Connect to a db node of the secondary RAC database and run the following with the user oracle to create and configure the service:

```
srvctl add service -db <SECONDARY_DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -preferred <INSTANCE_NAME1>,<INSTANCE_NAME2> -pdb <PDB_NAME>  -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb <PDB_NAME>
```

Example:

```
[oracle@secracnode ~]$ srvctl add service -db ORCL_fra3vb -service mydbservice.example.com  -preferred ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY"
[oracle@secracnode1~]$ srvctl config service -db ORCL_fra3vb -service mypdbservice.example.com
Service name: mypdbservice.example.com
Server pool:
Cardinality: 2
Service role: PRIMARY,SNAPSHOT_STANDBY
Management policy: AUTOMATIC
DTP transaction: false
…
```

c) **List the new PDB services.**
Connect to a db node of the primary RAC and run the following with the user oracle:

```
[oracle@priracnode1 ~]$srvctl status service -db $ORACLE_UNQNAME
Service mypdbservice.example.com is running on instance(s) ORCL1,ORCL2
```

The service must be running in the primary RAC.

Connect to a db node of the secondary RAC and run the following with the user oracle:

```
[oracle@secracnode ~]$srvctl status service -db $ORACLE_UNQNAME
Service mypdbservice.example.com is not running.
```

The service in the standby RAC can be stopped.

d) **Update the connection values in the primary WLS for OCI.**
Make sure that the primary WLS for OCI system is connecting to the CRS service. The entry in the $DOMAIN/config/tnsadmin/tnsnames.ora used by the datasources and the jps config files must point to the CRS service instead of to the default pdb service. Example:

```
mypdb = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=mypdbservice.example.com)))
```


Copyright © 2024, Oracle and/or its affiliates | Public

4.4 Considerations for EXACS

When you use Oracle Exadata Database Service, follow these recommendations:

- Setup Data Guard as described in the Oracle Exadata Database Service documentation: Using Oracle Data Guard with Exadata Cloud Infrastructure.
- The databases in Oracle Exadata Database Service use CRS database services out-of-the-box. There are some services already configured for the CDB and for the PDB, which you can list with the srvctl command. The primary database has a PDB service pre-configured, with name <DBNAME_PDBNAME>.paas.oracle.com. Example:

```
[oracle@primary-exadb-host1 ~]$ srvctl status service -db $ORACLE_UNQNAME
Service exadb_dg is running on instance(s) EXADB1,EXADB2
Service exadb_dg_ro is not running.
Service exadb_pdb1.paas.oracle.com is running on instance(s) EXADB1,EXADB2  → this is primary PDB service
Service pdb1_dg is running on instance(s) EXADB1,EXADB2
Service pdb1_dg_ro is not running.
```

  However, this service is not configured in the standby database. By default, the standby database has only the "dg" and "read only" services only. Example:

```
[oracle@standby-exadb-host1 ~]$ srvctl status service -db $ORACLE_UNQNAME
Service exadb_dg is not running.
Service exadb_dg_ro is not running.
Service pdb1_dg is not running.
Service pdb1_dg_ro is not running.
```

  Follow these steps to get the appropriate service configuration in Exadata Data Guard for WLS for OCI DR:

  - **Create the PDB service in the standby database**. For consistency, use the same PDB service name that is used by the primary WLS to connect to the primary PDB. Example:

```
[oracle@standby-exadb-host1 ~]$ srvctl add service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -preferred EXADB_PHO1,EXADB_PHO2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"

[oracle@standby-exadb-host1 ~]$ srvctl modify service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb PDB1

[oracle@standby-exadb-host1 ~]$ srvctl start service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com
```

    Note that it is important to provide the 2 roles to the "-role" parameter when you create the service, so the service automatically starts when the database is in primary or snapshot standby role (which is needed for the setup and for lifecycle operations).

  - Also, **modify the PDB service in primary** to add the standby roles too, because it is normally configured with the "primary" role only. Example:

```
[oracle@primary-exadb-host1 ~]$ srvctl modify service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -role "PRIMARY,SNAPSHOT_STANDBY"
```

  - **Update the connection values in the primary WLS for OCI.**
    Make sure that the primary WLS for OCI system is connecting to the CRS service. The entry in the $DOMAIN/config/tnsadmin/tnsnames.ora used by the datasources and the jps config files must point to the CRS service instead of to the default pdb service. Example:

```
mypdb = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=mypdbservice.example.com)))
```

5. Provision WLS for OCI in Secondary Site

The secondary WebLogic for OCI stack will be created **pointing to the secondary DB system**, which must be **open in snapshot standby mode**.

So, before you provision the secondary WebLogic for OCI, **convert the standby database to snapshot standby**. This will make the standby database to stop applying changes from primary and be opened in read-write, to allow the secondary WebLogic creation. To do this, execute the following as oracle user in the **primary** DB host:

Copyright © 2024, Oracle and/or its affiliates | Public

```
[oracle@drdba ~]$dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> CONVERT DATABASE “secondary_db_unqname” to SNAPSHOT STANDBY;
Converting database "secondary_db_unqname" to a Snapshot Standby database, please wait...
Database "secondary_db_unqname" converted successfully
```

Then, follow the steps in the Using Oracle WebLogic Server for OCI documentation to create the secondary site WebLogic Cloud system pointing to the secondary DB System that was converted to snapshot in the previous step.

The Stack Name can be different, but you must use the **EXACT same resource name prefix** that you used in your primary location. Oracle recommends that the exact same capacity and compute configuration is used on both primary and standby locations for the ideal failover and switchover behavior.

Make sure that the WLS for OCI version and patch level provisioned in the secondary location matches the one running in the primary site. WLS for OCI provisioning offers the two latest PSUs, which means that the customer can provision the same PSU in timeframe 3-6 months. If, at the moment of the secondary provisioning, the PSU of the primary WLS for OCI is not available in provisioning, the primary WLS for OCI must be patched to the same level as the newly provisioned secondary site. Check https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/patches.html to see the patches applied in each WLS for OCI release.

The following table summarizes the provisioning wizard options for the set up:

WLS FOR OCI PROPERTY	VALUE IN PRIMARY / EXAMPLE	VALUE IN SECONDARY / EXAMPLE	REQUIREMENT FOR DR
Region	XXXX /Ashburn	YYYY / Phoenix	Must be different
Version	XXXX / 12.2.1.4.200414.03	XXXX / 12.2.1.4.200414.03	Must be the same**Error! Bookmark not d efined.**
Stack Name	XXXX / wlsociprefixprim	YYYY / wlsociprefixstby	Can be different
Resource Name Prefix	XXXX / wlsociprefix	XXXX / wlsociprefix	Must be the same
SSH Public Key	XXXX / my_public_key.pub	XXXX / my_public_key.pub	Must be the same
OCI Policies	Check or Unchecked	Unchecked	Policies must exist for primary and be aplicable to standby. See **POLICIES NOTE** after this table.
Provision Load Balancer	Must be checked	Must be checked	Checked in both cases
Enable Authentication Using Identity Cloud Service	Check or Unchecked	Check or Unchecked	Must be the same
Network Compartment	XXXX / wlsdr	YYYY / wlsdr	Can be different (but normally the same)
VCN	XXXX / wlsdrvcn1ash	YYYY / wlsdrvcn1pho	Must be different
Subnet	XXXX / ashsubnet1	YYYY / phosubnet1	Must be different
WebLogic Server Shape	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	Must be the same
Admin User Name	XXXX / weblogic	XXXX / weblogic	Must be the same
Admin Password (secret)	XXXX / secretadmin_a	YYYY / secretadmin_b	Must be the same password (although the secret could be

Copyright © 2024, Oracle and/or its affiliates | Public

			different because the secondary vault will be in the secondary region)
Provision with JRF	XXXX / Checked	XXXX / Checked	Must be the same
Database Strategy	Database System	Database System	Must be the same
DB system compartment	XXXX / wlsdr	XXXX / wlsdr	Can be different (but normally the same)
DB system network	XXXX / wlsdrvcn1ash	YYYY / wlsdrvcn1pho	Must be different (expected different regions)
DB system	XXXX / drdba	YYYY / drdbb	Must be different
DB version	XXXX / 19	XXXX / 19	Must be the same
Database in the system	XXXX / ORCL	XXXX / ORCL	Must be the same
PDB	XXXX / PDB1	XXXX / PDB1	Must be the same
Database administrator	SYS	SYS	Must be the same
Database administrator password (secret)	XXXX/ secretdb_a	XXXX / secretdb_b	Must be the same password (although the secret could be different because the vault will be in the secondary region). The Vault service is used only for encrypting and decrypting passwords during the provisioning. It is not used for WebLogic runtime or lifecycle tasks.
Deploy Sample Application	XXXX / Checked	XXXX / Checked	Must be the same
Configure Ports	XXXX	XXXX	Must be the same values. In case you customize the ports, ensure that you use the same ports than in primary.
Compute Shape	XXXX	XXXX	Must be the same
WebLogic Server Node Count	N / 2	N / 2	Must be the same
(if IDCS is checked) **IDCS host domain name**	XXXX/ identity.oraclecloud.com	XXXX/ identity.oraclecloud.com	Must be the same
(if IDCS is checked) IDCS port	XXX / 443	XXX / 443	Must be the same
(if IDCS is checked) IDCS tenant	XXXX/ idcs-5e890de5903jr…	XXXX/ idcs-5e890de5903jr…	Must be the same

Copyright © 2024, Oracle and/or its affiliates | Public

(if IDCS is checked) IDCS Client ID	XXXX / f88aa64hh8hgd…	XXXX / f88aa64hh8hgd…	Must be the same
(if IDCS is checked) **Secrets OCID for IDCS Client Secret**	XXXX / secret1	YYYY / secret2	IDCS Client Secret (unencrypted value) must be the same, but the secret can be different (created in each region's vault)
(if IDCS is checked) **IDCS Redirect Port**	XXXX / 9999	XXXX / 9999	Must be the same

POLICIES NOTE: When the flag "OCI Policies" is checked, the provisioning job automatically creates the OCI Policies and Dynamic Group needed for creating the WLS for OCI resources. The names of the OCI Policy and Dynamic Group are derived from the instance prefix. If you checked this flag when you provisioned primary, and you check it again when you provision the secondary instance, the provisioning job tries to create OCI Policies and Dynamic Group with the same names than the existing. This causes a naming conflict and the provisioning of the secondary fails. Follow these guidelines to avoid this problem:

- If you didn't check the "OCI Policies" flag when you provisioned primary, you must have created the OCI Policies and Dynamic Group manually, as described in https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/you-begin-oracle-weblogic-cloud.html. In this case:
    - Make sure that the Dynamic Group includes the resources of the secondary region. E.g.: if you created a Dynamic Group for all the resources in the compartment, and the secondary compartment is the same, hence the resourcesof the secondary WLS for OCI are included in the Dynamic Group
- If you checked the "OCI Policies" flag when you provisioned primary, the Dynamic Group and the OCI policy were already created. Do the following:
    - Edit the dynamic group that was created for the primary instance (the name is like "*<prefix>-wlsc-principal-group*" and it is created in the root compartment). Remove the matching rule for the tag. The remaining rule will include all the resources in the compartment.

Then, don't check the "OCI Policies" flag when you provision secondary. The new resources are already included in the existing Dynamic Group, so the previously created policies are enough.

Once the provisioning process completes, the WebLogic servers can be sanity verified.

NOTE: Verify that the db connect string in the datasources is compliant with the recommended formats:
If the database is a single instance, the recommended db connect string is:
jdbc:oracle:thin:@//<db-scan-address>:<port>/<pdb_service_name>
If the database is a RAC, the datasources must be GridLink datasources and the recommended db connect string is:
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=<db-scan-address>)(PORT=<port>)))(CONNECT_DATA=(SERVICE_NAME=<pdb_service_name>)))

If for any reason, a long time passes since you provision the secondary WebLogic for OCI instance until you continue with the DR setup steps, you can stop the WebLogic administration server and managed servers in secondary and convert the standby database to physical standby again. This way, the redo apply gap between standby and primary database does not increase. After you do this, do not try to start the WebLogic servers in secondary site until the DR setup is completed. Because, before the DR setup, the secondary servers look for the original secondary schemas in the database, and they are not longer there. This is expected because the changes performed to a snapshot database are lost when it is converted to physical standby again. In this type of scenario, just keep secondary WebLogic admin and managed servers stopped and continue with the DR setup tasks. You will be able to start them once the DR setup is completed.

6. Prepare Secondary mid-tier for the virtual front-end

Add the front-end name and IP to the /etc/hosts file in all secondary mid-tier hosts.
With **root** user, edit the /etc/hosts file and map the SECONDARY LBR IP to the virtual front-end name. Repeat in all secondary mid-tier hosts. Example:

Copyright © 2024, Oracle and/or its affiliates | Public

```
[oracle@wlsociprefix-wls-0 ~]$ more /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

...
# Front-end virtual name
222.222.222.222    mywebapp.mycompany.com
```

You do not need to update the front-end address for the WebLogic cluster in the WebLogic Console, because that information will be copied from the primary WebLogic Domain configuration.

NOTE: do not alter the /etc/hosts file of the secondary midtier hosts when there is a switchover or failover. Secondary hosts must always resolve the virtual front-end name with its front-end IP. The DNS update that is needed during the switchover and failover procedures is performed in the DNS or host files used by the clients.

7. Prepare secondary mid-tier for using TNS alias

Just as in the primary system, you should use a TNS alias in all datasource files under $DOMAIN_HOME/config/jdbc and in the JPS config files under $DOMAIN_HOME/config/fmwconfig.

The alias name used in the secondary system must be the **same one as in the primary**. The **tnsnames.ora** file of the secondary system must contain **the same TNS alias entry but pointing to the secondary database**.

To prepare secondary mi-tier for using TNS alias, you don't need to modify the datasources and JPS config files, because they will be copied over from the primary system during the DR setup. Just create the **tns admin folder** in the secondary midtier hosts in the same path that in primary system. Create the **tnsnames.ora** file and add the same alias than in primary but pointing to the secondary PDB, with appropriate scan address and service name. Ensure that the tnsnames.ora contains any other tns alias required by your system.

8. Configure required mid-tier host aliases

The WebLogic domain configuration in secondary will be a copy of the primary WebLogic domain once the DR setup is completed. Hence, the hostnames used as listen addresses by the primary WebLogic servers (which are the hostnames of the primary mid-tier hosts) need to be valid in the secondary location but mapped to the secondary IPs.

And the other way around: the hostnames of the secondary servers need to be valid in the primary location but mapping to the primary IPs. This part is not essential, because normally only the primary hostnames names are used in the WebLogic configuration. This is done to avoid errors in primary in case that any reference to secondary names is added to the config while the secondary site takes the primary role.

To configure the required hostnames mapping, you can use two approaches: adding the hostnames as aliases to the /etc/hosts files or adding them to private DNS views.

8.1 Option 1) Use the /etc/hosts files

The other site's hostnames are added as aliases to the /etc/hosts files in the mid-tier hosts.

This mode is valid in all the scenarios: when the same DNS server is used in primary and secondary sites, and when separated DNS servers are used in primary and secondary. Because the entries in the /etc/hosts file have precedence over the DNS resolution. This precedence is defined in the directive "hosts" of the /etc/nsswitch.conf. By default, it is set to "files", which means that /etc/hosts resolution takes precedence over the DNS.

A disadvantage of this method is that it requires to manually add the entries to all the WLS hosts. So, when you add new nodes in a scale-out operation in secondary, the new node is not able to resolve the names until you modify its /etc/hosts file. This requires additional manual steps in the scale-out operation (see Scale-out and Scale-in for BV Replica Method and Scale-out and scale-in for DBFS and FSS with rsync Methods sections of this document for more details).

To configure the required aliases:

a) In all the mid-tier hosts (primary and standby), **edit the file /etc/oci-hostname.conf** as root user and set PRESERVE_HOSTINFO=3, so the changes implemented in next steps in the /etc/hosts are preserved after node reboots.

Copyright © 2024, Oracle and/or its affiliates | Public

b) **Identify the hostnames** of each WebLogic hosts in primary and standby where the server listens. To get them, you can look for the listen addresses in the domain configuration file. For example, in primary domain:

```
[oracle@wlsociprefix-wls-0 config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-0 config]$ grep listen-address config.xml
   <listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
   <listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
   <listen-address>wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
    <listen-address>wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
    <listen-address>wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com</listen-address>
```

And in the secondary domain:

```
[oracle@wlsociprefix-wls-0 config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-0 config]$ grep listen-address config.xml
   <listen-address> wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com </listen-address>
   <listen-address>wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
   <listen-address>wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
    <listen-address>wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
    <listen-address>wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
```

You can also use the command “hostname --fqdn” in each mid-tier host to get its FQDN hostname.

c) **Edit the /etc/hosts** (as root) in **all the primary mid-tier nodes** and add the hostnames of standby as aliases of the primary hosts. Each host should have entries as the following:

```
<IP_prim_node1> <long_and_short_hostnames_primary_node1> <long_and_short_hostnames_secondary_node1>
<IP_prim_node2> <long_and_short_hostnames_primary_node2> <long_and_short_hostnames_secondary_node2>
```

(short names are expected to be the same in primary and stby)
Example of the resulting entries /etc/hosts in primary mid-tier hosts:

```
[oracle@wlsociprefix-wls-0 config]$ more /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
# Front-end virtual name
111.111.111.111    mywebapp.mycompany.com

# For WLS DR
10.0.2.10  wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com  wlsociprefix-wls-0  wlsociprefix-wls-
0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
10.0.2.11  wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com   wlsociprefix-wls-1  wlsociprefix-wls-
1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
```

d) In the same way, **edit the /etc/hosts** (as root) in **all the secondary mid-tier** nodes and add the hostnames from primary hosts as aliases of the secondary hosts. Each host should have entries as the following:

```
<IP_secondary_node1> <long_and_short_hostnames_secondary_node1> <long_and_short_hostnames_prim_node1>
<IP_secondary_node2> <long_and_short_hostnames_secondary_node2>  <long_and_short_hostnames_prim_node2>
```

Example of the resulting /etc/hosts in secondary mid-tier hosts:

```
[oracle@wlsociprefix-wls-1 ~]$ more /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
# Front-end virtual name
222.222.222.222    mywebapp.mycompany.com

# For WLS DR
10.1.2.5  wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com   wlsociprefix-wls-0   wlsociprefix-wls-
0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
10.1.2.4  wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com   wlsociprefix-wls-1   wlsociprefix-wls-
1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
```


Copyright © 2024, Oracle and/or its affiliates | Public

8.2 Option 2) Use OCI Private DNS views

Instead of adding the entries to all the /etc/hosts files, you can add the required entries to **OCI private DNS views** in each VCN. Adding the entries to the private DNS views has advantages for the scale-out operations, because the new nodes in secondary site will be able to resolve the primary names out-of-the-box. Scale-out operations are simplified with this approach.
Check the Github repository https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_dr for detailed instructions and terraform scripts.

Copyright © 2024, Oracle and/or its affiliates | Public

COMPLETE THE DISASTER RECOVERY SETUP

Configure Using Block Volume Replica

NOTE: these steps apply to the Block Volume cross-region replica model only, if you are using the DBFS or FSS approaches, please skip this section and refer to the Configure Using DBFS and FSS with rsync methods one.

Follow these steps to complete the DR setup when using Block Volume cross-region replica model:

1. Convert the standby DB into physical standby

At this point, the WebLogic servers in the secondary must be stopped and the standby Database can be converted into physical standby again:

a) **Stop Oracle processes in secondary mid-tier hosts.**
Stop the WebLogic Managed Servers, the Admin server and the Node Manager processes in all the Secondary mid-tier hosts.
Also, in case they are mounted**, umount the DBFS mounts** in all Secondary mid-tier hosts.

b) **Convert the standby database into physical standby.**
Execute these steps as oracle user in the primary Database host:

```
[oracle@drdbaa ~]$ dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> CONVERT DATABASE secondary_db_unqname to PHYSICAL STANDBY;
Converting database " secondary_db_unqname" to a Physical Standby database, please wait...
Oracle Clusterware is restarting database "orclb" ...
Continuing to convert database " secondary_db_unqname" ...
Database " secondary_db_unqname" converted successfully
```

2. Configure the Block Volume Cross-Region replication

To replicate the block volumes of the mid-tier hosts from primary region to secondary region, follow these steps:

a) Identify the Block Volumes of the primary mid-tier hosts.
To identify these block volumes in primary region:

- Go to **OCI Console**, select your **Primary** region.
- Navigate to **Storage > Block Volumes**
- Choose the **compartment** of your primary WLS for OCI and look for the block volumes.
  Each compute instance has 2 block volumes:
  - **<wlsociprefix>-data-block-N**, mounted in /u01/data. This volume contains the WebLogic domain configuration in each compute instance. This block volume **needs to be replicated** to the standby site.
  - **<wlsociprefix>-mw-block-N**, mounted in /u01/app. According to the explanation in the previous section Block Volumes Replicated in BV Replica Method, you don't need to replicate them on an ongoing basis. The steps to replicate these block volumes are included here as optional.
- Note down the **names and the AD where they are located**. For example:
  wlsociprefix-data-block-0, in AD1
  wlsociprefix-data-block-1, in AD2
  Optional:
  wlsociprefix-mw-block-0, in AD1
  wlsociprefix-mw-block-1, in AD2

These Block Volumes are mounted as in this example:

```
[opc@wlsociprefix-wls-0 ~]$ df -h
```


Copyright © 2024, Oracle and/or its affiliates | Public

```
Filesystem     Size  Used Avail Use% Mounted on
devtmpfs       7.2G     0  7.2G   0% /dev
tmpfs          7.2G     0  7.2G   0% /dev/shm
tmpfs          7.2G  161M  7.1G   3% /run
tmpfs          7.2G     0  7.2G   0% /sys/fs/cgroup
/dev/sda3       39G  3.7G   35G  10% /
/dev/sda1      200M  8.6M  192M   5% /boot/efi
/dev/sdb        49G  1.4G   46G   3% /u01/app      → (optional) this Block Volume can be replicated
/dev/sdc        49G  176M   47G   1% /u01/data     → this Block Volume will be replicated
tmpfs          1.5G     0  1.5G   0% /run/user/0
tmpfs          1.5G     0  1.5G   0% /run/user/994
tmpfs          1.5G     0  1.5G   0% /run/user/1000
```

b) Identify the Block Volumes of the secondary mid-tier hosts.
Repeat the steps described in a) to get the names and Availability domains of the block volumes of the secondary mid-tier hosts.

c) Create Block Volume Groups in primary and enable the cross-region replica.
Create Block Volume Groups in primary to group all the block volumes that are going to be replicated. The replica will be enabled for the Volume Group, so it applies to all the Block Volumes in that group. A Volume Group can contain only Block Volumes that are in the same AD, so if your compute instances are located in more than one AD, create a Block Volume Group per AD.
To create a Block Volume Group and enable the cross-region replica:

- Log on to the Oracle Cloud Infrastructure Console in the primary region.
- Navigate to the Storage > Volume Groups
- Create a block volume group in the same Availability Domain (AD) as the compute instances. For example: <wlsociprefix>-BVGroup-AD1
- Add the block volumes that you will replicate to the volume group.

NOTE: do not add Boot Volumes. They are not replicated.

- Enable cross-region replication in the Volume group.
  - Target region: select the secondary region.
  - Availability domain: set the AD in secondary region where the computes that will mount these volumes are located.
  - Volume Group Replica Name: the name for the block volume replication.
- If the Block Volumes have already the cross-region replica individually set to ON, switch it to OFF. The replication must be configured in the Volume group.
- After saving changes, check that the replicas are being created in the secondary region: in the OCI Console, select the secondary region and navigate to **Storage > Block Storage > Volume Group Replicas**.

Repeat the same to create Block Volume Group in the other AD when your primary compute instances reside in more than one AD.

d) Detach the original Block Volumes from the Secondary mid-tiers hosts.

NOTE: Boot Volumes must NOT be unmounted or detached

For **each** mid-tier host in secondary, run the following:

- Unmount the data block volume, which is mounted in /u01/data:

```
[opc@wlsociprefix-wls-0 opc]# sudo umount  /u01/data
```

  Make sure that there are not Oracle processes running. It is expected that they are stopped at this point, but if there is something still running on that folder, the umount will fail.
- Once unmounted, detach the block volume from the OCI Console
  Go to each **Block Volume > Attached Instances > Detach from Instance**
  The OCI Console will ask you to run some iscsci commands before completing the detachment.

Copyright © 2024, Oracle and/or its affiliates | Public

- With root user**, edit the /etc/fstab file** and remove the entry for /u01/data. This is to prevent it from trying to mount the original block volumes in next reboot. Example:

```
..
UUID=7e5f896e-6e05-4be2-99e5-ddf2b3cbb021 /u01/app ext4 auto,defaults,_netdev,nofail 0 2
#Remove this entry:
#UUID=9e87cf72-a75c-4dff-9825-432f1668d8f9 /u01/data ext4 auto,defaults,_netdev,nofail 0 2
```

Repeat these steps for the rest of the mid-tier nodes in Secondary.

In case that the mw block volumes are also being replicated, perform the same steps for the mw block volumes and mount point /u01/app.

e) **Delete or rename the detached Block Volumes in secondary.**
The original block volumes that have been detached from the secondary mid-tier hosts in the previous step are not going to be used anymore. You can delete them or rename and delete later.

f) Restart the systemd daemon in the secondary compute instances to refresh any cached references to the previously mounted devices ("systemctl daemon-reload" with root user).

3. Prepare the script for the environment specific replacements

During a switchover or failover operation, after mounting the replicated block volumes in the secondary mid-tier hosts, you need to perform a replacement on the WebLogic Domain configuration. This is because the WebLogic Domain configuration is a copy from primary, so the TNS entry in the tnsnames.ora points to primary database. Replace it with the secondary database connection details.

To automate this replacement, use the script **replacement_script_BVmodel.sh.** As well as replacing the connect string, this script also cleans up some state files of Weblogic serves (.lck and .state) for a clean startup.

a) Download the script from https://github.com/oracle-samples/maa/tree/main/wls_mp_dr/Block_Volume_Replica_Method
b) Upload it **to all the mid-tier hosts** (primary and secondary).
c) **Store it in a folder** that is NOT in the Block Volume that is replicated. For example, in a folder under the oracle user's home (for example, /home/oracle/scripts).
d) Change the ownership of the file **to oracle user** (this script will be executed by oracle user).
e) Edit the script and customize it in each host with the apropriate values, by providing the local and remote values for the database in each site.

DO not run the script at this point. The script will be used next time that a switchover or failover is performed.

The WLS for OCI DR setup is ready. Continue in Validate the DR Setup.

Copyright © 2024, Oracle and/or its affiliates | Public

Configure Using DBFS and FSS with rsync methods

NOTE: these steps apply to DBFS and FSS with rsync methods only.

Follow these steps to complete the DR setup when using DBFS or FSS with rsync models:

1. Configure the DBFS staging mount in DBFS method

A DBFS file system is a standard file system interface on top of files and directories that are stored in database table. As it is stored in the database, a DBFS file system can be used as a shared file system accessible by all the mid-tier hosts.

In WebLogic for OCI DR based on DBFS method, a DBFS mount is used for the initial setup, and for replicating WebLogic domain configuration changes during the system's lifecycle. This DBFS filesystem is an assistance filesystem used to sync changes from primary to standby: the required config files from primary domain are copied to the DBFS filesystem (local copy) and they are automatically replicated to the standby site thanks to Data Guard underlying replication. In the standby site, the replicated config files are then copied from the DBFS filesystem to the domain folder.

This document provides a script to automate this copy in primary and standby. This DBFS mount is not used for other WebLogic runtime operations, so it is not critical for the service nor has a big impact on the performance of the system.

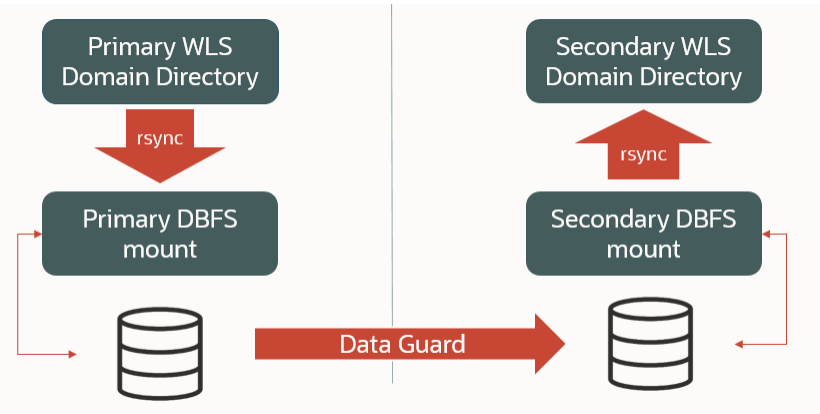


Figure 9 Using an assistance DBFS file system to replicate WLS domain configuration to secondary site.

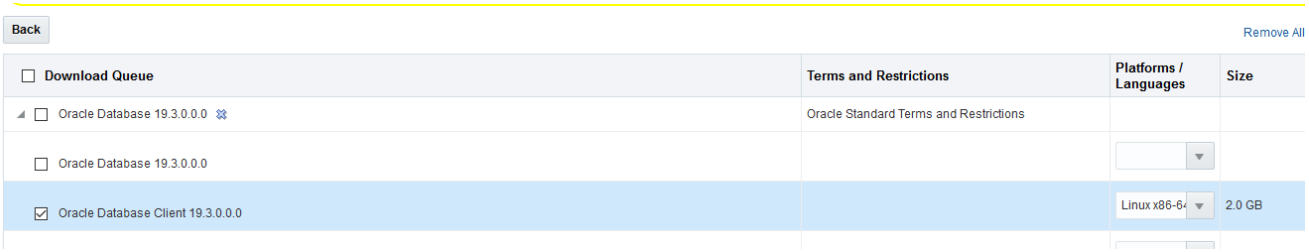
There is no DBFS mount configured by default in the WebLogic hosts, and you need to configure it manually. This requires the **installation of the Database client** and **some operating system packages.** To perform these actions in each mid-tier host, follow these steps:

a) **Download the DB client** from e-delivery and **upload** it to the mid-tier host (do NOT install it yet). Be sure that you download the **installer** version, not the image-based installation. It is recommended to use the latest version.
Here is an example to download the 19.3 db client from e-delivery.
Search for "Oracle Database Client":

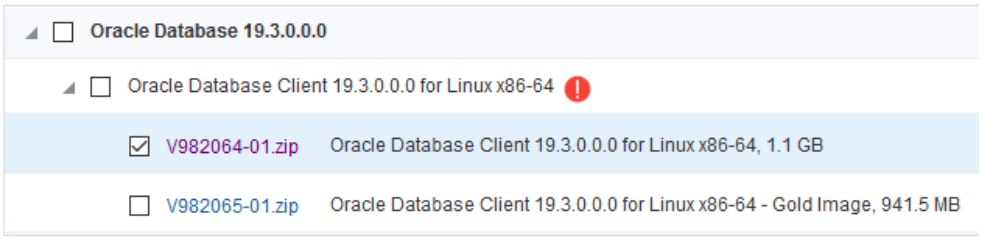
Then select 19.3 and continue:

Then select the database client only:

Copyright © 2024, Oracle and/or its affiliates | Public



Click continue and then select the installer version (not the gold image):



Download the software and upload it to all the midtier hosts, for example to /u01/install/V982064-01.zip

b) **Locate the script dbfs_dr_setup_root.sh** in the scripts you downloaded and **upload** it to the mid-tier hosts. This script performs the required tasks to get the DBFS mount ready in the host: it installs the db client and its required operating system packages, it configures the dbfs user and schema in the database, and it mounts the DBFS file system and creates a cron, so the DBFS file system is mounted on host boot.

c) Then, **execute** it as **root** user. Syntax is:

```
./dbfs_dr_setup_root.sh  <local_db_scan_name> <db_port>  <local_PDB_service> <pdb_sys_password>
<path_to_dbclient_installer>
```

As input parameters, you need to provide the connection data used to connect to the local database used by the WLS. Provide primary PDB connection data in primary site midtier nodes and provide secondary PDB connection data in secondary site midtier nodes. You can get the local scan name, port and PDB service name from the datasources of each domain.

NOTE: the standby database must be in snapshot standby mode at this point.

Example to run it primary midtier hosts (it must be a single line). You must provide primary PDB values:

```
./dbfs_dr_setup_root.sh  drdba-scan.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com 1521
PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com mypassword   /u01/install/V982064-01.zip
```

Example to run it secondary midtier hosts (it must be a single line). You must provide secondary PDB values:

```
./dbfs_dr_setup_root.sh  drdbb-scan.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
1521 PDB1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com mypassword   /u01/install/V982064-01.zip
```

When running the dbfs_dr_setup_root.sh script in a RAC scenario, use the scan address name of the local RAC: primary scan address when running it in primary mid-tier nodes, and secondary RAC scan address when running it in secondary mid-tier nodes. Provide the appropriate pdb service name, so dbfs will connect to a CRS service rathen than the default PDB service. Example of execution in a RAC scenario:

```
./dbfs_dr_setup_root.sh  drdbrac2a-scan.subnetlon1.myvcnlon.oraclevcn.com 1521 mypdbservice.mycompany.com
mypassword /u01/install/V982064-01.zip
```

d) **Verify** that the DBFS mount is present in the mid-tier host:

```
 [root@ wlsociprefix-wls-1]# df -h | grep dbfs
dbfs-@PDB1:/    32G  248K   32G   1% /u02/data/dbfs_root

[root@ wlsociprefix-wls-1]# ls /u02/data/dbfs_root
dbfsdir
```


Copyright © 2024, Oracle and/or its affiliates | Public

Repeat these steps in **all the mid-tier hosts** (primary and secondary).

As the result of the execution of this script:

- The **db client** software is installed in the midtier host, in the folder /u01/app/oracle/client. The required packages for the db client are installed using yum.
- A user is created in the PDB database for DBFS. The **username is *dbfsuser***, and its password is set to the same sys password provided as input of the script.
- A **DBFS tablespace** is created in the PDB database for the DBFS mount (tablespace name is *tbsdbfs*), and a **DBFS folder (name *dbfsdir***).
- A **new folder** is created in the domain, **DOMAIN_HOME/dbfs**. It contains the wallet is to store the dbfsuser's password, and other artifacts needed (tnsnames.ora, sqlnet.ora). This wallet is used by the db client to mount the dbfs mount in the midtier host.
- The **script dbfsMount.sh** is also created in DOMAIN_HOME/dbfs. This script is used to mount the dbfs mount in the midtier. It is also added to the cron on reboot, so the script is executed when the machine is rebooted.
- The DBFS mount is mounted in **/u02/data/dbfs_root mount point** as the **folder *dbfsdir***.

The script can be re-run. In case of reruning, there are warnings because some things are already created (db user, tablespace, etc.), but these messages can be ignored.

2. Configure the FSS staging mount in FSS with rsync method

When you use the **FSS with rsync method** to replicate the WebLogic configuration, create two FS filesystems: one in the primary site and another in the secondary site. These file systems are mounted by the local hosts only. There are no direct cross-region NFS mounts in the topology for security and performance reasons These filesystem mounts are used as staging areas for the content that is replicated between sites with rsync commands. They store a copy of the domain folder. They are not used for runtime.

During initial DR setup, the primary FSS volume is mounted on the **primary site WLS Administration host** and the secondary FSS volume is mounted **on all the secondary site WLS hosts**. The mount of the secondary site needs to be available on all the secondary midtier hosts because it is used as the source for the initial copy of the replicated domain performed during the DR setup phase.

NOTE: Once you have completed the first config sync (which is done during the initial DR setup), the FSS mounts are only required in the primary and standby WLS Administration hosts. You can umount them from the other WLS nodes, unless you use them to store additional artifacts that require to mount them in all the nodes.

Copyright © 2024, Oracle and/or its affiliates | Public

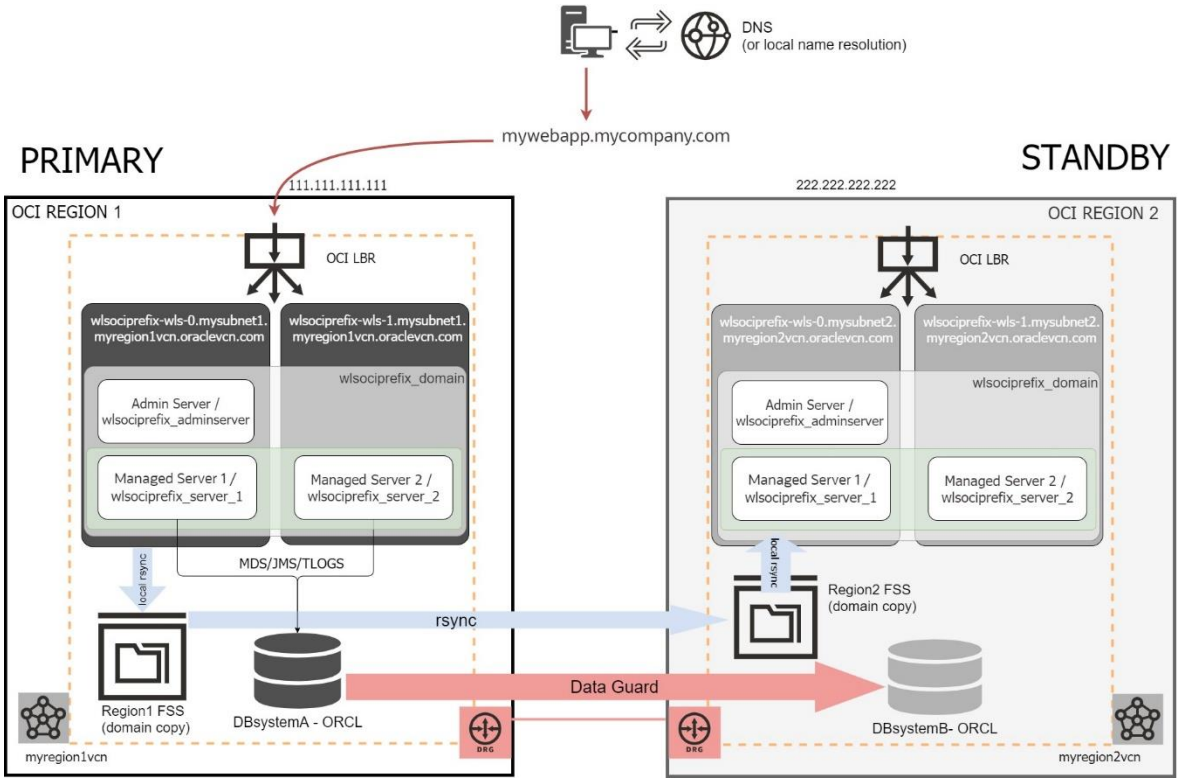


Figure 10 Using assistance FSS filesystems to replicate WLS domain config via rsync. The blue arrows just represent the logical flow of the configuration copy. The rsync commands run either in primary or standby site's WebLogic Administration hosts. I.e., for the remote copy, primary site's WebLogic Administration host connects to standby WebLogic Administration host with rsync.

The provisioning wizard of the **Oracle WebLogic Server for OCI stacks** allows **you to automatically create new file storage or reuse existing files storage,** mounting it in the /u01/shared directory of the nodes. Make sure you check the "Add File System" in your WLS for OCI stacks. This option can be checked during the provisioning and post-provisioning.

Otherwise, you will need to create and mount the file systems manually, as described in the following steps:

	Steps to configure and mount FSS	DETAILS	SAMPLE VALUES IN PRIMARY	SAMPLE VALUES IN SECONDARY
1	Create a **mount target** in each region (if it does not already exist)	-Connect to OCI Console -Select the proper region and compartment (primary or secondary) -Go to "File storage" > "Mount target". -Click "Create Mount target" -Once created, note down the IP of each one	New Mount Target Name: **primary_mt** Availability Domain: <same than primary wls> Virtual Cloud Network: <same than primary wls> Subnet: <same than primary wls>	New Mount Target Name: **secondary_mt** Availability Domain: <same than secondary wls> Vritual Cloud Network: <same than secondary wls> Subnet: <same than secondary wls>
2	Create **a file system** in each region	-Connect to OCI Console -Select the proper region (primary or secondary) -Go to "File storage" > "File System" -Click "Create File System" -Select the proper mount target in each case	Name: **primary_fs** Availability Domain: <same than primary wls> Export: **/primaryfs** Mount target: **primary_mt** (previously created)	Name: **secondary_fs** Availability Domain: <same than secondary wls> Export: **/secondaryfs** Mount target: **secondary_mt** (previously created)
3	Validate/set the network security	Some network rules in each subnet are required to allow the NFS traffic between hosts and mount target. Use the		

Copyright © 2024, Oracle and/or its affiliates | Public

	rules required for FSS mount	instructions in Configuring VCN Security Rules for File Storage to set up security rules correctly for your file systems.		
4	**Mount** the file system on the midtier hosts	In ALL WLS hosts: -Create the local mount point. Example: # sudo mkdir /u01/shared -With user root, edit /etc/fstab and add the mount, with the **apropriate** mount target IP (primary mount target IP in primary site and secondary mount target IP in secondary site): <mount_target_ip_address>:<export_name> <your_local_mount_point> nfs defaults,nofail,nosuid,resvport 0 0 -Mount the new filesystem: # sudo mount -a Reference: Mounting File Systems From Unix-Style Instances	Example line to add in primary hosts' /etc/fstab (must be in one line): 10.1.1.1:/primaryfs /u01/shared nfs defaults,nofail,nosuid,resvport 0 0 Example line to add in secondary hosts' /etc/fstab (must be in one line): 10.2.2.2:/secondaryfs /u01/shared nfs defaults,nofail,nosuid,resvport 0 0	
5	**Verify** mounted file system	In all WLS hosts: # df -h | grep /u01/shared # ls -la /u01/shared	Sample result in primary wls hosts: [root@wlsociprefix-wls-0 opc]# df -h | grep primaryfs 10.1.1.1:/primaryfs 8.0E 0 8.0E 0% /u01/shared Sample result in secondary wls hosts: [root@wlsociprefix-wls-0 opc]# df -h | grep secondaryfs 10.2.2.2:/secondaryfs 8.0E 0 8.0E 0% /u01/shared	
6	**Change the owner** of the mounted file system to oracle user	-Once the FSS volumes are mounted, make oracle user the owner of the mount point folder: # sudo chown oracle:oracle /u01/shared	Run this in primary admin host (the mount is shared by the rest of primary wls hosts so no need to repeat)	Run this in secondary admin host (the mount is shared by the rest of secondary wls hosts so no need to repeat)

3. Run DR setup script in Primary WebLogic Domain

Run the primary DR setup script **fmw_dr_setup_primary.sh** in the primary mid-tier Oracle WebLogic **Administration Server host node**. The script will copy the primary domain configuration to the secondary site (using DBFS or FSS with rsync method).

No downtime will be caused by the execution of the script in the WebLogic servers or in the backend Database.

a) **Upload** the scripts you downloaded from Github to the **primary admin** host. Place all the scripts in the same folder and make sure they have execution permission.

b) **Run** the script **fmw_dr_setup_primary.sh** as **oracle** user in the **primary WebLogic Administration host**.
It can be run interactively or non-interactively:

Interactive usage:

```
fmw_dr_setup_primary.sh
```

Non-interactive usage:

```
fmw_dr_setup_primary.sh <dr_method> [db_sys_password] [remote_admin_node_ip] [remote_keyfile] [fss_mount]
```

Where:

dr_method: is the method used to replicate the WLS domain config. Valid values are DBFS or RSYNC:
DBFS: use this value when using DBFS based method
RSYNC: use this value when using FSS with rsync method

Copyright © 2024, Oracle and/or its affiliates | Public

db_sys_password:	Required only if *dr_method* is DBFS. It is the password of the database sys user
remote_admin_node_ip:	Required only if *dr_method* is RSYNC. It is the IP address of the **secondary WLS Administration server host**. It needs to be reachable from the primary admin host. NOTE: **you must use the IP**, and not the hostname. Recommended approach is to provide the private IP, so the communication between sites goes through internal networks (primary and secondary networks communicated via Dynamic Routing Gateway).
remote_keyfile:	Required only if *dr_method* is RSYNC. Is the complete path to the private keyfile required to ssh to secondary admin host. Make sure that the file has been uploaded to the primay WLS Administration server host and is readable by oracle user only.
fss_mount:	Required only if *dr_method* is RSYNC. This is the path to the mount point where the OCI File Storage file system is mounted. This OCI File Storage file system will be used to stage the WLS domain configuration. Example: /u01/shared

Example when you are using the **DBFS based method**:

```
./fmw_dr_setup_primary.sh DBFS 'mypassword'
```

Example when you are using the **FSS with rsync method:**

```
./fmw_dr_setup_primary.sh RSYNC 10.1.0.2 /u01/install/sshkeys/my_priv_key.priv /u01/shared
```

NO need to run this script in the rest of the primary mid-tier hosts.

NOTE: the script looks for the datasource opss-datasource.xml by default. It uses it as a reference to gather database related information like the tns admin folder, db connect string etc. If the WLS for OCI environment is not a JRF-enabled domain and the opss-datasource.xml does not exists, the script will provide a message and stop the execution. If that is the case, you can edit the fmw_dr_setup_primary.sh script and provide an existing datasource config file name in the defined variable datasource_name.

4. Run DR setup script in Secondary WebLogic Domain

Run the script **fmw_dr_setup_standby.sh** script in all the secondary mid-tier hosts. This script performs the required operations to configure the secondary mid-tier cluster to act as standby for the primary one.

a) When the DR method is DBFS, the script requires access from the mid-tier hosts in secondary to the primary database in the listener port (usually 1521). **Enable** the appropriate ingress rules in the involved subnets for this **communication** in the OCI console.
If the communication between primary and secondary sites is done via Internet Gateway, open communication from secondary mid-tier hosts public IPs to the primary db.
If the communication between primary and secondary sites is internally routed **(recommended approach)**, using a Dynamic Routing Gateway, allow incoming connections from secondary mid-tier hosts private IPs to the primary database.

Note this is only required for setup and lifecycle operations. Cross-region communication from midtier to remote database will not be used for runtime.
A quick check can be run on all the secondary midtiers with user oracle to verify the connectivity to private/public primary database IPs before running the script, depending on the network scenario:

```
java -classpath /u01/app/oracle/middleware/wlserver/server/lib/weblogic.jar utils.dbping ORACLE_THIN system
<system_password>  <primary_db_ip>:1521/<primary_db_service>
```

c) **Upload** the scripts you downloaded from Github to all the secondary mid-tier hosts. Place all the scripts in the same folder and make sure they have execution permission.
b) **Stop** the secondary WebLogic servers (admin and managed) and node managers in **all secondary mid-tier hosts**.
c) Connect to primary DB and **convert** the standby database **to physical standby**:

Copyright © 2024, Oracle and/or its affiliates | Public

```
[oracle@drdba ~]$ dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> CONVERT DATABASE "secondary_db_unqname" to PHYSICAL STANDBY;
Converting database "secondary_db_unqname " to a Physical Standby database, please wait...
Database "secondary_db_unqname" converted successfully
```

Check the status and wait until it shows “SUCCESS”. Example:

```
DGMGRL> show configuration
Configuration - ORCL_lhr2bb_ORCL_fra22g_12:09:
 Protection Mode: MaxPerformance
 Members:
 ORCL_lhr2bb - Primary database
   ORCL_fra22g - Physical standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 25 seconds ago)
```

d) **Run** the script **fmw_dr_setup_standby.sh** in the first secondary node as user oracle. This script requires some input parameters. It can run in interactive as well as non-interactive mode. Open it to read the instructions before executing it.
Interactive usage:

```
 fmw_dr_setup_standby.sh   (NOTE: user will be prompted for all values)
```

Non-interactive usage:

```
 fmw_dr_setup_standby.sh <DR_METHOD> [A_DB_IP]  [A_PORT]  [PDB_SERVICE_PRIMARY]  [SYS_DB_PASSWORD]
[FSS_MOUNT]
```

Where:

DR_METHOD:	the method used to replicate the WLS domain config. Valid values are DBFS or RSYNC: DBFS: use this value when using DBFS based method. RSYNC: use this value when using FSS with rsync method.
A_DB_IP:	Required only if *dr_method* is DBFS. It is the **primary** database **IP.** Provide the **private IP** when the communication between sites is possible via internal IPs (when Dynamic Routing Gateway is used, which is the recommended approach). Otherwise, provide the primary database public IP. Provide one of the primary’s scan IPs in case of a RAC.
A_PORT:	Required only if *dr_method* is DBFS. Is the primary database listener port.
PDB_SERVICE_PRIMARY:	Required only if *dr_method* is DBFS. Is the primary PDB service name.
SYS_DB_PASSWORD:	Required only if *dr_method* is DBFS. Is the primary database sys password.
FSS_MOUNT:	Required only if *dr_method* is RSYNC. This is the path to the mount point where the OCI File Storage file system is mounted. This OCI File Storage file system will be used to stage the WLS domain configuration copy.

Example when you are using the **DBFS based method**:

```
./fmw_dr_setup_standby.sh DBFS '10.0.2.2' '1521' 'PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com' mypassword'
```

If the database is a RAC Database, provide one of the scan IP addressescof the primary RAC. You can also provide the Clusterware Database Service name instead of the default pdb service, to this service will be used to connect to the PDB during the DR setup. Example of execution in a RAC scenario:

```
./fmw_dr_setup_standby.sh DBFS '10.0.2.34' '1521' 'mypdbservice.mycompany.com' mypassword'
```

Example when you are using the **FSS with rsync method**:

```
./fmw_dr_setup_standby.sh RSYNC  '/u01/shared'
```


Copyright © 2024, Oracle and/or its affiliates | Public

NOTE: the script looks for the datasource opss-datasource.xml by default. It uses it as a reference to gather database related information like the tns admin folder, connect string, etc. If the WLS for OCI environment is not a JRF-enabled domain and the opss-datasource.xml does not exists, the script will provide a message and stop the execution. If that is the case, you can edit the fmw_dr_setup_standby.sh script and provide an existing datasource config file name in the defined variable datasource_name.

e) In DBFS DR method, the script will convert the database in snapshot standby mode so secondary servers in the node can be verified in the next steps.
In RSYNC DR method, the script does not perform any db role conversion. Before continuing, **you must manually convert the standby database to snapshot standby** if you want to start and validate the WLS processes in the standby site.

f) **Start** the **Node Manager** in the host. Example to start the node manager:

```
[oracle@wlsociprefix-wls-0 ~]$ cd $DOMAIN_HOME/bin
[oracle@wlsociprefix-wls-0 bin]$ nohup ./startNodeManager.sh > $DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 &
```

g) **Start** the **WebLogic servers** that run in the host (start the admin server if this is the admin server host, and then the managed server that run in this host).
Example to start the Administration server using WLST:

```
[oracle@wlsociprefix-wls-0 ~]$ cd $MIDDLEWARE_HOME/oracle_common/common/bin
[oracle@wlsociprefix-wls-0 ~]$./wlst.sh
wlst>  nmConnect ('weblogic','passwd','<admin_host_name>','5556','<domain_name>','/u01/data/domains/<domain_name>','SSL')
wlst> nmStart('<admin_server_name>')
```

To start the managed server in the host you can use the WLS Admin Console (Environments > Servers > Control)

NOTE: In non JRF-enabled domains (e.g. 14.1.1), the node manager does not use KSS as the identity keystore. It uses the $DOMAIN_HOME/security/DemoIdentity.jks by default. This file is different in each host. To avoid "Hostname verification failed" errors when connecting to the node managers for starting the servers, copy the DemoIdentity.jks file from primary node wls-1 to secondary node wls-1, from primary wls-2 to secondary wls-2, etc. The same applies if you are not using KSS and you have customized the JKS identity keystore files
This is a one time action. You only have to perform this copy after the DR setup.

h) **Verify** the server is started in RUNNING and check the sample app urls and administration consoles.
i) Repeat steps from **"c)"** (convert the database to physical standby) **to "h)"** (verify) to run the script **in all the rest of the mid-tier nodes** that are part **of the secondary cluster**.

Once finished, **stop Admin and Managed servers** in the secondary domain and **convert** the **standby database back to physical standby.**

The WLS for OCI DR setup is ready. Continue in Validate the DR Setup.

Copyright © 2024, Oracle and/or its affiliates | Public

Complete DR configuration when IDCS authentication is used

If you have enabled IDCS authentication for your WLS for OCI system, **additional steps** are required to complete the DR configuration. Both, primary and standby need to use the same IDCS services for the authentication data to be consistent.

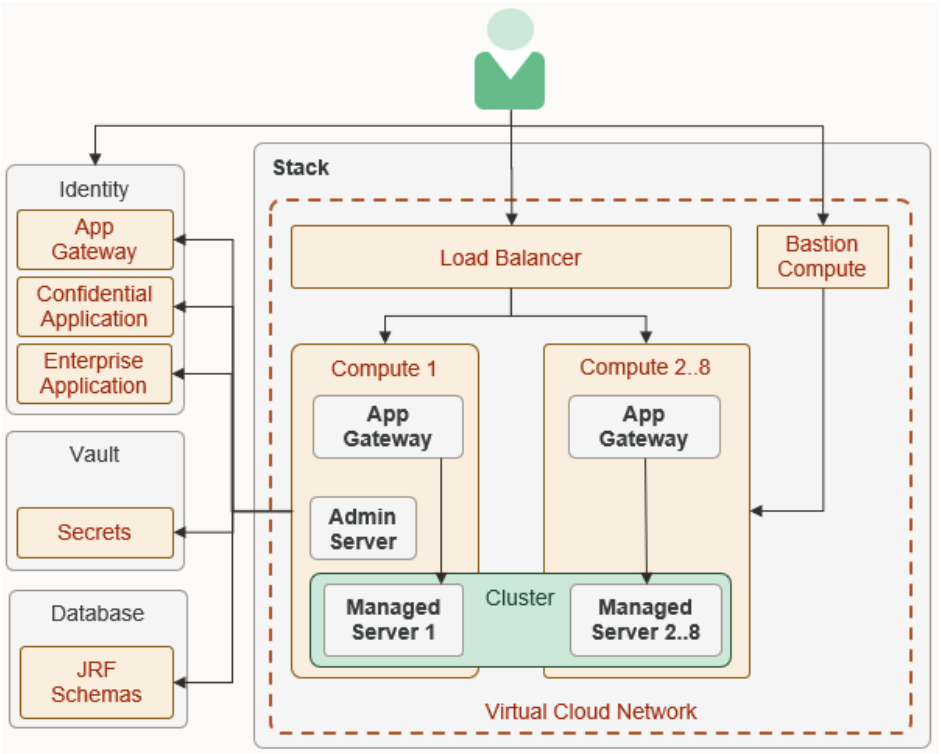
NOTE: This document addresses the implications of using IDCS authentication for WLS for OCI in context of a DR event. Notice, however, that IDCS's own disaster protection is out of the scope of this document [7].

As an overview: when WLS for OCI system is provisioned with "Enable authentication using IDCS" checked, the following applies:

- As a **pre-requisite, a Confidential Application** needs to be manually created previously. See information about this in the documentation of WLS for OCI at the Before You Begin with Oracle WebLogic Server for Oracle Cloud Infrastructure > Create a Confidential Application section
  The client ID/secret ID of this Confidential Application are provided by the customer in the provisioning screen. This is required for the provisioning time but it is not used anymore once the WLS for OCI system is created. More
- A **Confidential Application** is automatically created in the IDCS during provisioning for the WLS for OCI Service. This application is added as authenticator provider to WebLogic when the WebLogic domain is created. The application name is built using *service_name + _confidential_idcs_app_ + timestamp.*
- An **Enterprise Application** is automatically created in the IDCS during provisioning for the WLS for OCI Service. This application has the webtier policies (in the form of resources and authentication policies) to protect resources. The application name is built using *service_name + _enterprise_idcs_app_ + timestamp*.
- An **App Gateway** is automatically created in the IDCS during provisioning for the WLS for OCI Service.
  This is used to connect the app gateway with the Enterprise Application. The client id and client secret of this artifact is configured in the app gateway docker image that runs in each VM of the instance. The name App Gateway is built using *service_name + _gateway_ + timestamp*.

[7] Contact IDCS support to check the IDCS Disaster Recovery solution availability in your region and to get details on how IDCS itself is protected against disasters.

Copyright © 2024, Oracle and/or its affiliates | Public



According to this, once the primary and the secondary WLS for OCI systems have been provisioned, these are the artifacts that exist in the IDCS:

- The pre-required confidential application. The same can be used to provision primary and standby, it is used during provisioning but not used later.

- The Confidential Application, Enterprise Applications and App Gateway of the **PRIMARY** WLS for OCI system.
  Example:
  wlsmkpl7_confidential_idcs_app_2020-09-17T10:53:28.031194
  wlsmkpl7_enterprise_idcs_app_2020-09-17T10:53:28.031194
  wlsmkpl7_app_gateway_2020-09-17T10:53:28.031194

- The Confidential Application, Enterprise Applications and App Gateway of the **SECONDARY** WLS for OCI system.
  Example:
  wlsmkpl7_confidential_idcs_app_ 2020-09-18T11:55:44.241334
  wlsmkpl7_enterprise_idcs_app_ 2020-09-18T11:55:44.241334
  wlsmkpl7_app_gateway_ 2020-09-18T11:55:44.241334

NOTE: Although the names are similar in primary and standby because they are derived from the resource name prefix, note that the timestamp will be different. Later timestamp for secondary artifacts because secondary is created after primary.

This document explains how to rewire the secondary WLS for OCI system to use the same IDCS services that are used by primary, so the identity information in primary and secondary is consistent. To achieve this, secondary domains must use the same Confidential Application, Enterprise Application and App Gateway than primary. Follow these steps to complete this DR setup:

e) Configure the primary Confidential Application in the secondary WebLogic domain.
The Confidential Application is added as authenticator provider to WebLogic when the WebLogic domain is created. As the primary domain configuration has been replicated to secondary, **the secondary domain will be already configured with the primary Confidential** application after running the DR setups in secondary WebLogic Domain.

To **verify** that the primary Confidential Application is used by secondary:

Copyright © 2024, Oracle and/or its affiliates | Public

- Login into the IDCS console.
  To get the link of your IDCS Console, go to the OCI Console > Identity > Federation
  Then click in your Identity provider, the **Oracle Identity Cloud Service Console** is shown in the Identity Provider information. Example
  https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole
- Navigate to "Applications" and **identify the primary Confidential Application** name.
  The name os the Confidential Application is *service_name + _confidential_idcs_app_ + timestamp*
  Example: wlsmkpl7_confidential_idcs_app_2020-09-17T10:53:28.031194

NOTE: make sure you identify the Confidential Application created for primary and not the secondary. The names are similar, but the timestamp will be earlier in the primary one.

- Click in that application and then go to "Configuration" > "General Information"
- Identify the **Client ID.** Example: 87b58e51a4744f95897a2fd72ae5c987
- Login with SSH to the admin node of the secondary Domain, and view the content of the file <DOMAIN_HOME>/config/config.xml
- Look for the IDCSIntegrator, and verify that the client-id is the same as that for the primary Confidential Application. Example:

```
<sec:authentication-provider xmlns:idcs="http://xmlns.oracle.com/weblogic/security/idcs" xsi:type="idcs:oracle-identity-cloud-integratorType">
<sec:name>IDCSIntegrator</sec:name>
    ..
    <idcs:tenant>idcs-5e890de598888888888d70064c5e00718</idcs:tenant>
    <idcs:client-tenant>idcs-5e890de598888888888d70064c5e00718</idcs:client-tenant>
    <idcs:client-id>87b58e51a4744f95897a2fd72ae5c987</idcs:client-id>
```

To make sure that there are no accesses to the secondary Confidential Application, deactivate it:

- Login into your IDCS console. For example, https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole
- Navigate to "Applications" and identify the **secondary Confidential Application**.
  The name of the Confidential Application is *service_name + _confidential_idcs_app_ + timestamp*
  Example: wlsmkpl7_confidential_idcs_app_2020-09-18T11:55:44.241334
- Click in "Deactivate" to inactivate it.

NOTE: make sure to deactivate the secondary's confidential application and not the primary. The names are similar, but the timestamp will be later in the secondary one. In any case, if by mistake you deactivate the primary confidential application, you can activate it back.

f) Configure the primary Enterprise application and App Gateway in the secondary WebLogic domain

The App Gateway is configured in the Cloud App Gateway that runs in each WebLogic node. At this point, the secondary Cloud app gateways that run in the secondary domain will be pointing to the secondary Enterprise Application and App Gateway. In order to make them to point to the same artifacts than primary, follow these steps:

- Identify the Client ID and Client Secret of the **primary App Gateway**.
  For this, login in the IDCS console > Security > App Gateway and click in the primary App Gateway.
  The name of the App Gateway Application is *service_name + _app_gateway_ + timestamp*
  Example: wlsmkpl7_app_gateway_2020-09-17T10:53:28.031194

NOTE: make sure to identity the primary App Gateway and not the secondary. The names are similar, but the timestamp will be earlier in the primary one.

- Note down the Client ID and Client Secret. Example:

```
Client ID:      28dabe1c9de4410dab27b6e4febb2bfa
Client Secret:  f1f9589c-ee71-4a73-ae2b-f66ef7bbd6a5
```


Copyright © 2024, Oracle and/or its affiliates | Public

- Login SSH to the admin node in the secondary WLS for OCI domain.
- Stop the app gateway docker container, login as opc and run:

```
sudo docker stop appgateway
```

- Take a backup of existing cwallet file /u01/data/cloudgate_config/cwallet.sso

```
cp /u01/data/cloudgate_config/cwallet.sso /u01/data/cloudgate_config/cwallet.sso.bak
```

- Add the following lines to the file /u01/data/cloudgate_config/appgateway-env:

```
CG_APP_NAME=<client_id>
CG_APP_SECRET=<client_secret>
```

Where <client_id> and <client_secret> are the values of the PRIMARY app gateway.

- If the idcs tenant that was used to provision primary is different than the idcs tenant that was used to provision secondary (for example, if this is a cross-continent DR), you need to set the same idcs than primary in the following places:
In the file /u01/data/cloudgate_config/appgateway-env, in these properties:

```
CG_APP_TENANT=<primary_idcs_tenant>
# The URL required to access the instance of IDCS.
IDCS_INSTANCE_URL=https://<primary_idcs_tenant>.identity.oraclecloud.com:443
```

In the file cloudgate.config, in these properties:

```
"bootstrap": {
   "externalIdUrl": "https:// <primary_idcs_tenant>.identity.oraclecloud.com:443",
   ….
    "tenantName": "<primary_idcs_tenant>",
```

Sample value for <primary_idcs_tenant>" is “idcs-82507b399af654467dccf94029c895ab0”

- Regenerate the wallet and change the permisions and mode

```
sudo /opt/scripts/idcs/create_idcs_cloudgate_cwallet.sh
sudo chown opc:opc /u01/data/cloudgate_config/cwallet.sso
sudo chmod 775 /u01/data/cloudgate_config/cwallet.sso
```

- Start the docker contained of the app gateway again:

```
sudo docker start appgateway
```

Now it will point to the primary App Gateway (that uses the primary Enterprise Application).

- Finally, you can delete the backup file, in case you took one

```
sudo rm /u01/data/cloudgate_config/cwallet.sso.bak
```

- Repeat the same in the rest of the secondary nodes.

To make sure that there are no more acceses to secondary Enterprise Application or App Gateway, you can deactivate them:

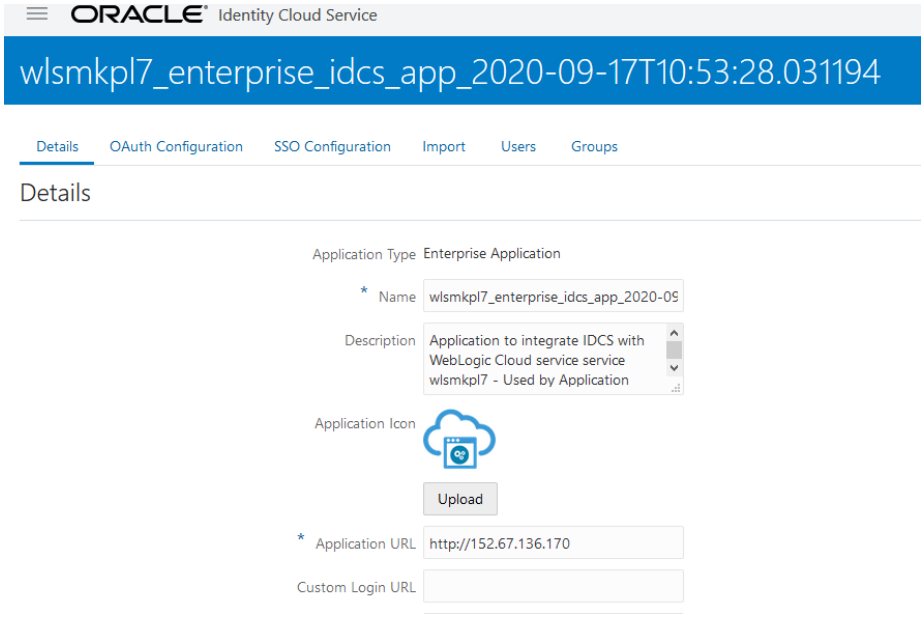
- Login into your IDCS console. For example: https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole
- Navigate to “Applications” and identify the **secondary** Enterprise Application.
Example: wlsmkpl7_enterprise_idcs_app_2020-09-11T11:55:44.241334
- Click in “Deactivate” to inactivate it
- Then, in the IDCS Console, navigate to Security > App Gateway and identify the **secondary** App Gateway
Example: wlsmkpl7_app_gateway_2020-09-187T11:55:44.241334
- Deactivate it.

NOTE: make sure you deactivate the secondary Enterprise Application and App Gateway and not the primary ones. The names are similar, but the timestamp will be later in the secondary ones.

Copyright © 2024, Oracle and/or its affiliates | Public

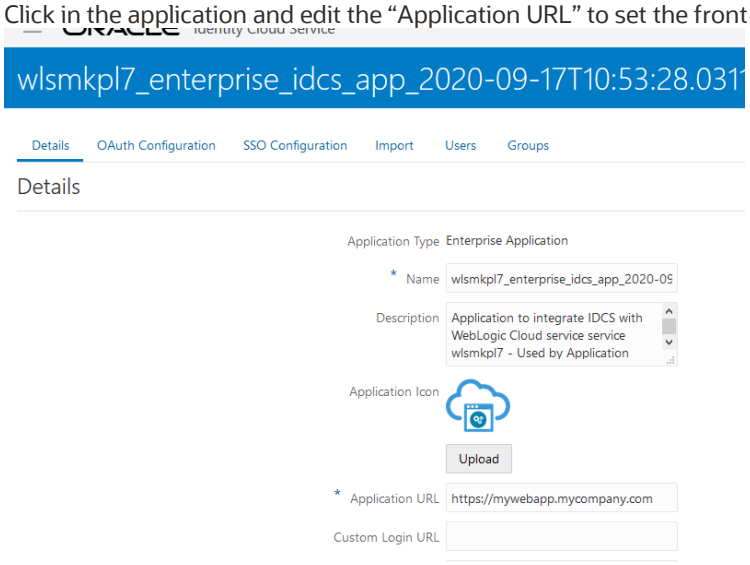
g) Configure the Enterprise application with the front-end name instead IPs

By default, the Enterprise application is configured with the primary IP as “Application URL”:



In case you have not done this before, configure it with the front-end name:

- Login into your IDCS console. For example: https://idcs-5e890de598888888888d70064c5e00718.identity.oraclecloud.com/ui/v1/adminconsole
- Navigate to “Applications” and identify the **primary** Enterprise Application. Example: wlsmkpl7_**enterprise_idcs_**app_2020-09-11T11:55:44.241334
- Click in the application and edit the “Application URL” to set the front-end name



h) Verify the IDCS protected URLS in the secondary location.

Verify that the secondary urls protected by IDCS are working properly in secondary location.

To run this verificatation, you can switchover to secondary location (as explained in Switchover) or alternatively, convert standby database to snapshot and start secondary servers (as explained in Open Secondary Site for validation).

Then, you can access to the IDCS protected sample application in secondary to verify that it works as expected:

https://<secondary_frontend_ip>/__protected/idcs-sample-app/

Example:

https://222.222.222.222/__protected/idcs-sample-app/

Copyright © 2024, Oracle and/or its affiliates | Public

You should be redirected to the IDCS login page, and once you provide the apropriate credentials, the IDCS sample application page will be shown.

Copyright © 2024, Oracle and/or its affiliates | Public

VALIDATE THE DR SETUP

After the DBFS, FSS or BV setup steps have been run, Oracle recommends that you immediately validate that the DR setup is correct by performing a complete **switchover** (see instructions for switchover in next pages). Alternatively, and to avoid downtime, you can **open the secondary site for validation** (see steps to open secondary site for validation in next pages). Make sure you follow the steps for the appropriate replica method of the system.

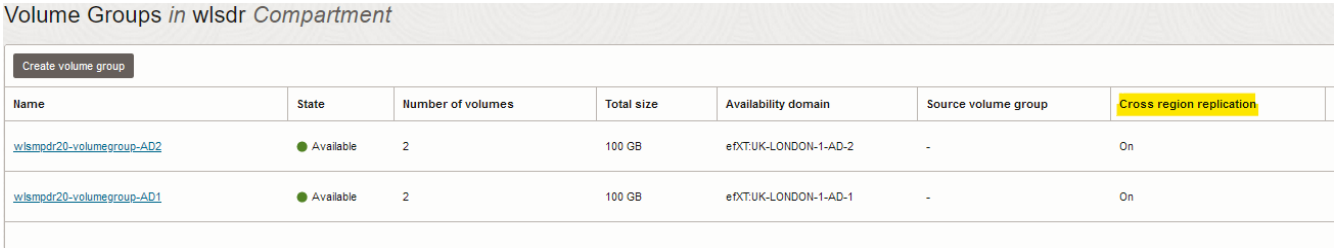
Copyright © 2024, Oracle and/or its affiliates | Public

LIFECYCLE PROCEDURES FOR BV REPLICA METHOD

Configuration Replication for BV Replica Method

The WebLogic Domain configuration is replicated to the standy database using the Block Volume cross-region replication feature. This replication is **automatic** and managed by the Oracle Cloud Infrastructure.

As described in the setup step Configure the Block Volume Cross-Region replication, the Block Volumes containing the WebLogic Domain configuration are grouped in a Volume Group (or more than one Volume Group when the compute instances are located in more than one AD). During the lifecycle of the system, make sure that these Volume Groups of the system with primary role have the cross-region replica enabled. Example:



Open Secondary Site for Validation for BV Replica Method

You can validate the standby site without performing a complete switchover, by converting the standby database to snapshot standby. This allows you to start the secondary WLS servers in the standby site and verify the secondary system. Any change performed in the standby site database while it is in snapshot standby mode will be discarded once it is converted to physical standby again, so primary data will not be affected by secondary site validations.

The steps to validate the standby site (Site2 in this case) without performing a switchover are the following:

	STEPS TO OPEN THE STANDBY SITE FOR VALIDATIONS	DETAILS
1	Activate the replicas in **Site2**	Until this point, the Block Volumes are being continously replicated from of Site1 to Site2. In order to mount them in Site2, you need to activate the replicas of Site 2. When you activate a BV replica, an "attachable" **BV is created as a clon from the replicated BV.** Then, you can attach these cloned BV to the compute instances. To activate the replicas in Site2, connect to OCI Console: - Go to **Site2, Block Storage > Volume Group Replicas** - Click in the Volume Group replica and "**Activate**" - For the Volume Group name, use the same name regardless the region where they are. For example, "<wlsociprefix>-BVGroup-AD1" - Repeat for all the Volume Group replicas in Site2.
2	Attach the replicated block volumes to mid-tier hosts in Site2	The attachable Block Volumes created as a result of the activation must be shown in Site2, in OCI Console > Storage > Block Volume To attach an activated Block Volume in Site2: -Attach the appropriate Block Volume to the host. Block Volume > click on the Block Volume > Attached Instances > Attach to Instance To simplify the procedure, select the check the flag "*use Oracle Cloud Agent to automatically connect to iSCSI-attached volumes*". The Cloud Agent will automatically run iSCSI commands, so you don't have to run them. To allow agent to run these commands on the compute instances, review requirements in https://docs.oracle.com/en-us/iaas/Content/Block/Tasks/enablingblockvolumemanagementplugin.htm#blockplugin-prereq__perms If you don't use the Oracle Cloud Agent, run the iSCSI commands manually. Click con "iSCSI Commands & Information" of the attached block volume and **run the iscsci commands** provided in "Commands for connecting" in the mid-tier host.

Copyright © 2024, Oracle and/or its affiliates | Public

		- Get the UUID of the new attached block volume: [root@wlsociprefix-wls-0 opc]# **sudo blkid** /dev/sda3: UUID="974147f5-d731-41de-bba8-56ff78ed1c9c" TYPE="xfs" PARTUUID="4a95c68a-bc70-4be9-bce8-b15e995fcf46" /dev/sda1: SEC_TYPE="msdos" UUID="593B-B893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="c5ac3089-6a91-40e0-bcc1-212ba0b43418" /dev/sda2: UUID="9ca12daa-d7ea-44a2-8680-5b676488b054" TYPE="swap" PARTUUID="682a63d1-d3ec-4019-b372-43720aaae717" /dev/sdb: UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4" **/dev/sdc: UUID="c293b5b5-005c-43e9-8c2f-02e873b76926" TYPE="ext4" ←** **NOTE:** it can take some time since it is attached until it appears in blkid command. - If it is not already, a**dd an entry for the apropriate UUID in /etc/fstab**, to mount and persist the mount after reboots UUID=c293b5b5-005c-43e9-8c2f-02e873b76926 /u01/data ext4 auto,defaults,_netdev,nofail NOTE: after the first switchover, the UUID of each replicated block volume will not change. You can comment the entry in /etc/fstab or keep it. However, Oracle recommends to keep it uncommented: the systemd daemon will automatically mount the block volume the next time it is attached. - If the appropriate entry already exists in the /etc/fstab when the device was attached**, the block volume is automatically mounted after being attached**. Otherwise, m**ount** the new attached block volume in the /u01/data and verify that it is correctly mounted. [root@wlsociprefix-wls-0 opc]# mount -a [root@wlsociprefix-wls-0 opc]# df -h| grep /u01/data /dev/sdb 49G 1.4G 46G 3% /u01/data Repeat the steps to attach all the activated block volumes. Optional: if mw block volumes are being replicated, repeat the steps for the mw block volume (and the mount point /u01/app) in the mid-tier host. Make sure you mount the proper UUID in the correct path.
3	Run the script that makes the replacements in Site2 mid-tier	Run the script **replacement_script_BVmodel.sh** in the Site2 administration host. Even if you replicate the data block volumes of all the nodes, you can run this script in the administration host only. The tnsnames.ora is under the DOMAIN_HOME/config folder, so the rest of the nodes will download the updated tnsnames.ora when the managed servers start.
4	Convert the standby DB into snapshot standby	Use DG broker in primary db host and convert the secondary to snapshot standby. As user oracle: [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database “secondary_db_unqname” to snapshot standby Use “show configuration” to verify that the conversion has been correctly performed.
5	Start the servers in the site2	Start the nodemanager in all the secondary servers. Example: $ cd $DOMAIN_HOME/bin/ $ nohup ./startNodeManager.sh > $DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 & Start the secondary admin server. Example :$ cd /u01/app/oracle/middleware/oracle_common/common/bin $ ./wlst.sh wlst> nmConnect ('weblogic','password','wlsociprefix-wls-0','5556','wlsociprefix_domain','/u01/data/domains/wlsociprefix_domain','SSL') wlst> nmStart('wlsociprefix_adminserver') Start secondary managed servers (use the secondary WebLogic Console or scripts). NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.

Copyright © 2024, Oracle and/or its affiliates | Public

6	Validate	This is not a swichover and the primary site is still active, so the virtual front-end name will resolve to the primary site's LBR IP address. Any browser access will, by default, be redirected to the active primary site. To directly access the secondary site's services, you must update the /etc/hosts file in a controlled client (laptop, etc.) and set the virtual front-end name to resolve to the secondary site's front-end LBR IP address. The run any validation from this client. NOTE: verify that the client used for validations does not access the system via an HTTP proxy, because the HTTP proxy may continue to resolve the virtual front-end name with the primary site's LBR IP address regardless of which name is in the /etc/hosts of the client. NOTE: Non-linux clients may require a reset of their local DNS cache before a browser will resolve the IP address using the customized host file entry.

Once you validate the secondary system, revert back to standby role:

	STEPS TO REVERT BACK STANDBY TO STANDBY ROLE	DETAILS
1	Stop processes in secondary Site2	You can connect to secondary WebLogic Console and shutdown managed servers and Admin server in secondary site. Stop the node manager processes too.
2	Convert the standby DB into a physical standby again	Use DG broker in primary db host and convert the secondary to physical standby again. As user oracle: [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to physical standby Use "show configuration" to verify that the conversion has been correctly performed.
3	Revert any updated /etc/hosts in clients	If you updated the virtual front-end name in a client's /etc/hosts file of a client, to point to secondary site, revert it back so the virtual front-end name points to primary front-end IP again.
4	Detach Block Volumes in Site2	For all the replicated block volumes, run the following: - Unmount the replicated Block Volumes: [opc@wlsociprefix-wls-0 opc]# sudo umount /u01/data Make sure that there are not Oracle processes running. It is expected that they are stopped at this point, but if there is something still running on that folder, the umount will fail. - Once umounted, detach the Block Volumes from the OCI Console. Go to each **block volume > attached instances > detach from instance** If you didn't use the Oracle Cloud Agent to attach the block volumes, the OCI Console **will ask you to run some iscsci commands** before completing the detachment. Repeat these steps for the rest of the replicated block volumes in Secondary.
5	Delete/rename the detached volumes in Site2 to prevent from mouting them by mistake.	Using the OCI Console, delete (or rename) the Block Volumes that have been detached from the Site2 mid-tier hosts in the previous step. Also delete the Volume Groups. They will not be used anymore.

Copyright © 2024, Oracle and/or its affiliates | Public

Switchover for BV Replica Method

A switchover is a planned operation where an administrator reverts the roles of the two sites. Primary becomes standby and standby becomes primary. A manual switchover requires a number of operations that may increase the RTO and the overall operational overhead. You can use Oracle Full Stack DR service to automate most of these tasks and simplify the required operations. Refer to the Learn About Using OCI Full Stack Disaster Recovery Service with Oracle WebLogic Server Domains for details on this.

To perform a **manual switchover from Site1 to Site2** in a WLS for OCI DR configuration based on Block Volume Cross-Region replication, follow these steps:

a) Pre-Switchover tasks:

These steps do not cause downtime.

	PRE-SWITCHOVER STEP	DETAILS
1	Detach previously used block volumes in **Site2**	If still attached, detach original or previously used block volumes from the mid-tier hosts in Site2 (umount, detach). Detach only the volumes that are replicated from primary. Do not detach the mw blocks if you are not replicating them. Then, delete or rename the detached volumes and volume groups in Site2 to prevent from mounting them by mistake. They will not be used anymore.
2	Activate the replicas in **Site2**	Until this point, the Block Volumes are being continously replicated from of Site1 to Site2. In order to mount them in Site2, the replicas of Site 2 need to be activated. When you activate a BV replica, an "attachable" **BV is created as a clon from the replicated BV.** Then, you can attach these cloned BV to the compute instances. To activate the replicas in Site2, connect to OCI Console: - Go to **Site2, Block Storage > Volume Group Replicas** - Click in the Volume Group replica and "**Activate**" - For the Volume Group name, use the same name regardless the region where they are. For example, "<wlsociprefix>-BVGroup-AD1" - Repeat for all the Volume Group replicas in Site2.
3	Attach the replicated BV to **Site2** mid-tier hosts	The attachable Block Volumes created because of the activation must be in Site2, listed in Block Volumes. To attach an activated Block Volume in Site2: - In OCI Console, Block Storage > Block Volume > click on the Block Volume > Attached Instances > Attach to Instance To simplify the procedure, select the check the flag "***use Oracle Cloud Agent to automatically connect to iSCSI-attached volumes*".** The Cloud Agent will automatically run iSCSI commands, so you don't have to run them. To allow the agent to run these commands on the compute instances, review requirements in https://docs.oracle.com/en-us/iaas/Content/Block/Tasks/enablingblockvolumemanagementplugin.htm#blockplugin-prereq__perms If you don't use the Oracle Cloud Agent, run the iSCSI commands manually. Click con "iSCSI Commands & Information" of the attached block volume and **run the iscsci commands** provided in "Commands for connecting" in the mid-tier host. - **Get the UUID** of the new attached block volume: [root@wlsociprefix-wls-0 opc]# **sudo blkid** /dev/sda3: UUID="974147f5-d731-41de-bba8-56ff78ed1c9c" TYPE="xfs" PARTUUID="4a95c68a-bc70-4be9-bce8-b15e995fcf46" /dev/sda1: SEC_TYPE="msdos" UUID="593B-B893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="c5ac3089-6a91-40e0-bcc1-212ba0b43418" /dev/sda2: UUID="9ca12daa-d7ea-44a2-8680-5b676488b054" TYPE="swap" PARTUUID="682a63d1-d3ec-4019-b372-43720aaae717" /dev/sdb: UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4" **/dev/sdc: UUID="c293b5b5-005c-43e9-8c2f-02e873b76926" TYPE="ext4"** ← **NOTE:** it can take some time since it is attached until it appears in blkid command.

Copyright © 2024, Oracle and/or its affiliates | Public

		- Add an entry for the apropriate UUID **in /etc/fstab**, to mount and persist the mount after reboots. Example: UUID=c293b5b5-005c-43e9-8c2f-02e873b76926 /u01/data ext4 auto,defaults,_netdev,nofail **NOTE:** after the first switchover, the UUID of each replicated block volume will not change. In subsequents switchovers, Oracle recommends keeping it uncommented: the systemd daemon will automatically mount the block volume the next time it is attached. -When the appropriate entry already exists in the /etc/fstab when the device was attached**, the block volume is automatically mounted after being attached**. Otherwise, m**ount** the new attached block volume in the /u01/data and verify that it is correctly mounted. [root@wlsociprefix-wls-0 opc]# mount -a [root@wlsociprefix-wls-0 opc]# df -h| grep /u01/data /dev/sdb 49G 1.4G 46G 3% /u01/data Optional: if mw block volumes are being replicated, repeat the steps for the mw block volume (and the mount point /u01/app) in the mid-tier host. Make sure you mount the proper UUID in the correct path. Repeat the steps all the activated Block Volumes in Site2.
4	Run the replacement script in Site 2	Run the script **replacement_script_BVmodel.sh** in the Site2 administrator host. Even if you replicate the data block volumes of all the nodes, you can run this script in the administration host only. The tnsnames.ora is under the DOMAIN_HOME/config folder, so the rest of the nodes will download the updated tnsnames.ora when the managed servers start.

Copyright © 2024, Oracle and/or its affiliates | Public

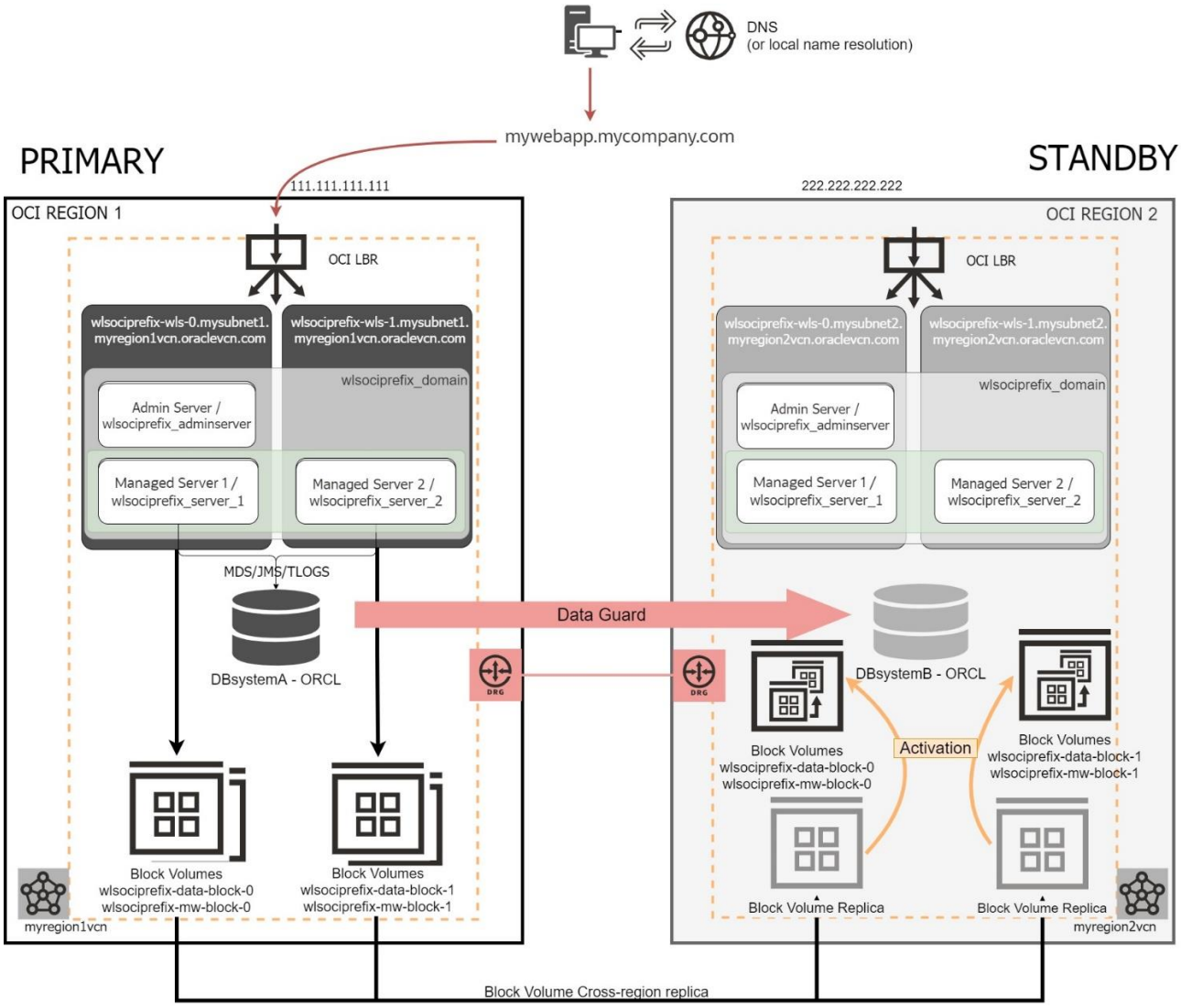


Figure 11 During the pre-switchover operations, the Block Volume Replicas are activated in the standby site.

b) Switchover

The actual switchover procedure starts at this point:

	SWITCHOVER STEP	DETAILS
1	Stop processes in primary Site	Stop WebLogic Administration Server, managed servers and node managers in Site1. NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.
2	Switchover Virtual Front-end DNS name	Perform the required DNS push in the DNS server hosting the names used by the system or alter the file host resolution in clients to point the front-end virtual address of the system to the public IP used by LBR in site 2.
3	Switchover Database	Use DG broker in primary db host to perform the switchover. As user oracle: [oracle@drdba ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> switchover to “secondary_db_unqname”

Copyright © 2024, Oracle and/or its affiliates | Public

4	Start the servers in Site2 (new primary)	Start the nodemanager in all the secondary servers .Example: $ cd $DOMAIN_HOME/bin/ $ nohup ./startNodeManager.sh > $DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 & Start the secondary admin server. Example: $ cd /u01/app/oracle/middleware/oracle_common/common/bin $ ./wlst.sh wlst> nmConnect ('weblogic','password','wlsociprefix-wls-0','5556',' wlsociprefix_domain','/u01/data/domains/ wlsociprefix_domain','SSL') wlst> nmStart(' wlsociprefix_adminserver') Start secondary managed servers (use the secondary WebLogic Console or scripts) NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.

c) **Post-Switchover tasks:**
At this point, the services are active in the Site2, hence, no additional downtime is required. However, there are additional tasks needed to complete the switchover procedure and leave the system in the appropriate role-reversed state. Oracle recommends running them immediately as following:

- Enable the Block Volume Replication in the other way, in the Volume Groups in Site2 (new primary). Make sure you provide the appropriate Availability Domain for the replicas.
- Disable the replica in the Site1 (new standby) Volumes groups.
- Unmount the block volumes in the Site1 (new standby) that are replicated from the new primary.
- Detach the unmounted block volumes from Site1 mid-tier hosts to prepare them for the future. If you used Oracle Cloud Agent to attach the block volume, the agent runs the iSCSI commands to log off the iSCSI targets.
- You can comment the entry in /etc/fstab or keep it. Oracle recommends keeping it: the systemd daemon will automatically mount the block volume the next time it is attached.
- Delete or rename the detached volumes from the Site1 mid-tier hosts to prevent from mounting them by mistake. Also delete the unused Volume groups in Site1. They will not be used anymore.

Copyright © 2024, Oracle and/or its affiliates | Public

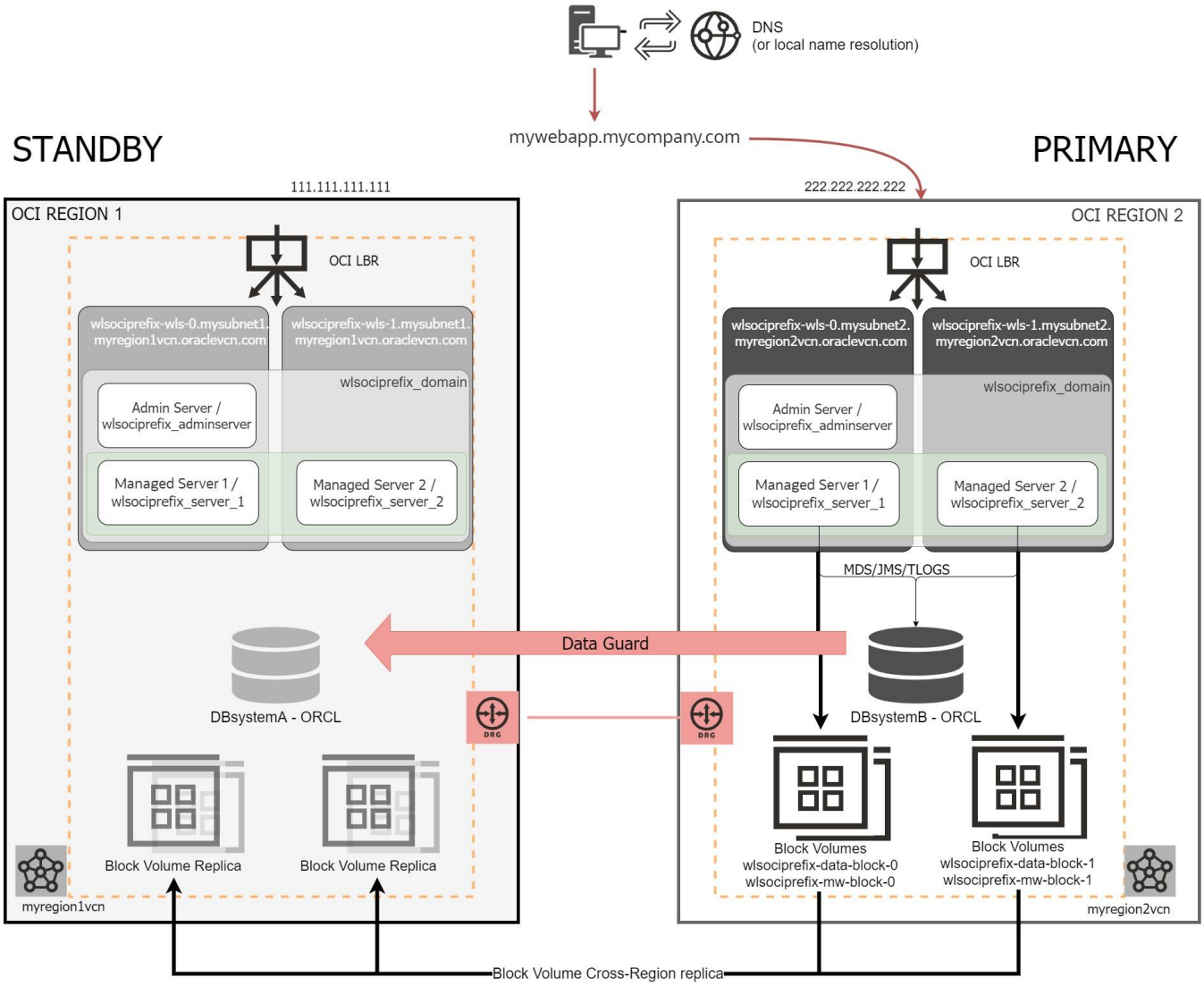


Figure 12 After the switchover, there is a post step to enable the Block Volume replica in the other way

Failover for BV Replica Method

A failover operation is performed when the primary site becomes unavailable, and it is commonly an unplanned operation. You can role-transition a standby database to a primary database when the original primary database fails and there is no possibility of recovering the primary database in a timely manner. There may or may not be data loss depending upon whether your primary and target standby databases were consistent at the time of the primary database failure.

A manual failover requires a number of operations that may increase the RTO and the overall operational overhead. You can use Oracle Full Stack DR service to automate most of these tasks and simplify the required operations. Refer to the Learn About Using OCI Full Stack Disaster Recovery Service with Oracle WebLogic Server Domains for details on this.

Manual failover steps are the same as in a switchover, only that the pre-switchover tasks listed above need to be added to the total RTO (there is no possible preparation for the activation and attachment of BVs before an unplanned operation).

The other difference is how the Database role change is performed. In a failover, you need to connect to the standby DB and run the failover command instead of the switchover command:

```
[oracle@drdbb ~]$ dgmgrl sys/your_sys_password@secondary_db_unqname
DGMGRL> failover to “secondary_db_unqname”
```

The rest of the steps are the same than in a switchover, including pre and post steps.

Normally, a failover operation is executed when an outage affects the primary region. Hence, there may be some tasks that you can’t perform in primary. For example, you may not be able to stop the WLS processes in primary because the hosts

Copyright © 2024, Oracle and/or its affiliates | Public

are unreachable. So, once a failover operation is finished, and the previous primary site is reachable again, you must perform some manual tasks to prepare the system for a future switchback. These tasks are:

- **Stop the WebLogic processes** in the failed site. If you didn't stop them during the failover, the processes may be hung. Make sure they are stopped.
- Perform a **Data Guard reinstate operation**. After a failover, the failed primary shows as "Disabled Standby". During the reinstate operation, the database in failed site will be flashbacked and converted as physical standby database. Perform this operation using the OCI Console preferable, so the status is updated in the OCI Console accordingly.
- If not done during the failover operation, perform the **post steps** described in the switchover operation to **reverse cross-region replication** and to **unmount, detach and terminate** the Volume Groups and Block Volumes in the failed site.

Scale-out and Scale-in for BV Replica Method

You can scale-out and scale-in a WLS for OCI system following the steps described in the WLS for OCI documentation Scale a Stack.

When you perform a scale-out or scale-in a WLS for OCI DR environment, there are some characteristics specific to a DR environment that must be considered: there are 2 WLS for OCI systems (primary and secondary) and the domain configuration in secondary is a copy of the primary configuration, so it uses primary hostnames as listen-addresses.

When the listen-address hostnames are added as aliases in the midtier's /etc/hosts, the new nodes provisioned during a scale-out operation do not include these aliases in its /etc/hosts file by default. This can cause the scale-out procedure to fail in the secondary location, because the new nodes cannot connect to WebLogic administration server. To avoid this problem during scale-out of the WLS for OCI DR environment, required steps are documented in this point.

When you added the primary hostnames entries to a DNS private view in secondary, as described in https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_dr, the scale-out procedures are simplified, because any new node is able to resolve the primary hostnames as soon as it is created.

See the following points for detailed steps.

Scale-out

NOTE: To scale-out environments that are using the TNS alias approach in the datasources, the WLS for OCI instance must have the patch 34988073. The WLS for OCI instances with PSU 12.2.1.4230328 or newer have this patch included.

This is the procedure to **scale-out** a WLS for OCI DR environment **in Block Volume cross-region replica model**:

a) First, before proceeding with any scale-out, follow the steps described in point Open Secondary Site for Validation for BV Replica Method to open the secondary site, but do not convert the standby database to snapshot yet, and do not start the admin and managed servers yet. This is just to mount in secondary hosts a version of the block volumes prior to any scale action, so the scale action can be performed in primary and secondary independently.

b) Now you can **scale-out primary** WLS for OCI stack:
   1. Follow the steps described in Add or Remove WebLogic Server Nodes in the primary stack by increasing the number of the nodes (in this example it is assumed that the number is increased by 1 node).
   2. Once the scale-out has finished correctly, connect with ssh to **the new node** and:
      a. Edit /etc/hosts to add the front-end FQDN with primary front-end LBR IP address. Example:

```
# Front-end virtual name for DR, poiting to primary front-end IP
111.111.111.111  mywebapp.mycompany.com
```

      b. (**Not needed if you are using the DNS private view approach for hostname aliases**)
      Edit /etc/hosts in the new node and add the aliases that already exist in the rest of primary nodes, that include secondary names. Example:

```
10.0.0.82 <prim_midtier1_fqdn> <prim_midtier1_hostname> <sec_midtier1_fqdn> <sec_midtier1_hostname>
10.0.0.81 <prim_midtier2_fqdn> <prim_midtier2_hostname> <sec_midtier2_fqdn> <sec_midtier2_hostname>
```


Copyright © 2024, Oracle and/or its affiliates | Public

c. (**Not needed if you are using the DNS private view approach for hostname aliases**)
Edit the /etc/oci-hostname.conf and set PRESERVE_HOSTINFO to 3 so these changes are persisted across reboots.

3. Restart the new managed server.

NOTE: The terraform scripts used by WLS for OCI to scale-out create redundant/unnecessary block volumes in the existing nodes. These duplicated block volumes have the same names than the existing block volumes, and they are attached but not mounted to the nodes.These duplicated block volumes are NOT needed and it is strongly recommended to detach and delete them immediately after the scale-out, to prevent mistakes and mounting the incorrect BV. Make sure you delete the duplicated block volumes created by the scale-out job both.

c) **Scale-out secondary** WLS for OCI stack:
Scaling-out the secondary requires intervention before the scale-out, specially when you have added the primary address aliases t othe /etc/hosts instead than to a DNS private view. Detailed steps explained here:

1. Convert the standby database into snapshot standby.
2. (**Not needed if you are using the DNS private view approach for hostname aliases**)
Remember that the WebLogic domain configuration in the standby is a copy from primary and it uses the primary hostnames as listen addresses for the servers. The new node that is added to secondary when scaling-out will not be aware of them (aliases of the primary names are not included by default in the /etc/hosts file of the new node). To allow the scale-out in the secondary to finish successfully, previous to proceed with the scale-out, modify the listen-addresses in the secondary domain and set there the secondary midtier hostnames. This makes that the scale-out procedure run without issues. Steps:
   - Identify **primary midtier hosts FQDN** (the existing nodes previous to the scale-out). Example:

     ```
     wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
     wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
     ```

     Primary midtier1 fqdn is *wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com*, and its hostname is *wlsociprefix-wls-0*.
     Primary midtier2 fqdn is *wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com* and its hostname is *wlsociprefix-wls-1*.

   - Identify **secondary midtier hosts FQDN** (the current existing nodes). Example:

     ```
     wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
     wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
     ```

     NOTE: hostnames are expected to be the same in primary and secondary midtier hosts, only the fqdn values will differ.

   - In the secondary site's admin server node, replace primary instance's FQDN with the secondary instance's FQDN in the <DOMAIN_HOME>/config/config.xml file:

     ```
     cd <DOMAIN_HOME>/config/
     cp config.xml config.xml_backup_pre_scale-out
     sed -i 's/primary_midtier1_fqdn/secondary_midtier1_fqdn/g' config.xml
     sed -i 's/primary_midtier2_fqdn /secondary_midtier2_fqdn/g' config.xml
     ```

     Example:

     ```
     sed -i 's/ wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /g' config.xml

     sed -i 's/ wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /g' config.xml
     ```

3. Start admin and managed servers in the secondary site.
4. Follow the steps described in Add or Remove WebLogic Server Nodes in secondary stack to add a node.
5. Once the scale-out process finishes, add the required aliases in the new added node:
   - Edit /etc/hosts in the new node and add the front-end FQDN for the secondary front-end LBR IP address, as it is in the rest of the secondary nodes.

Copyright © 2024, Oracle and/or its affiliates | Public

```
# Front-end virtual name for DR, poiting to secondary front-end IP
222.222.222.222  mywebapp.mycompany.com
```

- (**Not needed if you are using the DNS private view approach for hostname aliases**)
  Edit /etc/hosts in the new node and add the existing aliases that secondary midtier nodes already have, where the primary node FQDN are aliases of the secondary local IP addresses.

```
10.2.0.12  <sec_midtier1_fqdn> <sec_midtier1_hostname>  <prim_midtier1_fqdn> <prim_midtier1_hostname>
10.2.0.11  <sec_midtier2_fqdn> <sec_midtier2_hostname>  <prim_midtier2_fqdn> <prim_midtier2_hostname>
```

  Edit the /etc/oci-hostname.conf and set PRESERVE_HOSTINFO to 3 so these changes are persisted across reboots.

6. (Only if IDCS authentication is used) In the new added node, run the steps explained in 10b (*Configure the primary Enterprise application and App Gateway in the secondary WebLogic domain*) of Complete DR configuration when IDCS authentication is used to point its Cloudgate agent to the same App Gateway (and Enterprise Application) than primary, as it was done for the rest of the secondary nodes during the DR setup.
7. Stop servers in secondary site (managed servers and admin).
8. Convert the standby database to **physical standby.**
9. (**Not needed if you are using the DNS private view approach for hostname aliases**)
   Optionally, you can revert the change done in step 2 and set again the primary FQDN in the listen addresses of the domain configuration. Alternative, this will be automatically overridden later when you replicate the configuration from primary.

d) **Once both primary and standby are scaled out,** complete configuration by adding the aliases for the new nodes to all the midtier hosts (existing and new nodes).
   If you are using the /etc/hosts approach for the hostname aliases:
   1. In primary, add it to ALL the existing primary midtier nodes (and also in the new one). Example (this must be in a single line):

```
<primary_newnode_IP>  <primary_newnode_fqdn> <primary_ newnode _hostname>  <secondary_ newnode _fqdn>
<secondary_ newnode_hostname>
```

   2. In secondary, add it to ALL the existing midtier nodes (and also in the new one). Example (this mus be in a single line):

```
<secondary_newnode_IP> <secondary_newnode_fqdn> <secondary_ newnode _hostname>  <primary_ newnode _fqdn>
<primary_ newnode_hostname>
```

   If you are using the DNS private view approach:
   3. Add the hostnames of the new nodes to the appropriate DNS private views instead of adding them to the /etc/hosts. I.e.: the name of the new secondary node to the primary private view, and the name of the new primary node to the secondary private view.

e) Then, revert the secondary to the standby role, by detaching the volumes, as described in the point Open Secondary Site for Validation for BV Replica Method.
f) For the Block Volume of the new node in primary, enable the Cross-Region replica in the same way as it is already configured for the existing primary nodes by adding it to the appropriate Volume Group

NOTE: The terraform scripts used by WLS for OCI to scale-out create redundant/unnecessary block volumes in the existing nodes. These duplicated block volumes have the same names than the existing block volumes, and they are attached but not mounted to the nodes.These duplicated block volumes are NOT needed and it is strongly recommended to detach and delete them immediately after the scale-out, to prevent mistakes and mounting the incorrect BV. Make sure you delete the duplicated block volumes created by the scale-out job both.

Scale-in

This is the procedure to **scale-in** a WLS for OCI DR environment **in Block Volume cross-region replica model**:

a) First, follow the steps described in previous point Open Secondary Site for Validation for BV Replica Method to open the secondary site, but do not convert the standby database to snapshot yet, and do not start the admin and managed servers yet. This is just to mount in secondary hosts a version of the block volumes prior to any scale action, so the scale action can be performed in primary and secondary independently.

Copyright © 2024, Oracle and/or its affiliates | Public

b) Disable the Cross-Region replica in the block volumes of the primary node that is going to be deleted, by removing the Block Volume from the Volume Group. The scale-in job will fail to delete a block volume that has the cross-region replica enabled.
c) **Scale-in primary** WLS for OCI stack:
    1. Follow the steps described in Add or Remove WebLogic Server Nodes to reduce the number of servers.
d) **Scale-in secondary** WLS for OCI stackstance:
    2. Convert the standby database into snapshot standby.
    3. Start the admin server only (starting managed servers is not required).
    4. Follow the steps described in Add or Remove WebLogic Server Nodes to reduce the number of servers.
    5. Once finished, stop any running WebLogic process, and convert secondary database to physical standby.
e) Remove the aliases of the deleted node from the /etc/hosts in primary and secondary midtier hosts, or from the DNS private views if you are using that approach.
f) Then, revert the secondary to the standby role, by detaching the volumes, as described in the previous point Open Secondary Site for Validation for BV Replica Method.

NOTE: The terraform scripts used by WLS for OCI to scale-in create redundant/unnecessary block volumes in the existing nodes. These duplicated block volumes have the same names than the existing block volumes, and they are attached but not mounted to the nodes.These duplicated block volumes are NOT needed and it is strongly recommended to detach and delete them immediately after the scale-in to prevent mistakes and mounts pointing to the incorrect BV. Make sure you delete the duplicated block volumes created by the scale-in job.

Copyright © 2024, Oracle and/or its affiliates | Public

LIFECYCLE PROCEDURES FOR DBFS AND FSS WITH RSYNC METHODS

Configuration Replication

As explained in previous sections, any data that resides in the database is automatically replicated to the standby site via the Data Guard.

But **most of the configuration of a WebLogic domain resides in the WebLogic domain folder files**. You can automate this replication process (based on DBFS and FSS) using the script config_replica.sh. It replicates the WebLogic configuration from primary to standby via DBFS or FSS with rysnc, depending on the method choosen for the DR topology. The same script is used in primary and standby, and it is valid for both DBFS and FSS. The script contains logic to determine which site is acting as primary and which site is acting as secondary to perform the replication in the right direction.

Option 1) DBFS method

In this approach, the DBFS file system created in step Configure the DBFS staging mount in DBFS method is used as **an assistance file system** to store a copy of the primary site's domain configuration.

NOTE: The WebLogic Server domain configuration cannot reside directly on the DBFS mount because that would make the middle tier dependent on the DBFS infrastructure (on the database, on FUSE libraries, mount points, etc.) to come up.

The information in this filesystem is automatically replicated to standby location via Data Guard. In the standby site, the system can be also mounted, although it is not available unless the standby database is open in read-only mode (when Active Data Guard is used), or when the database is converted to snapshot standby.
The steps of this procedure are as follows:

- The **primary WebLogic domain configuration directory contents are copied to the primary DBFS** file system. Files and folders that are irrelevant or not required (i.e: tmp, tnsadmin folders) are excluded.
- The files copied into the DBFS, as they are stored in the database, are **automatically transferred to the standby database via Data Guard.**
- In the standby site, the **database is converted to snapshot standby** and the DBFS mount is mounted in standby midtier hosts.
- In the standby site, the WLS domain configuration files are copied **from the DBFS mount to the standby domain** folder.

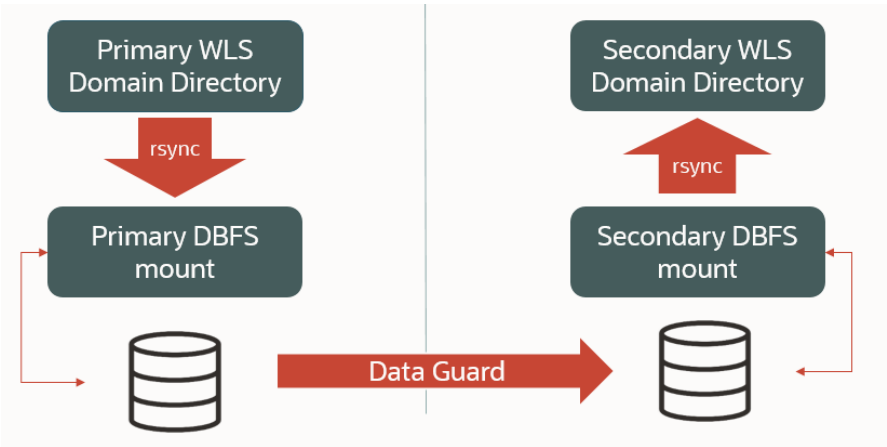


Figure 13 Replicating WebLogic domain configuration to standby WebLogic Cloud with DBFS based method.

Copyright © 2024, Oracle and/or its affiliates | Public

NOTE: The midtier mounts the dbfs mount by connecting to the local pdb database with a tns alias. This alias is in the $DOMAIN_HOME/dbfs/tnsnames.ora file. This alias is created with a retry parametrization, so in case that there is an issue in connecting to the database during the copy from or to the dbfs mount, these retries will help. The values configured by default (total time of 10 mins, to support a minimum db host reboot) can be adjusted or reduced to meet your specific requirements, if needed. Note that operating system commands that retrieve info from the dbfs filesystem (like "df -h", or a "ls" in the dbfs mount folders) may take long periods of time to return due to the retries, if the PDB's service is not reachable.

Option 2) FSS with rsync method

This method uses rsync to replicate the primary site WLS Domain configuration to the secondary site on a regular basis. The steps of this procedure are as follows:

- On the primary site, the domain configuration is copied to the local FSS filesystem. Then, the content of the primary FSS filesystem is copied to the remote site's FSS filesystem.
- On the secondary site, the domain configuration is copied from the local FSS filesystem to the WLS domain directory.

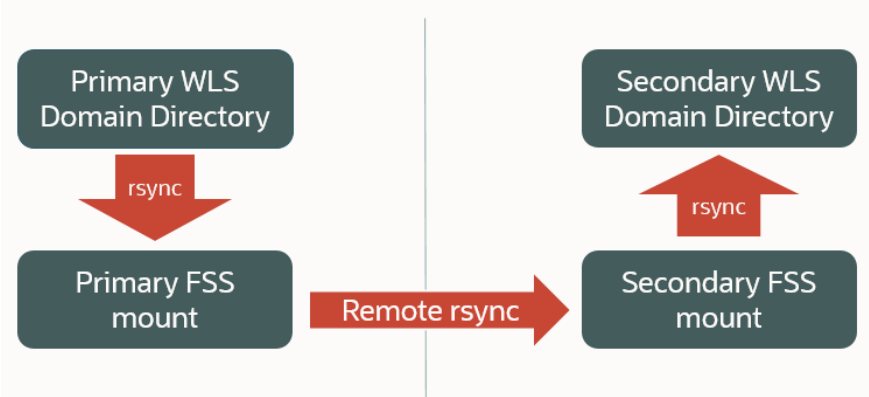


Figure 14 Replicating WebLogic domain configuration with FSS with rsync method.

Follow these steps to use the **config_replica.sh** script to replicate the WebLogic configuration:

1. The script has these communication requirements:
    a. In DBFS method: it requires access from each WebLogic Administration host to the remote Database listener port to perform db role changes. Because when the script runs in standby role, it converts the standby database to snapshot to mount the dbfs mount.
    b. In RSYNC method: it requires SSH access from each WebLogic Administration host to the remote WebLogic Administration host (ssh port). It connects to the remote Administration host to perform the remote rsync copy.

    Make sure you create the appropriate rules to allow this communication. This communication can be done through public IPs (in case that Internet Gateway is used for the connectivity between the sites), or through internal IPs (in case that the sites are connected via **Dynamic Routing Gateway**, which is the **recommended** approach).
2. If not already done, **upload** the **config_replica.sh** script to the **primary WebLogic Administration host** and to **the secondary WebLogic Administration host**.Make sure you place it in the same folder than the rest of the scripts and that all of them have execution permission.
3. In **primary** WebLogic Administration host**, open** the config_replica.sh script. **Edit the customizable parameters** sections. Make sure you provide the appropriate variables for the primary.
4. In **secondary** WebLogic Administration host, **open** the config_replica.sh script. Edit **the customizable parameters** sections. Make sure you provide the appropriate variables for the secondary.
5. **Execute** the config_replica.sh script **first in the primary** WebLogic Administration host (with oracle user). Monitor the execution and watch for any errors. The script will verify the current site role and will copy the domain configuration from the primary WebLogic domain to the secondary site (via DBFS or via FSS with rsync method).
6. Once it completes, **execute the config_replica.sh** script in **the secondary** site's WebLogic Administration Server host (with oracle user). Ensure you use the appropriate values in the customized parameters. The script will verify

Copyright © 2024, Oracle and/or its affiliates | Public

the database role. As it is the **standby**, it will copy the domain configuration from the secondary staging filesystem to the secondary WebLogic domain.

The secondary WebLogic Administration server is normally stopped when the changes are replicated. The changes will take effect next time it is started (during the switchover, failover or when opening secondary site for validations).
In case the WebLogic Administration server is up in secondary location, you need to restart it for the changes to take effect.
Note that to start the secondary WebLogic Admin server, it is required to have the secondary DB in snapshot standby mode or use Active Data Guard. Oracle recommends keeping the standby WebLogic Administration server stopped to avoid running stale configuration. Start it only when you are validating the standby site or during the switchover or failover procedure.

NOTE: The configuration under domain_home/config is automatically copied over to all other nodes that are part of the WebLogic domain when the managed servers are restarted and connect to the Administration Server. Any other configuration that resides out of the domain_home/config directory will be copied ONLY to the first node and will have to be manually replicated to each of the managed servers nodes. This includes any customizations to start scripts under domain_home/bin domain_home/security, etc.

Furthermore, the script only transfer changes for files under the domain. Any data or files that are created OUTSIDE the domain directory in the Weblogic Administration Server node, are not taken care of by the config_replica.sh script and need to be synchronized separately.

NOTE: For application deployment operations, Oracle recommends to use the WebLogic deployment "Upload your files" option in the WebLogic Administration Console. This way, the deployed files are placed under the upload directory of the Administration Server (under domain directory/servers/admin_server_name/upload) and will be synced to standby by the config replica script.

Once this initial execution in primary and secondary is complete, the scripts can be added to the cron list in the system (or to any other scheduling tool used by the customer) so that they are executed regularly and/or after a configuration change in primary system. The script must always be run both in primary and standby**, first in the primary WebLogic Administration host** (to copy the domain config to the staging folder) and **then in the standby WebLogic Administration host** (to copy the domain config copy from the staging to the domain folder).

When there is a role change, the script automatically adapts the execution to the new role, because it checks the actual role of the site in order to take one action or other.

Notice that "croning" the copy script automates synchronization but also has the following implications:

- Synchronization may incur in latency as high as the frequency of the cron jobs in both locations added up. I.e., if the cron jobs are set to execute every 30 minutes each, the changes may take 60 minutes to be available if the window in primary overlaps with the one on the secondary location. Before you perform a switchover, make sure that this amount of time has passed by after the last configuration change. Otherwise, you could switchover before the change is present on standby and overwrite the changes originally applied with the role switch.
- The cron frequency should be set at minimum to the largest amount of time than the execution of config_replica.sh may take. Otherwise, copy jobs may overlap.

Validation of Configuration Replication

The script config_replica.sh is role dependant. It checks the current role of the site by gathering the role of the local database. If the site has primary role, it copies the content of the domain configuration to the staging folder (DBFS or FSS). If the site has the standby role, it copies the content from the staging folder (DBFS or FSS) to the domain folder. For a complete replication from primary to standby, the script must be always run in both sites: first in the site with primary role, second in the site with standby role.

Make sure you verify that the configuration replica works also after a switchover or a failover. Follow these steps for a complete verification of the config replication:

1) First, validate config replication from primary to secondary. This does not incur in primary's downtime:
   a) Do some configuration change in primary WebLogic domain. For example, increase a connection pool size in one datasoure or apply any other non-intrusive change in primary WebLogic.

Copyright © 2024, Oracle and/or its affiliates | Public

b) Use the config_replica.sh script to replicate the configuration from primary to secondary. This is a two steps process. Run config_replica.sh first in the primary WebLogic Administration host, and then run config_replica.sh in standby WebLogic Administration host.
c) Verify that the configuration change applied in primary is present in the secondary domain directory.
d) Convert the standby database to snapshot standby, as described in Open Secondary Site for Validation.
e) Start the WebLogic Administration Server in secondary midtier to validate that the configuration is correct.
f) Stop the WebLogic Administration Server in secondary.
g) Revert the standby database from snapshot standby back to physical standby.
h) You can revert the config change in primary; it was done just for testing purposes.

2) Validate config replication in the opposite direction (after a switchover or failover to secondary)
This validation incurs in primary's downtime because it requires a switchover. Perform these steps in some maintenance window.
a) Do a switchover of the complete system to secondary, as described in the point Switchover for DBFS and FSS with rsync Methods.
b) Perform some configuration change in the new primary Weblogic domain (old standby).
c) Use the config_replica.sh script to replicate config from the new primary to the new standby: run the config_replica.sh in the new primary administration host and then run the script in the new standby administration host.
d) Verify that the WebLogic configuration has been properly replicated to the new standby site.
e) Switchback the system to revert to the original status.

Open Secondary Site for Validation

You can validate the standby site without performing a complete switchover, by converting the standby database to snapshot standby. This allows the secondary WLS servers to start in the standby site, so you can run validations in secondary. Any change performed in the standby site database while it is in snapshot standby mode will be discarded once it is converted to physical standby again, so primary data will not be affected by secondary validations.

The steps to validate the standby site without performing a switchover are the following:

	STEPS TO OPEN THE STANDBY SITE FOR VALIDATIONS	DETAILS
1	Convert the standby DB into snapshot standby	Use DG broker in primary db host and convert the secondary db to snapshot standby. As user oracle: [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to snapshot standby Use "show configuration" to verify that the conversion has been correctly performed.
2	Start the servers in secondary site	Start the secondary admin server. Example: cd /u01/app/oracle/middleware/oracle_common/common/bin ./wlst.sh wlst> nmConnect ('weblogic','password','wlsociprefix-wls-0','5556','wlsociprefix_domain','/u01/data/domains/wlsociprefix_domain','SSL') wlst> nmStart('wlsociprefix_adminserver') Start secondary managed servers (use the secondary WebLogic Console or scripts). NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.
3	Validate	As this is not a switchover, the primary is still active. The front-end name will be pointing to primary site's LBR IP, so any browser access will be redirected to primary by default. To access directly to secondary WLS system, you can update the /etc/hosts file in a controlled client (laptop, etc.) and set the virtual front-end name resolved to the secondary site's front-end IP. Then run any validation from this client.

Copyright © 2024, Oracle and/or its affiliates | Public

		NOTE: verify that the client used for validations does not access to the WLS for OCI system via an HTTP proxy, because the HTTP proxy may continue to resolve the virtual front-end name with the primary site's LBR IP address regardless which name is in the /etc/hosts of the client. NOTE: Non-linux clients may require a reset of their local DNS cache before a browser will resolve the IP address with the customized host file entry.

Once you have finished validations on the secondary site, follow these steps to revert it back to standby role again:

	STEPS TO REVERT BACK STANDBY TO STANDBY ROLE	DETAILS
1	Stop managed servers and admin servers in secondary	You can connect to secondary WebLogic Console and shutdown the administration and managed servers in secondary site.
2	Convert the standby DB into a physical standby again	Use DG broker in primary db host and convert the secondary to physical standby again. As user oracle: [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to physical standby Use "show configuration" to verify that the conversion has been correctly performed.
3	Revert any updated in client's /etc/hosts	If you updated the virtual front-end name in the /etc/hosts file of a client, to point to secondary site, revert it back so the virtual front-end name points to primary front-end IP again.

Switchover for DBFS and FSS with rsync Methods

A switchover is a planned operation where an administrator reverts the roles of the two sites. The roles change from the primary to the standby as well as from standby to primary.
You can use Oracle Full Stack DR service to automate most of these tasks and simplify the required operations. Refer to the Learn About Using OCI Full Stack Disaster Recovery Service with Oracle WebLogic Server Domains for details on this.

To perform a **manual switchover** in an Oracle WLS for OCI DR configuration follow these steps:

a) Pre-Switchover tasks

The pre-switchover steps do not cause downtime:

	PRE-SWITCHOVER STEP	DETAILS
1	Propagate any pending configuration changes	If there are pending changes to replicate, see Configuration Replication for this method to replicate changes to secondary site. After this replication, disable any scheduled replication so it does not run during the switchover.

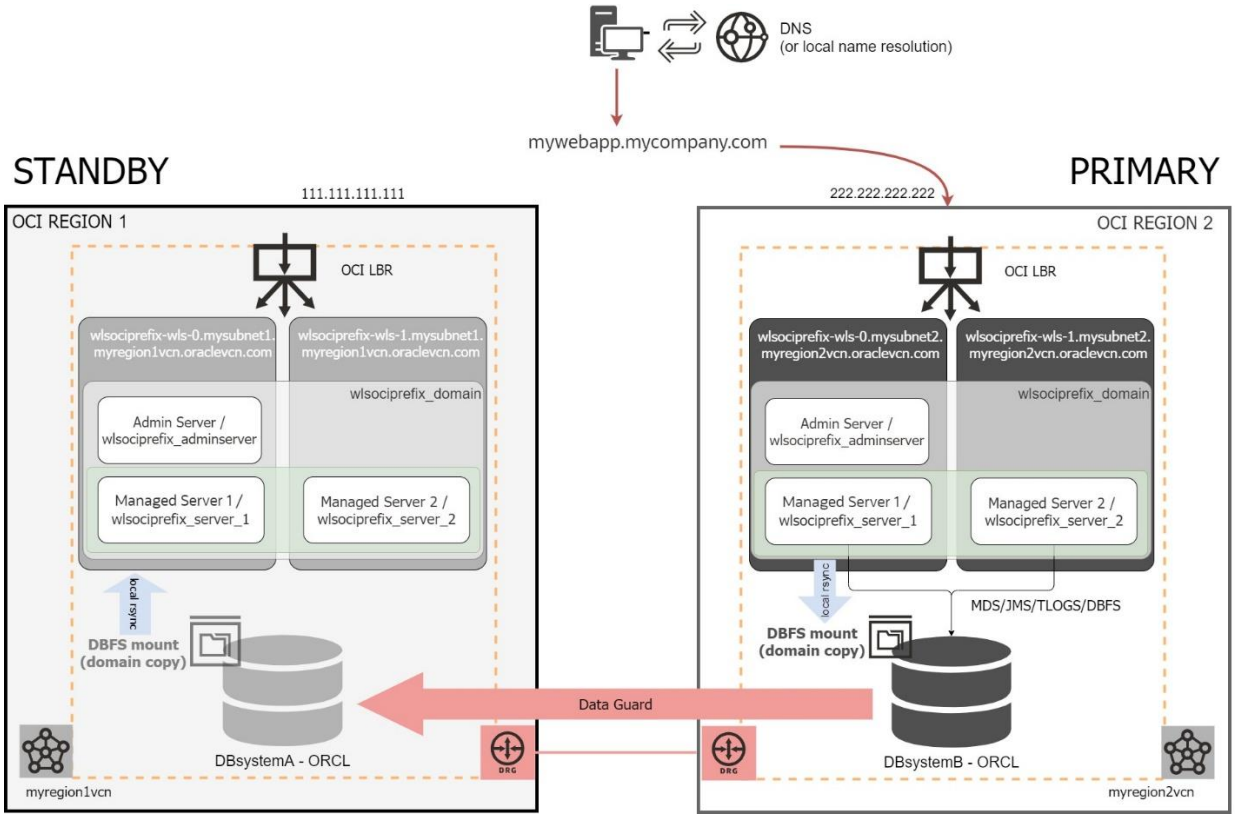
b) Switchover

The actual switchover procedure starts at this point:

	SWITCHOVER STEP	DETAILS
1	Stop servers in primary Site	Use WebLogic Administration Server Console or scripts to stop managed servers in primary Site. The admin server can remain up, although it is recommended to stop it too. NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in

Copyright © 2024, Oracle and/or its affiliates | Public

		https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.
2	Switchover DNS name	Perform the required DNS push in the DNS server that hosts the names used by the system or alter the file host resolution in clients to point the front-end address of the system to the public IP used by LBR in site2. For scenarios where DNS is used for the external front-end resolution (OCI DNS, commercial DNS, etc.), appropriate API can be used to push the change. An example that push this change in an OCI DNS can be found here. Note that the TTL value of the DNS entry will affect to the effective RTO of the switchover: if the TTL is high (example, 20 mins), the DNS change will take that time to be effective in the clients. Use lower TTL values to make this faster. However, low TTLs can cause an overhead because the clients check the DNS more frequently. A good approach is to set the TTL to a low value temporarily (example, 1 min), before the change in the DNS. Then, perform the change, and once the switchover procedure is completed, set the TTL to the normal value again.
3	Switchover Database	Use DG broker in primary db host to perform the switchover. As user oracle: [oracle@drdba ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> switchover to “secondary_db_unqname”
4	Start the servers in secondary site (new primary)	Start the secondary Admin Server (or restart if it was already started, so the configuration changes that were replicated while this was standby take effect.) Start secondary managed servers (use the WebLogic Console or scripts). NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.



Copyright © 2024, Oracle and/or its affiliates | Public

Figure 15 WebLogic for OCI Disaster Recovery system AFTER a switchover (in DBFS based method)

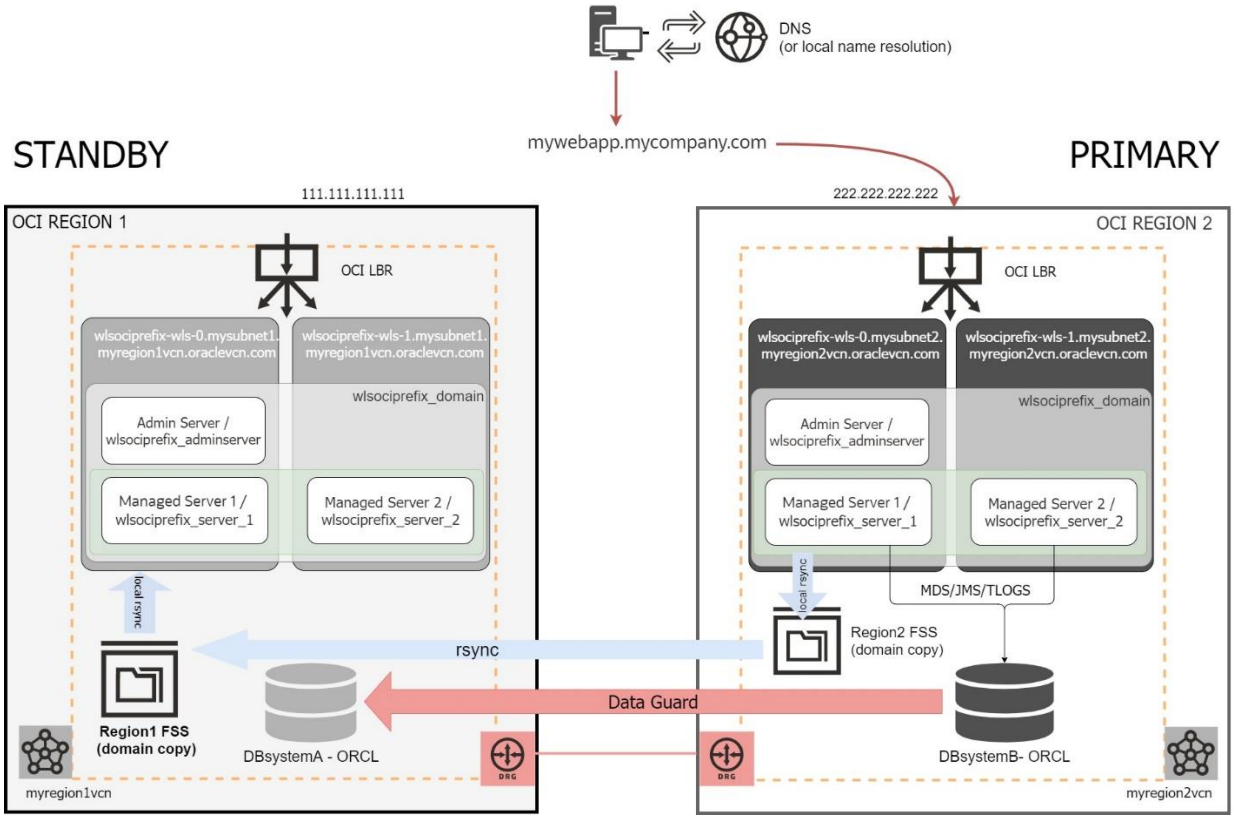


Figure 16 WebLogic for OCI Disaster Recovery system AFTER a switchover (in FSS with rsync method)

Failover for DBFS and FSS with rsync Methods

A failover operation is performed when the primary site becomes unavailable, and it is commonly an unplanned operation. You can role-transition a standby database to a primary database when the original primary database fails and there is no possibility of recovering the primary database in a timely manner. There may or may not be data loss depending upon whether your primary and target standby databases were consistent at the time of the primary database failure.

You can use Oracle Full Stack DR service to automate most of these tasks and simplify the required operations. Refer to the playbook Learn About Using OCI Full Stack Disaster Recovery Service with Oracle WebLogic Server Domains for details on this.

To perform a manual failover in a WLS for OCI DR configuration follow these steps:

	FAILOVER STEP	DETAILS
1	Switchover DNS name	Perform the required DNS push in the DNS server that hosts the names used by the system or alter the file host resolution in clients to point the front-end address of the system to the public IP used by LBR in site2. For scenarios where DNS is used for the external front-end resolution (OCI DNS, commercial DNS, etc.), appropriate API can be used to push the change. An example that push this change in an OCI DNS can be found here.
2	Failover Database	Use DG broker in secondary db host to perform the failover. As user oracle: [oracle@drdbb ~]$ dgmgrl sys/your_sys_password@secondary_db_unqname DGMGRL> failover to “secondary_db_unqname”

Copyright © 2024, Oracle and/or its affiliates | Public

3	Start the servers in secondary site	Start the secondary Admin Server (or restart if it was already started, so the configuration changes that were replicated while this was standby take effect.) Start secondary managed servers (use the WebLogic Console or scripts) NOTE: the WLS for OCI hosts are shipped with scripts to stop and start the proceses out-of-the-box. But you can use the start/stop scripts provided by MAA in https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts These scripts provide more granularity and improved shutdown procedures.

Normally, a failover operation is executed when an outage affects the primary region. Hence, there may be some tasks that you can't perform in primary. For example, you may not be able to stop the WLS processes in primary because the hosts are unreachable. So, once a failover operation is finished, and the previous primary site is reachable again, you must perform some manual tasks to prepare the system for a future switchback. These tasks are:

- **Stop the WebLogic processes** in the failed site. If you didn't stop them during the failover, the processes may be hung. Make sure they are stopped.
- Perform a **Data Guard reinstate operation**. After a failover, the failed primary shows as "Disabled Standby". During the reinstate operation, the database in failed site will be flashbacked and converted as physical standby database. Perform this operation using the OCI Console preferable, so the status is updated in the OCI Console accordingly.
- **Verify** the correct execution of the **configuration replica** (from the new primary to the reinstated secondary).

Scale-out and scale-in for DBFS and FSS with rsync Methods

You can scale-out and scale-in a WLS for OCI system following the steps described in the WLS for OCI documentation Scale a Stack.

When you perform a scale-out or scale-in a WLS for OCI DR environment, there are some characteristics specific to a DR environment that must be considered: there are 2 WLS for OCI systems (primary and secondary) and the domain configuration in secondary is a copy of the primary configuration, so it uses primary hostnames as listen-addresses.

When the listen-address hostnames are added as aliases in the midtier's /etc/hosts, the new nodes provisioned during a scale-out operation do not include these aliases in its /etc/hosts file by default. This can cause the scale-out procedure to fail in the secondary location, because the new nodes cannot connect to WLS administration server. To avoid this problem during scale-out of the WLS for OCI DR environment, required steps are documented in this point.

When you added the primary hostnames entries to a DNS private view in secondary, as described in https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_dr, the scale-out procedures are simplified, because any new node is able to resolve the primary hostnames as soon as it is created.
See the following points for detailed steps.

Scale-out

NOTE: To scale-out environments that are using the TNS alias approach in the datasources, the WLS for OCI instance must have the patch 34988073. The WLS for OCI instances with PSU 12.2.1.4230328 or newer have this patch included.

The recommended procedure to scale-out a WLS for OCI DR environment **in DBFS and FSS with rsyn methods** is as follows:

a) **Scale-out primary** WLS for OCI system:

1. Stop any periodic scheduled execution of config_replica.sh
   IMPORTANT: DO NOT run config_replica.sh replication to secondary until the secondary it is scaled out also. If secondary system configuration has a weblogic server node that is not recognized by secondary servers (the secondary WLS for OCI will not have an equivalent node until it is scaled-out), the startup of some applications may fail in secondary. See About having different number of managed servers in primary and standby.
2. Follow the steps described in Add or Remove WebLogic Server Nodes in the primary stack by increasing the number of the nodes (in this example it is assumed that the number is increased by 1 node).

Copyright © 2024, Oracle and/or its affiliates | Public

3. Once the scale-out has finished correctly, connect with ssh to **the new node** and:
    a. Edit /etc/hosts to add the front-end FQDN with primary front-end LBR IP. Example:

```
# Front-end virtual name for DR, poiting to primary front-end LBR IP
111.111.111.111  mywebapp.mycompany.com
```

    b. (**Not needed if you are using the DNS private view approach for hostname aliases**)
    Edit /etc/hosts in the new node and add the aliases that already exist in the rest of primary nodes, that include secondary names. Example:

```
10.0.0.82  <prim_midtier1_fqdn> <prim_midtier1_shortname> <sec_midtier1_fqdn> <sec_midtier1_shortname>
10.0.0.81  <prim_midtier2_fqdn> <prim_midtier2_shortname>  <sec_midtier2_fqdn> <sec_midtier2_shortname>
```

    c. (**Not needed if you are using the DNS private view approach for hostname aliases**)
    Edit the /etc/oci-hostname.conf and set PRESERVE_HOSTINFO to 3 so these changes are persisted across reboots.
4. Restart the new managed server.

b) **Scale-out secondary** WLS for OCI system:

Scaling-out the secondary requires intervention before the scale-out. Remember that the WebLogic domain configuration in the standby is a copy from primary and it uses the primary hostnames as listen addresses for the servers. When using the /etc/hosts approach for resolving the primary names, the new node that is added to secondary when scaling-out is not aware of them (aliases of the primary names are not included by default in the /etc/hosts file of the new node). To allow the scale-out in the secondary to finish successfully, before proceeding with the scale-out, set the listen-addresses in the secondary domain to the secondary hostnames. This makes that the scale-out procedure run without issues.
If you are using the DNS private view approach for hostnames aliases some of these manual steps can be skipped.

Detailed steps explained here:

1. Convert the standby database into snapshot standby.
2. (**Not needed if you are using the DNS private view approach for hostname aliases**)
Change the listen address of the weblogic servers in the secondary domain to use the secondary instance's fully qualified domain names instead the primary instance fully qualified domain names. This change will be reverted later, it is needed because the new added node will not have in the /etc/hosts the aliases for the primary servers' names that are set in the config.xml. For this:
    - Identify **primary** midtier hosts FQDN names (the existing nodes previous to the scale-out). Example:

```
wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com
```

      Primary midtier1 fqdn is *wlsociprefix-wls-0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com*, and its hostname is *wlsociprefix-wls-0*.
      Primary midtier2 fqdn is *wlsociprefix-wls-1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com* and its hostname is *wlsociprefix-wls-1*.

    - Identify **secondary** midtier hosts FQDN names (the current existing nodes). Example:

```
wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com
```

      NOTE: hostnames are expected to be the same in primary and secondary wls hosts, only the fqdn values will differ.

    - In the secondary site's admin server node, replace primary instance's hosts FQDN values with the secondary hosts FQDN, in the <DOMAIN_HOME>/config/config.xml file:

```
cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_pre_scale-out
sed -i 's/primary_midtier1_fqdn_name/secondary_midtier1_fqdn_name/g' config.xml
sed -i 's/primary_midtier2_fqdn_name/secondary_midtier2_fqdn_name/g' config.xml
```

      Example:

```
sed -i 's/ wlsociprefix-wls-0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-
0.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /g' config.xml
```


Copyright © 2024, Oracle and/or its affiliates | Public

```
sed -i 's/ wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /wlsociprefix-wls-1.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com /g' config.xml
```

3. Start admin and managed servers in the secondary site (start nodemanagers previously if they are not already started).
4. Follow the steps described in Add or Remove WebLogic Server Nodes in secondary stack to add a node.
5. Once the scale-out process finishes, add the required aliases in the new added node:
   - Edit /etc/hosts in the new node and add the virtual front-end FQDN for the secondary front-end LBR IP address, as it is in the rest of the secondary nodes.

```
# Front-end virtual name for DR, pointing to secondary front-end IP
222.222.222.222  mywebapp.mycompany.com
```

   - (**Not needed if you are using the DNS private view approach for hostname aliases**)
     Edit /etc/hosts in the new node and add the existing aliases that secondary midtier nodes already have, where the primary node FQDN are aliases of the secondary local IP addresses. Example:

```
10.2.0.12  <secondary_midtier1_long_name> <secondary_midtier1_shortname>  <primary_midtier1_long_name> <primary_midtier1_shortname>

10.2.0.11  <secondary_midtier2_long_name> <secondary_midtier2_shortname>  <primary_midtier2_long_name> <primary_midtier2_shortname>
```

   - Edit the /etc/oci-hostname.conf and set PRESERVE_HOSTINFO to 3 so these changes are persisted across reboots.
6. (Only if IDCS authentication is used) In the new added node, run the steps explained in 10b (*Configure the primary Enterprise application and App Gateway in the secondary WebLogic domain* of Complete DR configuration when IDCS authentication is used to point its Cloudgate agent to the same App Gateway (and Enterprise Application) than primary, as it was done for the rest of the secondary nodes during the DR setup.
7. Stop servers in secondary site (managed servers and admin).
8. Convert the standby database to **physical standby.**
9. (**Not needed if you are using the DNS private view approach for hostname aliases**). Optionally, you can revert the change done in step 2 and set again the primary FQDN names in the listen addresses. Alternatively, this will be done later when you replicate the conf from primary using config_replica.sh.

c) **Once both primary and standby are scaled out,** complete configuration by adding the aliases for the new node to all the midtier hosts (existing and new nodes):
If you are using the /etc/hosts approach for the hostname aliases:

1. In primary, add it to ALL the existing primary midtier nodes (and also in the new one). Example (this must be in a single line):

```
<primary_newnode_IP>  <primary_newnode_fqdn> <primary_ newnode_hostname>  <secondary_ newnode _fqdn> <secondary_ newnode_hostname>
```

2. In secondary, add it to ALL the existing midtier nodes (and also in the new one). Example (this must be in a single line):

```
<secondary_newnode_IP> <secondary_newnode_fqdn> <secondary_ newnode _hostname>  <primary_ newnode _fqdn> <primary_ newnode_hostname >
```

If you are using the DNS private view approach:

1. Add the hostnames of the new nodes to the appropriate DNS private views instead of adding them to the /etc/hosts. I.e.: add the name of the new secondary node to the primary private view and add the name of the new primary node to the secondary private.

d) At this point, run the config_replica.sh immediately (as usually, first in primary and then in secondary) to **propagate the configuration** from primary to standby.

Scale-in

The recommended procedure to **scale-in** a WLS for OCI DR environment in **DBFS and FSS with rsync methods** is the following:

a) Stop any periodic scheduled execution of the config_replica.sh.
IMPORTANT: DO NOT run config_replica.sh replication to secondary until the secondary it is scaled-in also.
a) **Scale-in primary** WLS for OCI system:

Copyright © 2024, Oracle and/or its affiliates | Public

1. Follow the steps described in Add or Remove WebLogic Server Nodes to reduce the number of servers.

b) **Scale-in secondary** WLs for OCI:

2. Convert the standby database into snapshot standby.
3. Start the admin server only (starting managed servers is not required)
4. Follow the steps described in Add or Remove WebLogic Server Nodes in secondary stack.
5. Once finished, stop processes in secondary and convert secondary database to physical standby.

c) Remove the aliases of the deleted node from the /etc/hosts in primary and secondary midtier hosts, or from the DNS private views if you are using that approach.

d) (optional) Run the config_replica.sh (first in primary and then in secondary) to **propagate the configuration** from primary to standby and verify secondary. You can now enable any periodic scheduled execution of the config_replica.sh.

Auto scaling

When using the auto-scale feature, the scale-out and scale-in operations are automatic operations raised by predefined threshold events. No manual steps can be performed during these operations. This feature can be used in WLS for OCI DR scenarios with the following considerations:

- The auto scale-out/in operations will happen in the system that has the primary role. They should not happen in the site that has the standby role.
- To use the auto-scale feature in the secondary site (when it takes the primary role), it is required that you use the **DNS private view approach** as described in Configure required mid-tier host aliases.
- Auto scale-out/in operations may result in scenarios where the primary site has different number of nodes than secondary. If this is a temporary situation, only during short times, it is not relevant. But if it lasts over time, Oracle recommends to manually scale-out/in secondary site accordingly, to match the same number of nodes than primary.

Recreate the dbfs wallet

NOTE: This applies to DBFS based method only.

During the DR setup, you created a DBFS mount with the script dbfs_dr_setup_root.sh is executed. As explained in Configure the DBFS mount, this creates a user in the PDB (with name dbfsuser) and a wallet is created in DOMAIN_HOME/dbfs/wallet where the username and password are stored. This wallet is used by the dbfs client to connect to the database and mount the dbfs.
If the password of the “dbfsuser” is updated in the database, you must recreate the wallet with the new password. Follow these steps to recreate the wallet:

- Login in the host and switch to “oracle” user. All these commands must be run with oracle user.
- Go to the DOMAIN_HOME/dbfs folder.
- Identify the tns alias that is used to connect to the PDB for the dbfs mount. Normally it is the PDB name. Check the dbfsMount.sh script and look for the dbfs_client line. The tns alias used is shown there with”@”. Example:

```
..
$ORACLE_HOME/bin/dbfs_client -o /u01/data/domains/<domain_name>/dbfs/wallet /@PDB1 -o direct_io -o allow_other
$MOUNT_PATH &>>dbfs.log &
…
```

- Backup the old wallet:

```
mv wallet wallet_bckup
```

- Generate the Oracle Wallet by executing the following command:

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -create
```

  You will be prompted for a password for the wallet.

- Add the new credentials in the wallet by executing the following command. In this example, dbfsuser is the dbfs user name:

Copyright © 2024, Oracle and/or its affiliates | Public

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet  -createCredential <tnsalias_for_pdb> dbfsuser <dbfsuser_password>
```

Example:

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet  -createCredential PDB1 dbfsuser dbfuser_password
```

You will be prompted for the password of the wallet.

- To verify if the wallet is updated with new password, enter the following command:

```
$MIDDLEWARE_HOME/oracle_common/bin/mkstore -wrl /u01/data/domains/<domain_name>/dbfs/wallet -listCredential
```

Then you can remount the dbfs:

- Unmount

```
fusermount -u  /u02/data/dbfs_root
```

- Remount. You run the script:

```
$DOMAIN_HOME/dbfs/dbfsMount.sh
```

Repeat this in all the midtier hosts, primary and standby. Note that the content in the folder $DOMAIN_HOME/dbfs/ is not replicated between primary and standby (and should not be replicated).

Copyright © 2024, Oracle and/or its affiliates | Public

About having compute instances stopped in standby site

The standby database should not be shutdown during normal business operation, because it will not receive updates from primary and it will become out-of-sync. This can result in a data loss in case a switchover needs to be performed. Furthermore, unresolvable gaps in redo between the primary and secondary database may require a full reinstantiation and configuration of the physical standby. It is hence recommended to avoid long periods of disconnection between the primary and standby database. This includes scenarios where the secondary is stopped or problems at the network level that could prevent the communication between the two sites during normal business operations.

The standby midtier compute instances can be stopped without affecting primary, but this has the following implications on disaster recovery:

- Impact in RPO: the domain configuration changes that are replicated from the primary site will not be pushed to the standby domain configuration for DBFS and FSS methods if the standby admin server host is stopped. In case of a failover, the standby domain can be out-of-sync from the primary configuration. This effect is not applicable to the BV approach since the replication happens under the covers without dependencies on the WLS administration node.
- Impact in RTO: the recovery time is increased if the secondary midtier hosts are stopped and need to be started. Recovery time is further increased as the domain configuration synchronization would then need to be executed in the secondary domain to apply any primary site domain configuration changes before a switchover or failover.

To minimize these implications, maintenance operations time and effort when you want to have some secondary midtier compute instances stopped, minimally keep the secondary site's WebLogic Administration compute instance up and shut down only the other WLS managed server compute instances.

NOTE:

Customer billing conditions are out-of-scope of this document. To confirm the impact on your billing of having some servers stopped, contact your Oracle license team in order to get confirmation about your billing conditions.

Note that in all the cases, stopping an instance using the instance's OS does not stop billing for that instance. If you stop an instance this way, be sure to also stop it from the Console or API.

About having different number of managed servers in primary and standby

Oracle strongly recommends having the exact same resources (number of nodes, memory, etc.) in primary and standby WLS for OCI systems, and in case of scaling-out/in primary location, proceed with the same action in secondary as described previously. Having different number of nodes can cause issues at the functional and performance levels. For example, if primary is scaled-out from 2 to 3 nodes, and that configuration is replicated to standby where there are 2 nodes only, some applications may because there is a new node that is unknown for secondary location (not resolvable because it does not exist any equivalent node in secondary site). There can be errors like the following:

> *<May 18, 2020 10:55:48,394 AM GMT> <Error> <Deployer> <BEA-149231> <Unable to set the activation state to true for the application "myCustomApp".*
>
> *weblogic.application.ModuleException: java.net.UnknownHostException: wlsmpdr1-wls-2.sub101714401l0.vncash.oraclevcn.com*

If you face this scenario due to a human error or a recovery situation, as a work-around, you can add a "fake" alias in the secondary wls hosts for the node that exists in primary but not in secondary, so the existing servers can start. The fake alias would point to a non-existing IP address (or the IP address of the secondary db could be used). This would allow the application to start in the existing secondary servers. Although the "new" node in secondary does not exists and won't be contacted, the "unknowknhostexception" error will not happen and the application will be able to start in the existing nodes. Note that you should not try to scale-out this secondary domain to add a new node in this situation, because it is not consistent status (it has the new server in the configuration but there is no real host for it). The correct way to recover from this inconsistent situation would be to switchover back to original primary and scale-in it to make it consistent with the secondary number of nodes again, and then run the config_replica.sh replication to replicate primary config to secondary that will now have the same number of nodes.

Copyright © 2024, Oracle and/or its affiliates | Public

As a summary, **having different number of configured servers in primary and secondary can cause inconsistencies hence it is not recommended.**

Patching the WLS for OCI DR environment

These are the guidelines to apply patches to the Oracle software in a WLS for OCI DR system. Using a Disaster Recovery topology helps (in some cases) to reduce the patching downtime:

- **Database patches**
  WLS for OCI DR topology uses Data Guard. The advantage of having Data Guard instead of only a primary database, is that you can first patch one site and then the other. But not all the database patches allow this approach. The downtime and procedure to patch the database depends on the type of patch. The database patches can be:
    - Data Guard Standby-First. These can be applied first in standby and then in primary. Various options possible. See *"Oracle Patch Assurance - Data Guard Standby-First Patch Apply (Doc ID 1265700.1)"*
    - Non Data Guard Standby-First. These kinds of patches require to be applied on both primary and standby databases at the same time and require shutdown.

  So, if the patch is standby first applicable, the downtime can be minimized or reduced to a switchover.
  If not, it requires shutdown of primary and standby and apply in both.

- **Midtier only patches (that patch only midtier bits)**
  Some FMW patches are marked as FMW_ROLLING_ORACLE_HOME in their readme. This type of patches does not incur in any downtime, regardless of using DR or not.
  However, other patches are not FMW_ROLLING_ORACLE_HOME enabled and require a midtier shutdown. For those cases, a Disaster Recovery topology helps, you can:
  1. Convert secondary database to snapshot standby.
  2. Patch the secondary midtier domain first.
  3. Test the secondary domain with the patch.
  4. Once everything is validated on secondary, convert secondary database back to physical standby.
  5. Switchover to secondary (at this point secondary region becomes your primary and runs the business).
  6. Convert old primary database to snapshot.
  7. Patch old primary midtier and test it.
  8. Convert database back to physical standby.
  9. Then switchback to original site.

  In these cases, the downtime is only the time spent on the switchover operation.
  Without a standby system the downtime would include the patching time, plus the time to stop and start the system.

- **Midtier patches that include db schema changes**
  If the patch is not FMW_ROLLING_ORACLE_HOME enabled the approach in is a bit different to avoid lose db changes (db schema changes require to patch midtier and db at the same time). With a standby system you can:
  1. Convert secondary database to snapshot standby.
  2. Patch the secondary midtier domain first.
  3. Test the secondary domain with the patch.
  4. Once everything is validated on secondary, convert secondary database back to physical standby. At this point, secondary WebLogic domain is misaligned: the midtier has one version but the schemas are in the older version.
  5. Patch primary.

  So, the downtime is the same than without DR, but with the advantage that you can verify the patching procedure and verify the systems behavior in standby first.

Copyright © 2024, Oracle and/or its affiliates | Public

Reassemble the WLS for OCI DR after recreating the standby DB System

There are a few scenarios where the standby DB system may need to be completely recreated. For example, if the primary DB System is restored from a backup, the OCI Console does not yet provide a feature to recreate the standby database from the UI Console. To restore primary database from a backup, it is required to remove the Data Guard association (which is done by terminating the standby DB System) and re-enable it again once the primary database has been restored. This operation will create a new standby DB System.

In WLS for OCI DR environments, when you re-enable DG in primary DB system to re-create the Standby DB System, Oracle recommends **providing the same values for the standby DB System that it had before** (same VCN, same subnet, same hostname prefix). This way, minimal changes are required in the WLS for OCI DR systems in order to use this new DB System as the standby DB.

Follow the steps described below **to reassemble the WLS forOCI DR with a new standby DB system**:

a) Note down the **DB unique name (**$ORACLE_UNQNAME**), private and public IP, VCN, subnet, and hostname prefix** of the **original standby DB System** that is going to be terminated.
b) Once the standby DB System has been terminated, review the /etc/hosts file in the primary DB System host(s). If there is any entry for the terminated standby DB host(s), delete or comment it. A new entry for the standby DB host(s) will be added automatically when it is is created.
c) When you re-enable DG in the primary DB System using OCI Console, make sure you provide the **same VCN, same subnet, same hostname prefix than the previous standby DB System** was using. With this, the only different values in the new standby DB System vs the previous standby DB System will be the DB unique name, the private IP and the public IP.
d) Once the new DB System has been successfully created and the Data Guard configuration is completed in the OCI Console, note down the following values of the new standby DB system: **DB unique name and Private IP**.

For the **Block Volume cross-region replica method**, perform this additional step:

a) Verify that the custom properties in the script that performs the connect string replacements (**replacement_script_BVmodel.sh**) are consistent with the new standby database. If you have reused the same VCN, subnet and hostname prefix values for the new standby, and you connect to the database using the same pdb service name, you shouldn't need to modify anything. If not, you will have to update the custom values in the script, so the replacements are correctly performed in the next switchover.

For the **DBFS and FSS for rsync methods**, perform these additional steps:

a) Update the custom values in the config_replica.sh of the secondary location. Specifically, update the local CDB service name with the value for the recreated standby DB.
b) **(Not needed if you are using the TNS Alias approach)**
This step is needed only in WLS for OCI DR environments configured before TNS alias approach. In the **standby WLS** hosts:
    - Edit the file **/u01/data/domains/local_CDB_jdbcurl.nodelete** and update the standby DB uname with the new standby DB unique name.
c) **(Not needed if you are using the TNS Alias approach)**
This step is needed only in WLS for OCI DR environments configured before TNS alias approach.
If DBFS based method is used, in the **standby WLS** hosts:
    - Edit the file **$DOMAIN_HOME/dbfs/localdb.log**.
    It contains the DB unique name of the original standby System. Replace it with the DB unique name of the new standby DB System.
    - Edit the file **$DOMAIN_HOME/dbfs/tnsnames.ora**. It contains a few aliases. One of the aliases is the original standby DB System unique name. Replace the original standby DB unique name with the new standby DB unique name, in the alias and in the service name of the alias.

    If DBFS based method is used, in the **primary WLS** hosts:
    - Edit the file **$DOMAIN_HOME/dbfs/tnsnames.ora**. It contains a few entries. One of the aliases is the original standby DB System unique name. Replace the original standby DB unique name with the new standby DB unique name (in the alias and service name) and replace the original standby IP with the new standby IP.
    Note that the aliases in tnsnames.ora for the standby CDB may be different in primary and standby WLS hosts. In primary, the standby IP is used to point to secondary CDB, while in standby WLS hosts the

Copyright © 2024, Oracle and/or its affiliates | Public

standby hostname is used. This is expected behavior because it is not expected to have DNS resolution cross-regions.

- No need to update the localdb.log file in primary WLS hosts, as it contains the primary unique name, and this has not changed.

As an example, let's assume the following values:

	Original Standby DB System	New Standby DB System
DB unique name ($ORACLE_UNQNAME)	ORCL_phx1kg	**ORCL_phx1c3**
DB System private IP	10.2.0.2	**10.2.0.5**
DB System hostname	drdb6b.mysubnet.region2vcn.oraclevcn.com	<same value>
DB System scan name	drdb6b-scan.mysubnet.region2vcn.oraclevcn.com	<same value>

Hence, in the **standby WLS hosts**:

File to update	Original content	New content
/u01/data/domains/ local_CDB_jdbcurl.nodelete	drdb6b-scan.mysubnet.region2vcn.oraclevcn.com:1521/ ORCL_phx1kg.mysubnet.region2vcn.oraclevcn.com	drdb6b-scan.mysubnet.region2vcn.oraclevcn.com:1521/ ORCL_phx1c3.mysubnet.region2vcn.oraclevcn.com
(only if DBFS method) **$DOMAIN_HOME/ dbfs/localdb.log**	ORCL6_phx1kg	ORCL6_phx1c3
(only if DBFS method) $DOMAIN_HOME/ dbfs/tnsnames.ora	... ORCL6_phx1kg = (DESCRIPTION = (SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS = (PROTOCOL = TCP)(HOST = drdb6b-scan. mysubnet.region2vcn.oraclevcn.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = ORCL6_phx1kg. mysubnet.region2vcn.oraclevcn.com))) ...	... **ORCL6_phx1c3** = (DESCRIPTION = (SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS = (PROTOCOL = TCP)(HOST = drdb6b-scan.mysubnet.region2vcn.oraclevcn.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = **ORCL6_phx1c3**. mysubnet.region2vcn.oraclevcn.com))) ...

And in the **primary WLS hosts**:

File to update	Original content	New content

Copyright © 2024, Oracle and/or its affiliates | Public

(only if DBFS method) **$DOMAIN_HOME/dbfs/tnsnames.ora**	… ORCL6_phx1kg = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=10.2.0.2)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_ NAME=ORCL6_phx1kg. mysubnet.region2vcn.oraclevcn.com)(UR=A))) ….	…. ORCL6_phx1c3 = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=10.2.0.5)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_ NAME= ORCL6_phx1c3 .mysubnet.region2vcn.oraclevcn.com)(UR=A))) ….

d) Verify that any existing oci security rule created for the original Standby DB System specific IPs is updated to use the new Standby DB System IPs (this is only needed if the rules were specific to the IPs instead of to the CIDRs).

The WLS for OCI DR environment is now ready to use the new Standby DB System!

Copyright © 2024, Oracle and/or its affiliates | Public

RTO AND RPO OVERVIEW

NOTE: The following values are typical values provided for reference purpose only, and they must NOT be taken as contractual values. These times can be different in the customer's system, depending on many factors (the application, the connection pool configuration, the host shapes, the load, the tuning, etc.). Notice that there are formal SLA/SLO values in the Oracle Cloud Pillar documents which are the real contractual obligations in terms of availability by Oracle. You can check those here: https://www.oracle.com/assets/paas-iaas-pub-cld-srvs-pillar-4021422.pdf

Expected RTO

The Recovery Time Objective (RTO) describes the maximum acceptable downtime should an outage occur for a particular system. The downtime caused by a failover depends on multiple "uncontrollable" factors, because it is normally an unplanned event caused by a critical issue that affects the system. But it is possible to measure the required downtime for a planned switchover event.

The following table shows typical times taken by each switchover step in sample WLS for OCI system. This particular system taken as example use VM.Standard2.1 shapes in the WLS hosts, 1G heap memory size for wls servers. They are using a JRF-enabled domain, with out-of-the-box configuration in the connection pools of the WebLogic servers, and the demo application (sample-app) deployed.

	SWITCHOVER STEP	SAMPLE TIMES IN WLS FOR OCI DR
1	Pre-switchover tasks	This does not cause downtime
Downtime starts....		
2	Stop servers in primary Site	
	2.1 Stop managed servers	~ 30 sec (Force) / ~2 min (Graceful)
	2.2 Stop Admin server	~ 8 sec (Force) / ~2 min (Graceful)
3	Switchover DNS name	This is customer specific. For example, if using OCI DNS it can be as low as 30 sec, but it could take hours depending on the DNS provider used. This can be done in parallel with the rest of the steps.
4	Switchover Database	~3 min
5	Start the servers in secondary site	
	5.1 Start Admin	~ 2 min
	5.2 Start managed servers (in parallel)	~ 3 min
... Downtime ends		

Natural delays between steps, or any other additional validation, are not included in the above times, because it depends on how the switchover steps are executed (e.g.: manually, automated with custom scripts, with orchestration custom tools, with Oracle Full Stack DR, etc). So obviously, some additional time must be considered for the total time, not just the arithmetic sum of the times. The time for DNS switchover is also exluded because it is customer specific.

Normally, the **total switchover time is expected** to be **in the 15-30 min range.** Here is a list of tips to minimize the downtime during the switchover operation:

- Perform any switchover related activity that does not require downtime before you stop the primary servers. For example, the WebLogic configuration replication based on config_replica.sh script does not require downtime, you

Copyright © 2024, Oracle and/or its affiliates | Public

can perform it while the primary system is up and running. Another example is to start any shutown host in the standby site.

- If possible, stop the managed servers and admin server in parallel.
- If applications and business allow it, use force shutdown to stop the WebLogic servers.
- The max time taken by the WLS servers to shutdown is limited by the parameters "server lifecycle timeout" (normally set to 30 secs) and "graceful shutdown" (normally set to 120 secs). Make sure that these parameters are configured, to limit the maximum shutdown time.
- The front-end update in DNS is customer dependant. Use a low TTL value in the appropriate DNS entry (at least during the switchover operation) to reduce the time for update. Once the switchover finished, the TTL can be reverted to its original value.
- Using Data Guard Broker commands (dgmgrl) to switchover the database is faster than using the OCI Console. The RTO can be as low as two (2) minutes. However, the roles of each DB System in the OCI Console UI are not refreshed automatically[8]. The database switchover with OCI Console automatically refreshes the roles in the OCI Console, but the DB switchover takes longer when performed with the OCI Console.
- The OCI LBR takes some time also to realize that the servers are up and to start sending reques to them. It is usually some seconds, depending on the frequency of the OCI LBR health checks. Lower the interval used for the checks is, faster it realizes that the servers are up. However, be cautious when you use too low intervals: if the healthcheck is a heavy check, it could overload the backend.

Expected RPO

The Recovery Point Objective (RPO) describes the maximum amount of data loss that can be tolerated. In WLS for OCI, this is related to transaction logs, JMS messages and any other application's information which all resides in the database. Given that the database and the WebLogic configuration are replicated with different mechanisms, we can differentiate between **the RPO for the runtime data** and **the RPO for the WebLogic configuration**.

The actual achievable **RPO for the runtime data** relies upon the RPO of the database, because the runtime data (JMS messages, TLogs, customer data, etc.) are stored in the database. In some cases, there can be runtime artifacts stored in the file systems too (like files consumed by a File Adapter). So, the RPO for the runtime data depends upon the following:

- The available network bandwidth and network reliability between primary and standby. When Dynamic Routing Gateway and Remote VCN peering are used to interconnect primary and standby, the Oracle Cloud Infraestructre backbone network is used. The OCI backbone network provides privately routed inter-region connectivity with consistent performance for bandwidth, latency, and jitter when compared to the public Internet (for more information about the network latency between regions, check Inter-Region Latency Dashboard in the console). Using the OCI backbone When DB systems Data Guard is enabled and the OCI network backbone is used, the RPO is up to five (5) minutes. For an optimum behavior, manual configuration of Fast-Start Failover Observer may be required. Refer to the Oracle DB System documentation to configure Observer.

- The Data Guard protection mode used: either Maximum Availability, Maximum Protection or Maximum Performance (default).
    - **Maximum Availability** mode ensures zero data loss except in the case of certain double faults, such as failure of a primary database after failure of the standby database.
    - **Maximum Performance** mode offers slightly less data protection than maximum availability mode and has minimal impact on primary database performance.
    - **Maximum Protection** mode ensures that no data loss occurs if the primary database fails. To ensure that data loss cannot occur, the primary database shuts down, rather than continue processing transactions, if it cannot write its redo stream to at least one synchronized standby database.
      The best data guard protection mode for a system depends on the business requirements. In some situations, a business cannot afford to lose data regardless of the circumstances. In other situations, the

[8] Open a Service Request in My Oracle Support to get the DB Systems roles updated in the OCI Console in case they are not automatically refreshed after switching over with dgmgrl commands.

Copyright © 2024, Oracle and/or its affiliates | Public

availability of the database may be more important than any potential data loss in the unlikely event of a multiple failure. Finally, some applications always require maximum database performance, and can therefore tolerate a small amount of data loss if any component fails. For more information, see the Oracle Data Guard Protection Modes in the Oracle DataGuard documentation.

- If, additionally, there are runtime artifacts stored in file systems that are not located in the database (e.g., files stored in custom File Storage Services, which are consumed or generated by the customer application), the RPO for this depends on how frequently they are synchronized to the secondary location. What, how and when should this content be synchronized is determined by the business needs. For example: if these runtime files are very volatile (created/consumed fast), syncing it maybe an unnecessary and an overkill. But if the content is more static, and it is required to have it in secondary in case of a DR event, the frequency to copy it should be according to the expected RPO of the system: the RPO will be the amount of data generated between the replications of this content.
  Alternatively, these runtime files can be located in a DBFS file system. In that case, they are replicated to standby via the underlying Data Guard replica, so the RPO is provided by the Data Guard protection mode.

The actual achievable **RPO for the WebLogic configuration** depends upon:

- **How frequently** the WebLogic configuration **is modified**. The WebLogic configuration does not change as dynamically as the runtime data. Despite the initial stages of a system, it is not common to have configuration changes continuously. The more frequently the configuration is modified, the higher amount of config changes could be lost in a disaster event.
- **How frequently** the WebLogic configuration **is synchronized to the standby**.
  In FSS and DBFS replica methods the WebLogic configuration can be replicated manually or automatically with the config_replica.sh script. One approach is to replicate the configuration after every configuration change that is performed in primary. This ensures that secondary WebLogic configuration is always up to date with primary but requires to include the replication process in every change performed to primary. Another approach is to schedule the replication on a regular basis (e.g., every night). In this case, under a DR unplanned event, the configuration changes performed in primary since the last replication will be lost.
  In the Block Volume cross-region replica model, the typical RPO is significantly less than thirty minutes, but it can vary depending on the change rate of data on the source volume.
- The **reliability of the procedure** used for the WebLogic configuration replication. All the replication methods are reliable, but obviously, any failure in the underlying infrastructure (e.g., unavailability of the staging folder, connectivity outages, etc.) can impact on the RPO. Thus, it is recommended to verify the proper functioning of the replication procedure, and to perform regular validations of the secondary site.

Copyright © 2024, Oracle and/or its affiliates | Public

BEST PRACTICES

During the lifecycle of a Disaster Recovery topology, Oracle recommends some best practices:

- Use JDBC persistent stores for your custom resources. Make sure that you create them as JDBC persistent stores. This way, the JMS messages will be stored in database tables, so this information will be replicated to secondary site via Data Guard.
- When you create a new datasource, use a TNS alias in the URL connect string. Make sure the appropriate TNS string exists in the tnsnames.ora both in primary and standby midtier systems. The tnsnames.ora file is particular to each site and is not replicated.
- Maintain the same patch level in primary and standby sites. The software is not replicated automatically to the secondary site in any tier. If you install a patch in primary, you have to install the same patch in the standby location. When you patch the database, check the specific patch's documentation on how to apply the patch in a Data Guard topology.
- Maintain the same configuration in primary and standby sites: any changes applied to the primary system that is not part of the WebLogic Configuration (thus, is not replicated) must be performed in the secondary system too, so both primary and secondary systems have the same configuration. For example: a modification in the primary Load Balancer, any modifications to the operating system, etc.
- Perform regular switchovers to verify the health of the secondary site. You can alternatively open the secondary site for validation without performing a complete switchover.
- For application deployment operations, Oracle recommends using the WebLogic deployment "Upload your files" option in the WebLogic Administration Console so that the deployed files are placed under the upload directory of the Administration Server (under domain directory/servers/admin_server_name/upload). That way these files will be synced to standby by the configuration replication script or by the BV copy (depending on whether DBFS, FSS or BV is used).
- Perform regular block volume backups, or configure automatic block volume backup, in the block volumes used by the WLS hosts in primary and standby. See About Volume Backups in the WLS for OCI documentation for more information.

Copyright © 2024, Oracle and/or its affiliates | Public

CONCLUSION

Disaster recovery in an Oracle WLS for OCI configuration consists of a production database and a standby database synchronized by Oracle Data Guard, two middle tier configurations pointing to their local database, and a solution to manage the minimally necessary file replication. With this Disaster Recovery solution, Oracle Cloud eliminates the costs and complexity of owning and managing a standby hardware, third party replication software, and remote data center, while achieving industry-leading Recovery Time Objective and Recovery Point Objective.

The use of Oracle Data Guard for disaster recovery provides better RTO and RPO than restoring a remote backup; production is quickly failed over to an already running and synchronized copy of your production database on the Oracle Cloud. The standby database in the cloud not only provides disaster recovery, but it can also be used to seed clone databases for development and test.

The use of middle tiers with a streamlined configuration replication facilitates maintenance and reduces the overhead caused by continuous configuration approaches. However, an appropriate methodology and regular standby verifications are needed to guarantee a consistent recovery. Depending on each system's lifecycle, different configuration synchronization approaches may be used for optimum behavior.

Copyright © 2024, Oracle and/or its affiliates | Public

APPENDIX A – CONSIDERATIONS FOR RAC DATABASES

When the database used by the WebLogic Server is an Oracle Real Application Cluster (RAC) database, you can configure Disaster Recovery as described in this document, with the following considerations.

About the WebLogic on OCI Edition when using RAC

This document applies to Oracle WebLogic Suite Edition and to WebLogic Enterprise Edition except when Oracle RAC database is used. When **RAC** is used, this document **supports only Oracle WebLogic Suite Edition** because it is the only edition that uses GridLink datasources. It is an MAA best practice to use GridLink datasources when RAC is used.

Network considerations when using RAC

Single Client Scan Address (SCAN) is a feature used in Oracle Real Application Clusters environments that provides a single name for clients to access any Oracle Database running in a cluster. The SCAN address is a fully qualified domain name configured to resolve to all the addresses allocated for the SCAN (usually 3 IP addresses). SCAN listeners can run on any node in the cluster. SCANs provide location independence for the databases, so that client configuration does not have to depend on which nodes run a particular database. When a connection is established against the SCAN listener, it is then redirected to one of database listener that run in each database nodes and listen in a **virtual IP (VIP)**. A typical connect attempt from a database client to an Oracle RAC database instance can be summarized, as follows:

- The database client connects to SCAN, providing a valid service name.
- The SCAN listener then determines which database instance hosts this service and routes the client to the local or node listener on the respective node.
- The node listener, listening on a node VIP and a given port, retrieves the connection request and connects the client to the instance on the local node.

Oracle WebLogic Server Suite on OCI datasources are automatically configured pointing to the scan listener when it is provisioned using a RAC, which is the best practice.

When using the DBFS method, each midtier **must be able connect to the remote RAC**. This communication is used for config replication based on the config_replica.sh script (in DBFS method only), but never will be used for runtime. **Ensure that the required network rules are defined to allow communication from each midtier hosts to the remote RAC scan IPs, VIPs and hosts IPs on port 1521.**

NOTE: Dynamic Routing Gateway do not provide DNS resolution for the names that are in the remote network. The tnsnames.ora aliases to point to the remote RAC database use the IP address rathen than the scan address name.

Create and use a CRS database service

The GridLink datasources provide dynamic load balancing and failover across the nodes in an Oracle Database cluster, and also receive notifications from the database cluster when nodes are added or removed. To take advantage of these capabilities, Oracle recommends using Oracle Database service that supports Cluster Ready Services (CRS) and the Oracle Notification Service (ONS). These services monitor the status of resources in the database cluster and generate notifications when a status changes.

Instructions to create CRS managed services in a RAC dataguard are provided in this document, in the step to setup the database: Configuring a custom PDB service in the DB Systems (for Oracle Base Database Service DB Systems) and Considerations for EXACS (for Oracle Exadata Database Service).

Apply other best practices

Regardless the Disaster Recovery setup, there are other best practices for the WebLogic Server data sources configuration that are recommended when connecting to a RAC database. These can be applied at any point of time:

- Use **auto-ons**
  If you are using an Oracle 12c database **or later**, the ONS list is automatically provided from the database to the driver. You can leave empty the ONS Nodes list in the datasources configuration.
- "**Test Connections On Reserve**"

Copyright © 2024, Oracle and/or its affiliates | Public

Verify that the "**Test Connections On Reserve**" is checked in the datasources.
Despite the GridLink datasources received FAN events when a RAC instance becomes unavailable, it is a best practice to enable the test connection on reserve in the datasource so you are sure that the connection returned to the application is good.

- "**Seconds to Trust an Idle Pool Connection"**
  For a maximum efficiency of the test, you can also set "**Seconds to Trust an Idle Pool Connection"** to 0, so the connections are always verified. Setting it to zero means that all the connections returned to the application will be tested. If this parameter is set to 10, the result of the previous test is valid for 10 seconds and if a connection is reused before that time has passed, the result is considered still valid.
- "**Test Frequency**"
  Verify that the "**Test Frequency**" parameter in the datasources is not 0.
  This is the number of seconds a WebLogic Server instance waits between attempts when testing unused connections. The default value of 120 is normally enough.

Copyright © 2024, Oracle and/or its affiliates | Public

APPENDIX B – DB SYSTEM BACKUPS ON MANUALLY CONFIGURED DATA GUARD

The back up of the DB System is a key aspect of any Oracle database environment. Oracle Cloud offer various approaches. You can: store backups in local or cloud storage; the backup can be automatic, custom rman, or dbcli. In a DR scenario, there are some special considerations because the databases are configured with Data Guard.

When the Data Guard was configured manually (Option 2) Configuring Data Guard manually) the backup needs to be configured manually to get the optimal configuration in a Data Guard environment. You need to perform the backups in one of the databases (primary or standby) and control the archivelog growth in the other one.

To configure manual backups in the primary DB System:

- If the automatic backup was enabled in OCI Console for this system, the backup module should be already configured by the automatic backups. In that case, disable automatic backup so you can customize it. If automatic backup have never been enabled before, you can follow the steps described in Backing Up a Database to Object Storage Using RMAN to install and configure the backup module in the Primary DB.
- Configure rman settings as recommended in the link. In addition to that, ensure that you also include the archivelog deletion policy recommended for Data Guards:

```
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 1 TIMES TO 'SBT_TAPE' APPLIED ON ALL STANDBY;
```

- Create your rman backup scripts as per your backup requirements and include it in the crontab. This is an example to run a full backup:

```
# Run RMAN
export ORACLE_HOME=/u01/app/oracle/product/18.0.0.0/dbhome_1
export ORACLE_SID=ORCL
$ORACLE_HOME/bin/rman <<RMAN
   connect target /
   SET ENCRYPTION ON;
   BACKUP DATABASE PLUS ARCHIVELOG TAG "FULL_BACKUP";
   exit;
RMAN
 echo "Completed full backup for" $ORACLE_SID
```

To control the archivelog growth in the standby:

- Disable automatic backup if it was enabled for this system, and then configure the proper archivelog deletion policy so archivelog are not deleted if they are not yet applied to standby with the following command.

```
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

- Although setting the correct archivelog deletion policy should be enough to control the archivelog growth in the FRA, you can also create a cleanup script to delete old archive logs. This is an example to clean old archive logs that uses an archivelog deletion policy to prevent undesired archivelog deletion:

```
#######################################################
# Use this script to clean old archive logs from disk
# when the database is in STANDBY role and no backups are performed
# Run RMAN
export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
export ORACLE_SID=ORCL
$ORACLE_HOME/bin/rman <<RMAN
   connect target /
 # To prevent undesired archivelog deletion if this DB takes primary role
   CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
   # Delete archivelog older than 20 days
   delete noprompt archivelog all completed before 'SYSDATE-20';
   exit;
RMAN
 echo "deleted applied old archivelogs on $ORACLE_SID"
#######################################################
```


Copyright © 2024, Oracle and/or its affiliates | Public

If Data Guard was configured with the Cloud Console UI, you can enable automatic backups in the primary or standby database with the Cloud UI Console. The default rman configuration in those cases should use the recommended archivelog deletion policy for the Data Guard scenario. However, you can control the archivelog growth in the secondary database as well as explained before.

NOTE: The Oracle Data Guard configuration in the topology should provide protection for most database failure scenarios. I.e. in most cases, should a failure occur in the primary database, switching over to standby will allow to resume operations. In the extreme case where the primary is failing and a switchover to standby is impossible, the primary may need to be restored from a backup. In such a infrequent scenario, the standby database will have to be recreated as well. To recreate the standby database:
*In a **manual Data Guard**, you can re-run the scripts that are provided in the step Option 2) Configuring Data Guard manually in order to recreate the standby database and reconfigure the Data Guard again after a restore in the primary database.*
*In an **automated Data Guard**, however, the OCI Console does not yet provide a feature to recreate the standby database from the UI Console. To restore primary database from a backup, it is required to remove the Data Guard association (which is done by terminating the standby DB System) and re-enable it again once the primary database has been restored. This will create a new standby DB System. Some properties need to be updated in the WLS for OCI midtiers to reassemble them with this new standby system. See the point Reassemble the WLS for OCI DR after recreating the standby DB System.*

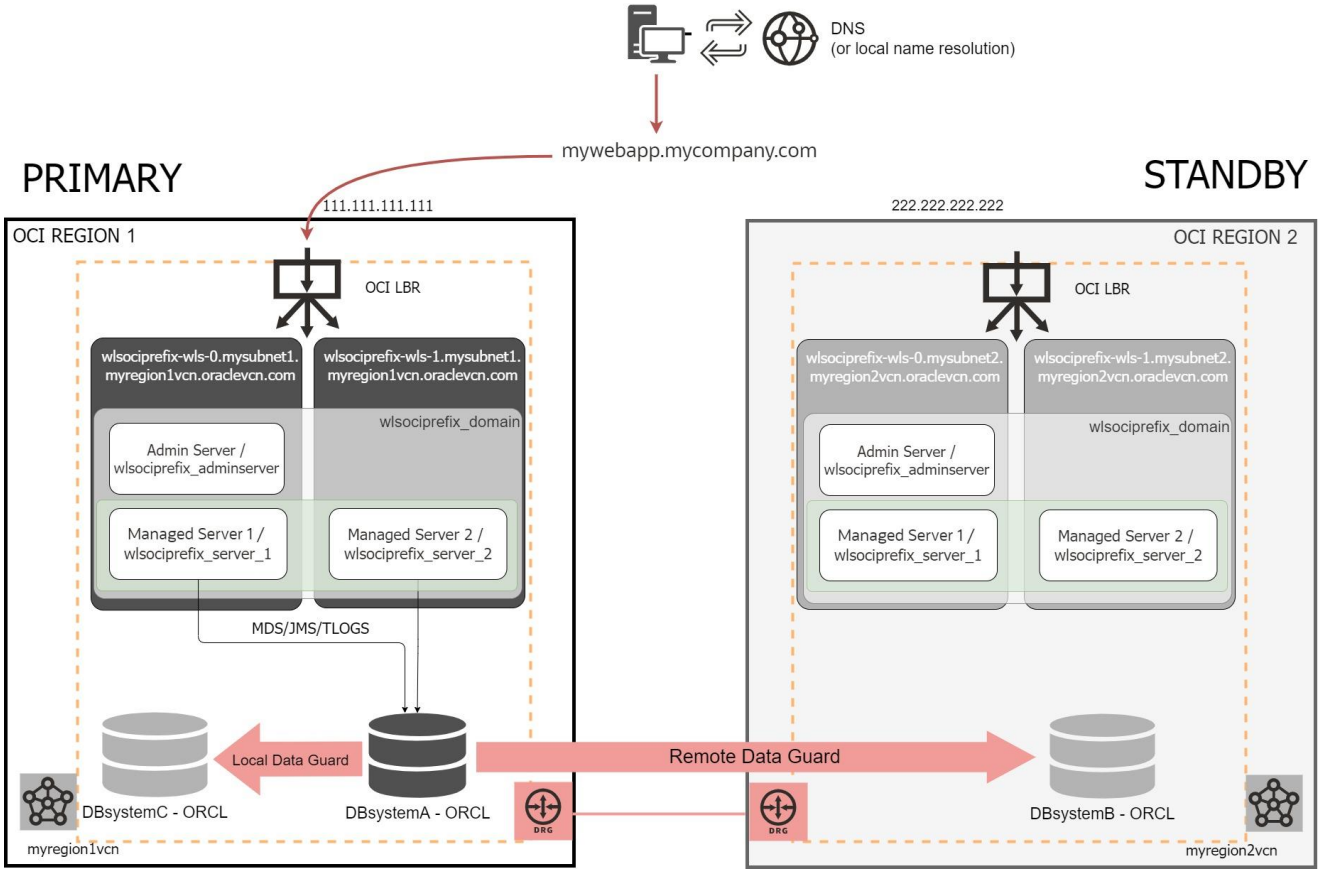
Copyright © 2024, Oracle and/or its affiliates | Public

APPENDIX C – USING ADDITIONAL STANDBY DATABASE IN PRIMARY

The OCI Console supports **only one standby database per primary database**. Consistently, this document assumes that there is only ona standby in the secondary region.

In scenarios where you have manually added an additional standby database in the primary site, you can perform the cross-region DR setup with the following considerations:

1. If you use the **Block Volume cross-region replica method**, you can perform the DR setup as usual. You only must make sure that you update the appropriate connect string in each site.
2. **If you used DBFS or FSS with rsync, you need to remove the local standby from the Data Guard broker configuration before running any DR setup scripts.** You can add it again later once the DRS has been executed. See more details in following steps.



Additional local standby Pre-Configuration Steps (DBFS and FSS with rsync Methods)

Before running DR scripts to setup the cross-region DR:

1. Disable and remove the local standby from the Data Guard configuration. Only the cross-region standby must exist in the Data Guard configuration before running DR setup scripts. Example:

Copyright © 2024, Oracle and/or its affiliates | Public

```
DGMGRL> show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
    ORCL_frankfurt - Physical standby database
    ORCL_london2 - Physical standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 130 seconds ago)

DGMGRL> disable database ORCL_london2
Disabled.

DGMGRL>  show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
    ORCL_frankfurt - Physical standby database
    ORCL_london2 - Physical standby database (disabled)
 ORA-16749: The member was disabled manually.
Fast-Start Failover:  Disabled
Configuration Status:

SUCCESS   (status updated 1 seconds ago)

DGMGRL>  remove database ORCL_london2
Removed database "orcl_london2" from the configuration

DGMGRL>  show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 1 seconds ago)
```

2. **(Not needed if you are using the TNS Alias approach)**. This step is needed only in WLS for OCI DR environments configured before TNS alias approach. Make sure that the dual string connection is NOT used in primary system. The dual string connection can be set once the DR setup has been completed. Before setting up DR, the syntax in the **datasources and jps-config.xml** file must be compliant with the recommended formats:
    - If the database is a single instance, the recommended db connect string is:
      jdbc:oracle:thin:@//<db-scan-address>:<port>/<pdb_service_name>
    - If the database is a RAC, the datasources must be GridLink datasources and the recommended db connect string is:
      jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=<db-scan-address>)(PORT=<port>)))(CONNECT_DATA=(SERVICE_NAME=<pdb_service_name>)))

Additional local standby Cross-Region DR configuration (DBFS and FSS with rsync Methods)

Once the pre configuration steps have been performed, you can do the DR setup as described in this document. During the DR setup, the local standby is out of the picture.

Copyright © 2024, Oracle and/or its affiliates | Public

Additional local standby Post-Configuration Steps (DBFS and FSS with rsync Methods

After the DR setup has been completed, you can add again the local standby to the topology and perform the needed adjustments to the system. Execute the following steps:

1. Add and enable the local standby database to the Data Guard configuration again. Example:

```
DGMGRL> add database 'ORCL_london2' as connect identifier is 'ORCL_london2' maintained as physical;
Database "ORCL_london2" added

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby database
  ORCL_london2 - Physical standby database (disabled)
    ORA-16905: The member was not enabled yet.
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 29 seconds ago)

DGMGRL> enable database ORCL_london2;
Enabled.

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
    ORCL_frankfurt - Physical standby database
    ORCL_london2 - Physical standby database
     Warning: ORA-16857: member disconnected from redo source for longer than specified threshold
Fast-Start Failover:  DisabledConfiguration Status:

WARNING   (status updated 57 seconds ago)
```

The new added member will take some time until it receives and apples the pending redo. Example:

```
DGMGRL> validate database ORCL_london2
  Database Role:    Physical standby database
  Primary Database:  ORCL_london1
  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
  Managed by Clusterware:
    ORCL_london1:  YES
    ORCL_london2:  YES

  Standby Apply-Related Information:
    Apply State:      Running
    Apply Lag:        19 hours 45 minutes 12 seconds (computed 1 second ago)
    Apply Delay:      0 minutes

DGMGRL> /
  Database Role:    Physical standby database
  Primary Database:  ORCL_london1
  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
  Managed by Clusterware:
    ORCL_lhr3xb:  YES
    ORCL_lhr132:  YES
```


Copyright © 2024, Oracle and/or its affiliates | Public

```
DGMGRL> show configuration
Configuration - ORCL_DG_CONF
  Protection Mode: MaxPerformance
  Members:
  ORCL_london1 - Primary database
    ORCL_frankfurt - Physical standby database
    ORCL_london2 - Physical standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS   (status updated 55 seconds ago)
```

In case that the required archives are not available anymore, it requires manual intervention. You will have to manually locate pending archives or refresh the local standby form primary with “restore from service”.

2. The pdb service name must be the same in the primary and local standby, this way you can use the dual string in the primary WebLogic system. If not already done, create a new CRS service in primary and local standby systems as follows:
   In the primary database system, create a service for the PDB with the primary and snapshot standby roles. Example:

```
srvctl add service -db $ORACLE_UNQNAME -service mypdbservice.example.com -preferred ORCL1,ORCL2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"

srvctl modify service -db $ORACLE_UNQNAME -service mypdbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT

srvctl config service -db $ORACLE_UNQNAME -service mypdbservice.example.com
```

   In the local standby database system, do the same. Example:

```
srvctl add service -db $ORACLE_UNQNAME -service mypdbservice.example.com -preferred ORCL1,ORCL2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY "

srvctl modify service -db $ORACLE_UNQNAME -service mypdbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT

srvctl config service -db $ORACLE_UNQNAME -service mypdbservice.example.com
```

   Start the new service in the one that is the primary role in that moment (next times, this service will be automatically stopped/started by DG broker). Example:

```
srvctl start service -db $ORACLE_UNQNAME -service mypdbservice.example.com
```

3. In the midtier hosts of the site that uses the local standby database, the tns alias in the tnsnames.ora of the tns admin folder should be dual. It should include the addresses of primary and local standby. Example of the tns alias entry in the tnsnames.ora:

```
MYALIAS =
(DESCRIPTION=
(CONNECT_TIMEOUT=15)(RETRY_COUNT=5)(RETRY_DELAY=5)
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdba-scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdbc-scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=mypdbservice.example.com)))
```

4. **(Not needed if you are using the TNS Alias approach)**.
   Configure the **dual string connection in the primary site** WebLogic configuration. Configure it both in the datasources and in the jps-config.xml file. The string must be the same in the datasources and in the jps-config.xml file. Example of dual datasource connection string that includes the local standby:

Copyright © 2024, Oracle and/or its affiliates | Public

```
jdbc:oracle:thin:@(DESCRIPTION=
(CONNECT_TIMEOUT=15)(RETRY_COUNT=5)(RETRY_DELAY=5)
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdba-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdbc-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=mypdbservice.example.com)))
```

***NOTE**: In this example, the line breaks are used for clarity. **Do not add line breaks nor blank spaces** in the connect string. Add this as a SINGLE line in the datasources and jps-config.xml files.*

The secondary site will not use the dual connect string. Do not modify it. The secondary midtier connects to its local database only. The script config_replica.sh automatically performs the required connect string replacements in the copied configuration when it runs in the standby site.

5. Make sure you are using the latest version of the config_replica.sh script. Update it accordingly: you need to provide the CDB connection string for the local standby database.
   WARNING: latest version of the config_replica.sh script is valid only for TNS alias approach.

6. **(Not needed if you are using the TNS Alias approach)**.
   This step is needed only in WLS for OCI DR environments configured before TNS alias approach. In the primary site midtier, update the file **/u01/data/domains/local_CDB_jdbcurl.nodelete** in the mid-tier hosts. You must add an additional line with the jdbc url connection for the local standby CDB service.
   Before the modification, it contains only 1 row that points to the local primary database. Example:

```
drdba-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london1.mysubnet.region1vcn.oraclevcn.com
```

   After adding the local standby, it must contain 2 rows: the local primary database and the local standby. Example:

```
drdba-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london1.mysubnet.region1vcn.oraclevcn.com
drdbc-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london2.mysubnet.region1vcn.oraclevcn.com
```

   IMPORTANT: in this file, you must provide the CDB service names, not the PDB service name.

   The DR setup scripts created this file during the DR configuration. During the lifecycle, the config_replica.sh script uses this info to connect to the local CDB, and to retrieve the current role of the site. When you use a local standby database, the file must contain the standby local CDB connect string too. This way, the config_replica.sh can retrieve the site role correctly if a local switchover has occurred.

7. **(Not needed if you are using the TNS Alias approach)**.
   This step is needed only in WLS for OCI DR environments configured before TNS alias approach
   If DBFS method is used for the replication, add an alias to the additional standby in the tnsnames.ora of the mid-tier hosts. The alias must use **the unique name of the local standby database**, and it must point to the CDB default service name. Add it both to primary and standby mid-tier hosts. It is used by the config_replica.sh script when the site is in standby role.
   - Edit the $DOMAIN_HOME/dbfs/tnsnames.ora file of each midtier
   - In the midtier hosts of the primary site (where the additional local standy is), add the alias to the local standby CDB. Example:

```
..

ORCL_london2 =
(DESCRIPTION =
(SDU=65536)(RECV_BUF_SIZE=10485760)(SEND_BUF_SIZE=10485760)
(ADDRESS = (PROTOCOL = TCP)(HOST = drdbc-scan.mysubnet.region1vcn.oraclevcn.com)(PORT =1521))
(CONNECT_DATA= (SERVER = DEDICATED) (SERVICE_NAME = ORCL_london2. mysubnet.region1vcn.oraclevcn.com)))

…
```

   In this example:
   “ORCL_london2” is the DB unique name of the local standby.
   “drdbc-scan.mysubnet.region1vcn.oraclevcn.com” is the scan address of the local standby.

Copyright © 2024, Oracle and/or its affiliates | Public

"ORCL_london2. mysubnet.region1vcn.oraclevcn.com" is the default service name of the local standby CDB.

- In the secondary site, the scan name is probably not resolved, so use IPs instead of the scan name. Example:

```
…
ORCL_london2=
(DESCRIPTION=
(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)
(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.42)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.43)(PORT=1521)))
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.44)(PORT=1521)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME= ORCL_london2. mysubnet.region1vcn.oraclevcn.com)))
..
```

In this example:
"ORCL_london2" is the DB unique name of the local standby.
"10.0.2.42", "10.0.2.43","10.0.2.44" are the scan IPs of the local standby.
"ORCL_london2. mysubnet.region1vcn.oraclevcn.com" is the default service name of the local standby CDB.

8. If the DBFS mount is used (either for the config replication or for other purposes), the entry that points to the PDB in the $DOMAIN_HOME/dbfs/tnsnames.ora should use dual connect string. In case of a local switchover, the DBFS must be able to connect to the local standby.
   - Identify which is the tns alias used by the dbfs mounts. Maybe "ORCL" or the PDB name.
   - Locate the alias in the $DOMAIN_HOME/dbfs/tnsnames.ora file
   - Modify the tns entry to a dual string format that includes the local standby PDB. Like this example:

```
PDB1 =
(DESCRIPTION =
 (CONNECT_TIMEOUT= 10)(RETRY_COUNT=10) (RETRY_DELAY=10)
 (ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=drdba-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
 (ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=drdbrc-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
 (CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = mypdbservice.example.com)))
```

Additional local standby local database switchover (DBFS and FSS with rsync Methods)

An additional lifecycle operation is possible in this scenario. You can switchover the primary database to the local standby database. In this operation, the midtier does not switch over. The WebLogic servers reconnect automatically to the new local primary database once the local database switchover is completed. This is achieved by using the dual database connection string configured in the datasources and jps configuration files.

Oracle recommends validating the configuration replica procedure under this scenario. Use these steps to validate:

1. Switchover the primary database to the local standby database. While the switchover is taking place, the database is not available, so this operation causes an outage in the system. Perform this step in a maintenance window to minimize the impact in the application.
2. Perform a configuration change in the primary WebLogic configuration. For example, change a connection pool size in one of the datasources.
3. Replicate the WebLogic configuration to the secondary site. As usual: first run the config_replica.sh in the primary WebLogic Administration host, and then run the config_replica.sh in the secondary WebLogic Administration host.
4. Verify that the WebLogic configuration has been properly replicated to the secondary site. Check whether the change is present in the secondary domain files.
5. For a more complete verification, you can convert the standby database (the remote standby, the one that is in the secondary region) to snapshot standby and start the secondary WebLogic Administration server. Verify whether

Copyright © 2024, Oracle and/or its affiliates | Public

the change is present in the WebLogic Configuration console. Then, stop the secondary WebLogic Administration server and convert the standby database from snapshot standby to physical standby again.

6. Switchover the primary database to the original primary to revert the system to the original status.

Copyright © 2024, Oracle and/or its affiliates | Public

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Copyright © 2024, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Oracle WebLogic Server for Oracle Cloud Infrastructure Disaster Recovery
February, 2024