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. This paper is oriented to a technical audience having knowledge of Oracle Cloud, Oracle WebLogic, Oracle Database, Data Guard and Oracle Database backup and recovery.

DISCLAIMER
This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

REVISION HISTORY
The following revisions have been made to this white paper:

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>August, 2020</td>
<td>1</td>
<td>Initial publication</td>
</tr>
<tr>
<td>December, 2020</td>
<td>2</td>
<td>Added Best practices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added scale-out procedure for DR environment in Appendix D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added support for IDCS Authentication in DR scenarios</td>
</tr>
<tr>
<td>March, 2021</td>
<td>3</td>
<td>Added additional setup option (based in FSS with rsync method instead on DBFS). Several sections updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added instructions for “Recreating the standby DB System” in the Appendix D</td>
</tr>
<tr>
<td>April, 2021</td>
<td>4</td>
<td>Added section “About having compute instances stopped in standby site”</td>
</tr>
<tr>
<td>May, 2021</td>
<td>5</td>
<td>Enhancements in wordings</td>
</tr>
<tr>
<td>July, 2021</td>
<td>6</td>
<td>Added additional setup option (based in Block Volume Cross-Region replication) in the Appendix F</td>
</tr>
<tr>
<td>September, 2021</td>
<td>7</td>
<td>Added footnote in page 6</td>
</tr>
</tbody>
</table>
# Table of Contents

**Purpose Statement** 1  
**Disclaimer** 1  
**Revision History** 1  
**Introduction** 4  
**Disaster Recovery Solution for Oracle WLS for OCI** 6  
  - Topology description 6  
  - Assumptions 9  
    - WebLogic Editions 9  
    - Authentication 10  
    - Load Balancer 10  
    - Database 11  
  - Requirements 12  
    - Front-end address 12  
    - WebLogic Resource Name Prefix 12  
    - Network communication between sites 12  
    - Staging filesystems for the WebLogic config replication 12  
    - Custom files 13  
**Disaster Recovery setup** 14  
  1. Choose a virtual front-end name 15  
  2. Prepare Primary mid-tier for the virtual front-end 15  
  3. Setup the Database in Secondary Site 16  
    3.1 Option 1) Configuring the Data Guard using OCI Console 17  
    3.2 Option 2) Configuring the Data Guard manually 17  
  4. Provision WLS for OCI in Secondary Site 20  
  5. Prepare Secondary mid-tier for the virtual front-end 23  
  6. Configure required host aliases 23  
  7. Configure the staging mounts for WebLogic config replication 24  
    7.1 Option 1) Configure the DBFS mounts for DBFS based method 24  
    7.2 Option 2) Configure the FSS mounts for FSS with rsync method 26  
  8. Run DR setup script in Primary WebLogic Domain 28  
  9. Run DR setup script in Secondary WebLogic Domain 29  
  10. Complete DR configuration when IDCS authentication is used 31  
  11. Verify the DR Setup 36  
**Lifecycle procedures** 37  
  - Replicating configuration changes 37  
  - Switchover 41  
  - Failover 43  
**Best practices** 44  
**Conclusion** 45  
**Appendix A – Considerations for RAC DB Systems** 46  
  - About the WebLogic on OCI Edition when using RAC 46  
  - Network considerations when using RAC DB System 46  
  - Data Guard Configuration for RAC DB System 46  
  - Create and use a CRS database service 46  
  - Running the dbfs_dr_setup_root.sh in RAC scenarios 48  
  - Running the DR setup scripts in RAC scenarios 48  
  - Apply other best practices 49
Appendix B – Using Enterprise Manager Site Guard to manage Disaster Recovery

Appendix C – DB System Backups on manual Data Guard

Appendix D – Additional lifecycle operations
  - Open Secondary Site for validation
  - How to recreate the dbfs wallet
  - About having compute instances stopped in standby site
  - Scale-out and Scale-in procedures in a DR environment
    - Scale-out
    - Scale-in
  - About having different number of managed servers in primary and standby
  - Reassemble the WLS for OCI DR after recreating the standby DB System

Appendix E – Additional Best Practices
  - Use JDBC stores for TLOGs and JMS stores

Appendix F – Disaster Recovery Based On Block Volume Cross-Region Replication
  - Topology Description
  - Advantages and Disadvantages of BV Replication DR model
  - Assumptions for BV Replication DR model
    - WebLogic Editions
    - Authentication
    - Load Balancer
    - Database
    - Block Volumes Replicated
  - Requirements for BV Replication DR model
    - Front-end address
    - WebLogic Resource Name Prefix
    - Network communication between sites
    - Regions with Block Volume Cross-Region Replication
  - Setup Process for BV Replication DR model
    1. Choose a virtual front-end name
    2. Prepare Primary mid-tier for the virtual front-end
    3. Setup the Database in Secondary Site
    4. Provision WLS for OCI in Secondary Site
    5. Prepare Secondary mid-tier for the virtual front-end
    6. Create the mid-tier hosts aliases
    7. Convert the standby DB into physical standby
    8. Configure the Block Volume Cross-Region replication
    9. Prepare the script for the environment specific replacements
    10. Complete DR configuration when IDCS authentication is used
    11. Verify the DR Setup
  - Lifecycle Operations for BV Replication DR model
    - Switchover
    - Failover
    - Open Secondary Site for Validation
    - Scale-out and Scale-in
  - Using OCI CLI commands for automating Block Volume Replication tasks
INTRODUCTION

Oracle’s Maximum Availability Architecture (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’s Maximum Availability Architecture best practices is one of the key requirements for any Oracle deployment. Oracle WebLogic and Oracle Databases include an extensive set of high availability features, such as process death detection and restart, clustering, server migration, clusterware integration, GridLink, load balancing, failover, backup and recovery, rolling upgrades, and rolling configuration changes, which protects application deployments from unplanned downtime and minimize planned downtime.

Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WLS for OCI) is a Platform as a Service (PaaS) computing platform solution provisioned via Oracle Marketplace that allows you to create your Java Enterprise Edition (Java EE) application environment in Oracle Cloud Infrastructure very quickly. In a typical Oracle WLS for OCI deployment, the application data (such as application-specific schemas, jms stores etc.) is stored in a database. Oracle WLS for OCI can be configured as a JFR-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.

Any critical system needs protection from unforeseen disasters and natural calamities. This protection is also required for systems deployed in the Cloud, like Oracle WLS for OCI. It needs to address front-end access (like load balancer based front-end host names), the middle tier (Oracle WebLogic) and the data tier (Database Cloud Service). The solution involves setting up a standby system at a geographically different Oracle Cloud data center than the production site. Although the standby system may have 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 and is activated when the primary site becomes unavailable. This deployment model is sometimes referred to as an active-passive model.

Oracle WLS for OCI Disaster Recovery provides the best 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 (DR) 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 DR for Oracle WLS for OCI. The Oracle WLS for OCI Disaster Recovery solution is achieved by replicating the configuration files of the Oracle WebLogic Server domain and the runtime and metadata in the database used. The application may require additional configuration files to be replicated. Options are provided in this paper to suit different application paradigms. Disaster protection for Oracle Database Cloud Service used by Oracle WLS for OCI is provided through Oracle Data Guard.
This paper is intended for a technical audience having knowledge of Oracle Cloud Infrastructure, Oracle WebLogic Server, Oracle Database, Data Guard, and Oracle Database backup and recovery.
DISASTER RECOVERY SOLUTION FOR ORACLE WLS FOR OCI

Topology description

The Disaster Recovery solution for Oracle WLS for OCI is an active-passive model. There is a primary system consisting on an Oracle WLS for OCI domain, load balancer, and Oracle Cloud Infrastructure DB system in one region, and a secondary system consisting on Oracle WLS for OCI domain, load balancer, and Oracle Cloud Infrastructure DB system in a different region. This same topology may be implemented in the scope of a single Oracle Cloud Data Center with multiple availability domains although it is strongly recommended to use different regions for maximum Disaster recovery protection.

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 (far enough) geographically separated not be affected by the same disaster event. For example: Ashburn and Phoenix are two different data centers, sites or regions in context of this paper.

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

The secondary Oracle WebLogic Server domain is a replica of the primary domain, using the same name, schemas, passwords, etc. but pointing to the secondary database. The listener addresses of the WebLogic Servers are configured with the primary midtier host names, so in secondary midtier hosts the pertinent aliases are created in the hosts file to resolve them with the secondary IPs.

On the front-end, there is a unique name configured to access the applications running in the system. This “virtual” front-end name will point to the IP of the OCI Load Balancer of the primary site. In case of a switchover, this front-end name is updated to point to the IP of the OCI Load Balancer of the secondary site. It always must point to the LBR IP of the site that has the primary role in each time.

In normal business operation, the standby database is a physical standby. It either is in 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. For some actions, during the DR setup and lifecycle steps described in this document, the standby database will be converted to a snapshot standby. A snapshot standby database is a fully updateable standby database created by

---

1 WLS for OCI provides out-of-the-box protection for the middle tier against Availability Domain’s failures in regions with more than one Availability Domain. This protection requires that WLS for OCI be deployed on a regional subnet. When deploying in this type of network, the nodes in the WLS cluster are distributed evenly across all available availability domains. Notice, however, that Oracle’s Database does not support placing instances of the same RAC cluster in different availability domains, so it is still necessary to use Oracle Data Guard to protect the Database tier against failures at the Availability Domain level.
converting a physical standby database into a snapshot standby database. A database in **snapshot standby** mode is fully updateable database. A snapshot standby database 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.

All the information that resides in the database (OPSS information, custom schemas, TLOGs, JDBC persistent stores, etc.) is automatically replicated to the secondary site by Data Guard. The WLS domain configuration, located on the local filesystem in each site, needs to be replicated from primary to secondary also. This replication is required and performed during the initial DR setup, and also necessary during the system’s lifecycle, whenever configuration change maintenance is performed in the primary domain. Oracle supports three different methods to perform this WebLogic domain configuration replication: **DBFS (Oracle Database File System), OCI File Storage Service (FSS) with rsync, and Block Volume Cross-Region Replication**. All these methods use the same topology and similar mechanics. The difference 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 replicated to the secondary site via Data Guard. A DBFS mount is used as an assistance (or “staging”) filesystem. A DBFS mount is a filesystem that resides in the database, and can be mounted like an NFS mount. The primary domain configuration is copied to that DBFS mount, and this is automatically replicated to standby via the underlying Data Guard. In the secondary site, the same DBFS mount can be mounted from the secondary db by the secondary midtiers, and then, the domain config is copied from the DBFS mount to the secondary domain. This paper provides a script that automatizes this process in primary and in standby.

This method takes advantage of the **robustness of the Data Guard replica**. It can be used in any scenario, and it is strongly recommended for DR scenarios that have medium or high latencies between regions.

---

**Figure 2 Complete WLS for OCI DR topology, using DBFS based method for WLS Domain config replication**
In the **OCI File Storage Service (FSS) with rsync method**, the domain configuration is transferred to the secondary site using direct rsync between two file systems, one in primary and another in standby. This approach uses, like the DBFS one, a filesystem as an intermediate “staging” point. For this, an FSS (File Storage Service) mount is used in each region. To replicate the primary domain config, the WLS domain is copied first to the local/staging FSS mount, and then, via rsync, to the remote FSS mount. Then, in secondary, the domain configuration is copied from the secondary FSS to the secondary domain directory. This paper provides a script that automatizes this process in primary and in standby.

This method is recommended only when the latency between the data centers is low and the connection is reliable, as it is the case between Oracle OCI regions that communicate using Dynamic Routing Gateways.
The **DBFS method** delivers better availability through Oracle Driver’s retry logic and provides a more resilient behavior than FSS with rsync. It can be used in any scenario, and it is strongly recommended for DR scenarios that have medium or high latencies between the data centers. However, using DBFS for configuration replication has also additional implications from the setup, database storage and lifecycle perspectives.

The **FSS with rsync method** is easier to maintain and configure. However, it is recommended only in DR scenarios between Oracle OCI data centers using Dynamic Routing Gateways. Data centers communicating over the public internet, for example, may not have sufficiently low latency for a reliable behavior of FSS with rsync. Notice also that the FSS with rsync method can incur in additional cost 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 in order to get details on this).

Regardless which method is used to replicate the domain configuration, note that if the standby database is shutdown 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 if 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 to reduce billing, however, the WLS domain configuration changes that are replicated from the primary site will not be pushed to the secondary domain configuration if they are not started. In case of a switchover event, the RTO is increased if the midtier hosts need to be started and synchronized with primary changes. Oracle recommends to have the secondary midtier hosts up (with manager servers stopped). Check About having compute instances stopped in standby site for more details.

- There is a third DR model based in **Block Volume Cross-region replication**. In this approach, 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. This capability allows you to perform ongoing automatic asynchronous replication of block volumes and boot volumes to other regions. No stage location is used for configuration replication, hence the set up and ongoing replication process differs significantly from the DBFS and FSS approaches. This model provides worse RTO, and the switchover operations are more complex than in DBFS and FSS with rsync methods. However, it provides 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 also provides a continued and automated replica. See the Appendix F – Disaster Recovery Based On Block Volume Cross-Region Replication for specific details on advantages and disadvantages, topology and setup of this model.

## 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**

![Figure 5 OCI FSS with rsync method for WLS domain config replication logical flow diagram](image)

---

The **DBFS method** delivers better availability through Oracle Driver’s retry logic and provides a more resilient behavior than FSS with rsync. It can be used in any scenario, and it is strongly recommended for DR scenarios that have medium or high latencies between the data centers. However, using DBFS for configuration replication has also additional implications from the setup, database storage and lifecycle perspectives.

The **FSS with rsync method** is easier to maintain and configure. However, it is recommended only in DR scenarios between Oracle OCI data centers using Dynamic Routing Gateways. Data centers communicating over the public internet, for example, may not have sufficiently low latency for a reliable behavior of FSS with rsync. Notice also that the FSS with rsync method can incur in additional cost 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 in order to get details on this).

Regardless which method is used to replicate the domain configuration, note that if the standby database is shutdown 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 if 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 to reduce billing, however, the WLS domain configuration changes that are replicated from the primary site will not be pushed to the secondary domain configuration if they are not started. In case of a switchover event, the RTO is increased if the midtier hosts need to be started and synchronized with primary changes. Oracle recommends to have the secondary midtier hosts up (with manager servers stopped). Check About having compute instances stopped in standby site for more details.

There is a third DR model based in **Block Volume Cross-region replication**. In this approach, 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. This capability allows you to perform ongoing automatic asynchronous replication of block volumes and boot volumes to other regions. No stage location is used for configuration replication, hence the set up and ongoing replication process differs significantly from the DBFS and FSS approaches. This model provides worse RTO, and the switchover operations are more complex than in DBFS and FSS with rsync methods. However, it provides 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 also provides a continued and automated replica. See the Appendix F – Disaster Recovery Based On Block Volume Cross-Region Replication for specific details on advantages and disadvantages, topology and setup of this model.

## 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 System is used. When the database is a RAC DB System, 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 paper².

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 is able to 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 paper, and should be provided by the specific LDAP product.

**Load Balancer**

The Disaster Recovery solution assumes that the Oracle WLS for OCI 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 Oracle WLS for OCI 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. 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.

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.

Similar topology is used in a private OCI load balancer, which can be used to isolate your system from internet, although it only uses one subnet to host primary and standby. It 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 located 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).

See documentation for OCI Load Balancing for additional details.

This configuration is sufficient for the disaster recovery configuration. No config replication is required between primary and secondary region’s Load Balancers, as each one needs to route only to its local WebLogic cluster. Only in case that the default configuration is modified in primary site load balancers (for example, by adding listeners, backends, or certificates), should it be manually modified on the secondary region load balancer.

---

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.
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 “JFR-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.

Per the above, the Disaster Recovery solution assumes that the Oracle WLS for OCI is configured with an Oracle Cloud Infrastructure DB System and this document applies to that scenario. This enables the usage of some useful database features like Oracle Database File System (DBFS). Along with Data Guard, DBFS is used during the initial disaster recovery setup and during the lifecycle management, to replicate the WebLogic Server domain configuration from primary to secondary domain in the DBFS based method.

When the DB System used by WLS for OCI is a RAC Database (this is, when it has more than 1 db node), see the Appendix A – Considerations for RAC DB Systems for specific instructions and best practices.

Note that Oracle Autonomous Processing (ATP) is out of the scope of this document. Although ATP (ATP-D) now supports cross-region Data Guard, it does not provide a number of features required for PaaS DR, (like snapshot standby conversion, dgbroker access and others) Oracle ATP can't be used in WLS for OCI disaster recovery topologies.
Requirements

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

Front-end address

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

It is possible to reuse the existing system’s front-end host name address (if such exists already) as the virtual front-end for disaster protection, i.e. 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.

Appropriate DNS services (Oracle Cloud DNS, other commercial DNS, local DNS or local hosts resolution) need to be used for the virtual front-end name to be mapped to either site. Later in this document it is explained how to configure WebLogic Cloud to use the virtual front-end name.

WebLogic Resource Name Prefix

When you provision a WLS for OCI, you need to provide the “Resource Name Prefix”. This property, limited to 16 chars, is used to construct the names of all the resources: the WebLogic Server domain name, the cluster name, the WebLogic servers, the VMs’ hostnames, etc.

This property must be set to 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 example for the recovery of JMS messages and Tlogs. It also simplifies customizations and operations in both sites.

There is no restriction to use 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. Secondary middle tier hosts need to communicate with the primary database for the initial setup also.

When using the FSS with rsync method approach, WebLogic Administration host at each site need to be permitted to communicate via ssh (TCP/22) with the remote WebLogic Administration host for the rsync copy to function properly.

The communication between primary and secondary sites can be through Oracle internal networks, by using Dynamic Routing Gateway, which is the recommended approach (refer to the Dynamic Routing Gateway documentation for additional details on the network configuration). The communication between sites can also happen over an Internet Gateway (Oracle Net’s traffic is encrypted), but this is not the recommended approach. Depending on which method is used, the appropriate ingress rules need to be enabled. Security rules are configured in the Security Lists for each Virtual Cloud Network (in the OCI console, Core Infrastructure > Networking Section > Virtual Cloud Network). More information about this is available in Security Rules section on the OCI documentation.

The amount of data replicated depends on the redo generated by the primary database, and this is directly related with application load, its transactionality, concurrency, etc. The overhead of the DBFS for replicating configuration is typically irrelevant compared to the runtime data that Data Guard synchronizes. To ensure a timely delivery of the redo log files to the secondary 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 provides also a consistent latency. See Oracle Cloud Infrastructure Data Center Regions for more details.

Staging filesystems for the WebLogic config replication

Two different methods are are included in this document to replicate the WLS domain configuration between sites (DBFS, and FSS with rsync). Both methods require an assistance filesystem, DBFS or FSS respectively, to be mounted in the WebLogic hosts. In the next sections of this document, more details and specific instructions are provided to configure the staging filesystem in each case.
**Custom files**

Most of the WebLogic Server domain configuration that WebLogic Cloud uses is synced initially across sites with the following considerations:

- Each WebLogic system will maintain the original JDBC URLs used to connect to their local DB even after the DR set up has completed. Only the schema prefix will be altered so that both locations point to the same schemas (primary schemas).
- All the configuration under `weblogic_domain_name/config` is automatically distributed, by the Weblogic Infrastructure, to the other nodes in the same site by the WebLogic cluster features.
- Custom application deployments (`ear/war files`, `deployment plans`, etc.) and everything residing under the Administration Server WLS domain directory (except temp data) is synced initially across sites using the procedures described in this paper.

In the next sections of this document, more details are provided about how the configuration replica is performed. In case that customer has any other data residing in other node's or outside the domain directory for the WebLogic Administration Server, it will have to be manually copied to the secondary location.
DISASTER RECOVERY SETUP

In this document it is assumed that, as starting point, the primary site, consisting of an OCI DB system, an Oracle WLS for OCI system, and the Load Balancer (OCI LBR) is live. The secondary DR configuration, residing in a geographically remote site, will be created for this existing primary system.

Since the primary 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).

The following flow chart describes the steps of the Disaster Recovery setup process for the three models described in this paper:

Figure 6 Flow chart of the DR setup steps for the models described in this paper
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 next points in this section explain how to perform the DR setup for the DBFS and the FSS with RSYNC models. This is a summary of the steps for the set up process:

1. **Choose a virtual front-end name**
2. **Prepare Primary mid-tier for the virtual front-end**
3. **Setup the Database in Secondary Site**
4. **Provision WLS for OCI in Secondary Site**
5. **Prepare Secondary mid-tier for the virtual front-end**
6. **Configure required host aliases**
7. **Configure the staging mounts for WebLogic domain config replication**
8. **Run DR setup script in Primary WebLogic Domain**
9. **Run DR setup script in Secondary WebLogic Domain**
10. **Complete DR configuration when IDCS authentication is used**

Since the **DR model based in Block Volume Cross-region replication differs significantly from the other two** (it does not use a staging location) it is explained separately in [Appendix F – Disaster Recovery Based On Block Volume Cross-Region Replication](#).

### 1. Choose a virtual front-end name

When an Oracle WLS for OCI instance is created, the Load Balancer provisioned listens on a specific front-end IP address and no front-end name is provided nor configured in the system. Primary site LBR listens in one front-end IP, and secondary site LBR will listen in another front-end IP.

In a disaster recovery topology, the clients must access the system using a URL with a “Cloud region or data center” agnostic front-end FQDN, 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 have to **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

You need to perform some 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 site should always resolve its 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 WebLogic hosts. Example:

```
[oracle@wlsociprefix-wls-o ~]$ 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 wls hosts must not be altered when there is a switchover or failover. Primary WebLogic 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 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.

![Cluster front-end host configuration](image)

**Figure 7 Cluster front-end host configuration**

- **Update t3/rmi urls (if used) with cluster syntax**
The urls used for RMI invocations in the WebLogic cluster need to be agnostic also 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.

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. Setup the Database in Secondary Site

The database in the secondary site is created as a **Data Guard physical standby of the primary database**. There are 2 options to do 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 using the OCI Console (option1)**. This way, it is integrated with the OCI Console User Interface and allows you to use the Console to manage Oracle Data Guard associations in your DB system. It also provides out of the box configuration for backups in the Data Guard. Follow the point **Option 1) Configuring the 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 (please refer to the Oracle Cloud Infrastructure Documentation Using Oracle Data Guard for checking the availability of the Data Guard across regions feature

---

3 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.
in each DB Systems flavor/edition), you can still configure the Data Guard manually using scripts provided in this paper. Follow steps described in Option 2) Configuring the Data Guard manually for this.

### 3.1 Option 1) Configuring the Data Guard using OCI Console

When enabling Data Guard using 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 in different regions 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. You can 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:

<table>
<thead>
<tr>
<th>DB SYSTEM CONFIGURATION PROPERTY</th>
<th>EXISTING PRIMARY DB SYSTEM / EXAMPLE</th>
<th>SECONDARY DB SYSTEM / EXAMPLE</th>
<th>REQUIREMENT FOR AUTOMATED DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Cloud Tenancy</td>
<td>XXXX / paasma</td>
<td>YYYY / paasma</td>
<td>must be the same</td>
</tr>
<tr>
<td>Compartment</td>
<td>XXXX / wlsdr</td>
<td>XXXX / wlsdr</td>
<td>must be the same</td>
</tr>
<tr>
<td>Region</td>
<td>XXXX / Ashburn</td>
<td>YYYY / Phoenix</td>
<td>must be different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(recommended different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>regions for DR)</td>
</tr>
<tr>
<td>Availability Domain</td>
<td>XXXX / efEXT-US-ASBURN-AD1</td>
<td>YYYY / efXT-PHX-AD-1</td>
<td>must be different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(expected different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>regions, connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>via remote VCN peering)</td>
</tr>
<tr>
<td>DB System Name</td>
<td>XXXX / drdbwlmp1a</td>
<td>YYYY / drdbwlmp1b</td>
<td>must be different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(expected different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>regions, connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>via remote VCN peering)</td>
</tr>
<tr>
<td>Shape</td>
<td>XXXX / VM.Standard2.1</td>
<td>XXXX / VM.Standard2.1</td>
<td>must be the same</td>
</tr>
<tr>
<td>Virtual Cloud network</td>
<td>XXXX / wlsdrvcn1ash</td>
<td>YYYY / wlsdrvcn1pho</td>
<td>must be different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(expected different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>regions, connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>via remote VCN peering)</td>
</tr>
<tr>
<td>Client subnet</td>
<td>XXXX / wlsdrvcn1ashAD1</td>
<td>XXXX / wlsdrvcn1phoAD1</td>
<td>must be different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(expected different</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>regions, connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>via remote VCN peering)</td>
</tr>
<tr>
<td>Hostname Prefix</td>
<td>XXXX / drdbwlmp1a</td>
<td>YYYY / drdbwlmp1b</td>
<td>must be different</td>
</tr>
<tr>
<td>Administrator password</td>
<td>XXXX / acme1234#</td>
<td>XXXX / acme1234#</td>
<td>must be the same</td>
</tr>
</tbody>
</table>

### 3.2 Option 2) Configuring the Data Guard manually

It is required to configure Data Guard manually when it is not possible to use the same cloud tenancy for primary and standby or when the automated Data Guard option provided by OCI is not enabled for the DB flavor and/or locations involved in the DR configuration. In this case, the secondary database must be first provisioned as a regular DB System and then it will be configured as the standby by executing some setup scripts provided in this document, as explained in these steps:
### 3.2.1 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 section 4 Provision WLS for OCI in Secondary Site.*

When configuring the Data Guard manually, you first need to provision the secondary database, using the same Database name, PDB name, release, patch level 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:

<table>
<thead>
<tr>
<th>DB SYSTEM CONFIGURATION PROPERTY</th>
<th>EXISTING PRIMARY DB SYSTEM / EXAMPLE</th>
<th>SECONDARY DB SYSTEM / EXAMPLE</th>
<th>REQUIREMENT FOR DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Cloud Tenancy</td>
<td>XXXX / paasmaa</td>
<td>YYYY / paasmaa</td>
<td>can be different</td>
</tr>
<tr>
<td>Compartment</td>
<td>XXXX / wlsdr</td>
<td>YYYY / wlsdr</td>
<td>can be different</td>
</tr>
<tr>
<td>Region</td>
<td>XXXX / Ashburn</td>
<td>YYYY / Phoenix</td>
<td>must be different (recommended different regions for DR)</td>
</tr>
<tr>
<td>Availability Domain</td>
<td>XXXX / efEXT-US-ASBURN-AD1</td>
<td>YYYY / efXT-PH-AD-1</td>
<td>must be different</td>
</tr>
<tr>
<td>DB System Name</td>
<td>XXXX / drdbwlmp1a</td>
<td>YYYY / drdbwlmp1b</td>
<td>must be different</td>
</tr>
<tr>
<td>Shape</td>
<td>XXXX / VM.Standard2.1</td>
<td>XXXX / VM.Standard2.1</td>
<td>must be the same</td>
</tr>
<tr>
<td>Total node count</td>
<td>N / 1</td>
<td>N / 1</td>
<td>must be the same</td>
</tr>
<tr>
<td>Available storage</td>
<td>XXXX / 256</td>
<td>XXXX / 256</td>
<td>must be the same</td>
</tr>
<tr>
<td>License type</td>
<td>LI, BYOL / BYOL</td>
<td>LI, BYOL / BYOL</td>
<td>can be different</td>
</tr>
<tr>
<td>SSH public key</td>
<td>XXXX</td>
<td>YYYY</td>
<td>can be different</td>
</tr>
<tr>
<td>Virtual Cloud network</td>
<td>XXXX / wlsdrvcn1ash</td>
<td>YYYY / wlsdrvcn1pho</td>
<td>must be different (expected different regions, recommended connect via remote VCN peering)</td>
</tr>
<tr>
<td>Client subnet</td>
<td>XXXX / wlsdrvcn1ashAD1</td>
<td>XXXX / wlsdrvcn1phoAD1</td>
<td>must be different</td>
</tr>
<tr>
<td>Hostname Prefix</td>
<td>XXXX / drdbwlmp1a</td>
<td>YYYY / drdbwlmp1b</td>
<td>must be different</td>
</tr>
<tr>
<td>Database Name</td>
<td>XXXX / ORCL</td>
<td>XXXX / ORCL</td>
<td>must be the same</td>
</tr>
<tr>
<td>Database Version</td>
<td>XXXX / 19c</td>
<td>XXXX / 19c</td>
<td>must be the same</td>
</tr>
</tbody>
</table>
PDB name | XXXX / PDB1 | XXXX / PDB1 | must be the same
--- | --- | --- | ---
Administrator password | XXXX / acme1234# | XXXX / acme1234# | must be the same
Enable automatic backups | X / Checked | Y / unchecked | must be disabled in stby

**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 both are at the same patch level. More precisely, a Data Guard configuration requires a fix for bug 22611671 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.

**Note for RAC scenarios:** see the appendix Appendix A – Considerations for RAC DB Systems.

### 3.2.2 Configuring the Data Guard between primary and secondary

**Note:** In case that the Data Guard has been enabled using the OCI Console, these steps must be skipped and you can continue with section 4 Provision WLS for OCI in Secondary Site.

In order to configure the Data Guard manually between the primary and secondary databases, follow these steps.

a) Primary and standby databases in a Data Guard need to communicate each other on the listener 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. You can verify this communication using nc command (use the public/private IPs depending on your network topology).
   - From primary to secondary:
     ```bash
     [opc@drdDBa ~]$ nc -vw 5 -z <secondary_db_ip> 1521
     ```
   - From secondary to primary:
     ```bash
     [opc@drdDBb ~]$ nc -vw 5 -z <primary_db_ip> 1521
     ```
   - Example for correct output:
     ```bash
     [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 in case that primary and secondary sites use Internet Gateway to communicate each other, and use internal DB System’s IPs in case the communication between primary and secondary VCNs is done internally, using a Dynamic Routing Gateway.

b) Download the `dataguardit_primary.sh` script and upload to the primary Database node.

c) Open the script as `oracle` user, read its instructions and customize the variables explained in its initial section. Once customized, execute the script as `oracle` user. As an output, it will create a tar file that needs to be copied from primary to standby. The location where the tar is created can be customized with a script variable.
d) **Copy** the tar file to the standby database node. Make sure that oracle user has read rights on the file. If it was uploaded with opc user, to make it readable by others. Example:

```
[opc@drDBb ~]$ chmod o+r /tmp/ORCL.GZ
```

e) **Download the** dataguardit_standby_root.sh **script and upload to the secondary Database node.**

f) **Open the script as root user**, read its instructions and **customize** the variables explained in its initial section.

g) Once customized, **execute** the script **as root user**. The script will **delete the existing database** instance and will create a new one duplicating the primary. It will also set up the database listener and configuration required for Oracle Data Guard broker.

h) **Monitor the execution of the script and check for any errors.** The script will create a log file (/tmp/dataguardit_date.log). Check this log for troubleshooting information. The script can be executed again in case of failure (it will do the cleanup of any previous failed executions).

i) **After the script completes,** 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_12:09:
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. Provision WLS for OCI in Secondary Site

Secondary WebLogic will be created pointing to the secondary DB system, which must be open in snapshot standby mode.

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

```
[oracle@drdbwlmp1a ~]$ 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 Oracle Cloud Infrastructure documentation](https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/patched-included/oracle-weblogic-cloud.html) to create the secondary site WebLogic Cloud system pointing to the secondary DB System.

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/switchover behavior.

Make sure that the WLS for OCI version and patch level to be provisioned in the secondary location matches the one running in the primary site.4

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

<table>
<thead>
<tr>
<th>WEBLOGIC CLOUD PROPERTY</th>
<th>VALUE IN PRIMARY / EXAMPLE</th>
<th>VALUE IN SECONDARY / EXAMPLE</th>
<th>REQUIREMENT FOR DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>XXXX / Ashburn</td>
<td>YYYYY / Phoenix</td>
<td>Must be different</td>
</tr>
</tbody>
</table>

---

4 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. See [https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/patched-included-oracle-weblogic-cloud.html](https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/patched-included-oracle-weblogic-cloud.html)
<p>| <strong>Version</strong> | XXXX / 12.2.1.4.200414.03 | XXXX / 12.2.1.4.200414.03 | must be the same |
| <strong>Stack Name</strong> | XXXX / wlsociprefixprim | YYYY / wlsociprefixstby | can be different |
| <strong>Resource Name Prefix</strong> | XXXX / wlsociprefix | XXXX / wlsociprefix | must be the same |
| <strong>WebLogic Server Shape</strong> | XXXX / VM.Standard2.1 | XXXX / VM.Standard2.1 | must be the same |
| <strong>WebLogic Server Node Count</strong> | N / 2 | N / 2 | must be the same |
| <strong>Admin User Name</strong> | XXXX / weblogic | XXXX / weblogic | must be the same |
| <strong>Admin Password (secret OCID)</strong> | XXXX / ocid1.vaultsecret.oc1.uk-london-1.amaaaaaaaj4,... | YYYY / ocid1.vaultsecret.oc1.eu-frankfurt-1.amaa | must be the same password (although the secret OCID could be different because the secondary vault will be in the secondary region) |
| <strong>WebLogic Server Instance Configuration (ports)</strong> | XXXX | XXXX | must be the same |
| <strong>Network Compartment</strong> | XXXX / wlsdr | YYYY / wlsdr | can be different |
| <strong>VCN</strong> | XXXX / wlsvcnSite1 | YYYY / wlsvcnSite2 | must be different |
| <strong>Subnet</strong> | XXXX / wlsvcnSite1Subnet1 | YYYY / wlsvcnSite2Subnet1 | must be different |
| <strong>Provision Load Balancer</strong> | Must be checked | Must be checked | checked in both cases |
| <strong>Enable authentication using IDCS</strong> | Check or Unchecked | Check or Unchecked | must be the same |
| (if IDCS is checked) <strong>IDCS host domain name</strong> | XXXX/identity.oraclecloud.com | XXXX/identity.oraclecloud.com | must be the same |
| (if IDCS is checked) <strong>IDCS port</strong> | XXXX / 443 | XXXX / 443 | must be the same |
| (if IDCS is checked) <strong>IDCS tenant</strong> | XXXX/ids-c-5e890de5903jr... | XXXX/ids-c-5e890de5903jr... | must be the same |
| (if IDCS is checked) <strong>IDCS Client ID</strong> | XXXX / f88aa64hh8hgd... | XXXX / f88aa64hh8hgd... | must be the same |
| (if IDCS is checked) <strong>Secrets OCID for IDCS Client Secret</strong> | XXXX / uk-london-1.amassshdjhdjfd... | YYYY / eu-frankfurt-1.jhjhfjhfjfdfjfd... | IDCS Client Secret (unencrypted value) must be the same, but the OCID of the vault secret can be |</p>
<table>
<thead>
<tr>
<th><strong>Table: Oracle WebLogic Server for Oracle Cloud Infrastructure Disaster Recovery</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(if IDCS is checked)</strong></td>
</tr>
<tr>
<td><strong>OCI Policies</strong></td>
</tr>
<tr>
<td><strong>Provision with JRF</strong></td>
</tr>
<tr>
<td><strong>Database Strategy</strong></td>
</tr>
<tr>
<td><strong>DB system compartment</strong></td>
</tr>
<tr>
<td><strong>DB system network</strong></td>
</tr>
<tr>
<td><strong>DB system</strong></td>
</tr>
<tr>
<td><strong>DB version</strong></td>
</tr>
<tr>
<td><strong>Database in the system</strong></td>
</tr>
<tr>
<td><strong>PDB</strong></td>
</tr>
<tr>
<td><strong>Database administrator</strong></td>
</tr>
<tr>
<td><strong>Database administrator password (secret OCID)</strong></td>
</tr>
</tbody>
</table>

**NOTE:** the Vault service is used only for encrypting and decrypting passwords during the provisioning. It is not used for WebLogic runtime or lifecycle tasks. You can use the same Vault and secret OCID for provisioning primary and secondary WebLogic Cloud services or use a different one. In this last case, the secrets OCID that are provided to the provisioning wizard will be different even if the clear password is the same.

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

---

5 See https://docs.oracle.com/en/cloud/paas/weblogic-cloud/user/you-begin-oracle-weblogic-cloud.html to manually create the dynamic group and policies
5. Prepare Secondary mid-tier for the virtual front-end

You need to 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 WebLogic hosts.

Example:

```bash
[oracle@wlsociprefix-wls-o -]$ more /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost6 localhost6.localdomain6 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 secondary cluster as that information will be copied from the primary WebLogic Domain configuration.

NOTE: the /etc/hosts file of the secondary WebLogic hosts must not be altered when there is a switchover or failover. Secondary WebLogic 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.

6. Configure required host aliases

The hostnames that are used as listen addresses for the WLS servers in primary need to be valid in the secondary location, but mapping to the secondary IPs. And the other way around: the hostnames used as listen address for the WLS servers in secondary need to be valid in the primary location but mapping to the primary IPs. To configure the required alias:

a) **First of all**, 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.

b) **Identify the Fully Qualified Domain names (FQDN)** of each WebLogic hosts in primary and standby where the server listen. You can look for the listen addresses in the domain configuration file. For example, in primary domain:

```bash
[oracle@wlsociprefix-wls-o config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-o config]$ grep listen-address config.xml
<listen-address>wlsociprefix-wls-o.wlsdrvcnfra1ad2.wlsdrvcnfra1.oraclevcn.com</listen-address>
<listen-address>long_and_short_hostnames_primary_node2> long_and_short_hostnames_secondary_node1>
<listen-address>long_and_short_hostnames_primary_node1> long_and_short_hostnames_secondary_node2>
```

And in the secondary domain:

```bash
[oracle@wlsociprefix-wls-o config]$ cd $DOMAIN_HOME/config
[oracle@wlsociprefix-wls-o config]$ grep listen-address config.xml
<listen-address>long_and_short_hostnames_secondary_node2> long_and_short_hostnames-primary_node1>
<listen-address>long_and_short_hostnames_secondary_node1> long_and_short_hostnames-primary_node2>
```

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

c) **Edit the /etc/hosts** (as root) in **all the primary mid-tier nodes** and add the FQDN (“long name” with domain names) of the standby hosts as aliases of the primary hosts. Each host should have entries as the following:

```bash
<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>
```

```bash
<IP_sec_node1> <long_and_short_hostnames_secondary_node1> <long_and_short_hostnames-primary_node1>
<IP_sec_node2> <long_and_short_hostnames_secondary_node2> <long_and_short_hostnames-primary_node2>
```
Note: “shortnames” are expected to be the same in primary and stby hosts
Example of the resulting entries /etc/hosts in primary mid-tier hosts:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost5 localhost6.
::1   localhost localhost6.localdomain6.

# Front-end virtual name

10.0.2.1 0.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com wlsociprefix-wls-o wlsociprefix-wls-0 wlsociprefix-wls
10.0.2.11 wlsociprefix-wls-1.wlsdrvcnlon2ad2.wlsdrvcnlon2.oraclevcn.com wlsociprefix-wls-1 wlsociprefix-wls-1.
```

```
# For WLS DR
```

```
d)  In the same way, edit the /etc/hosts (as root) in all the secondary mid-tier nodes and add FQDN ("long name"
with domain names) from primary hosts as aliases of the secondary hosts. Each host should have entries as the
following:

```
<IP_secondary_nodes> <long_and_short_hostnames_secondary_node1> <long_and_short_hostnames_prim_node1>
<IP_secondary_nodes2> <long_and_short_hostnames_secondary_node2> <long_and_short_hostnames_prim_node2>
```

FQDN ("long name" with domain names) and hostnames ("short name", without domain)
Example of the resulting /etc/hosts in secondary mid-tier hosts:

```
10.1.2.5 wlsociprefix-wls-0.wlsdrvcnfrad2.wlsdrvcnfrad2.oraclevcn.com wlsociprefix-wls-0 wlsociprefix-wls-0
10.1.2.4 wlsociprefix-wls-1.wlsdrvcnfrad2.wlsdrvcnfrad2.oraclevcn.com wlsociprefix-wls-1 wlsociprefix-wls-1.
```

NOTE FOR RAC: When the database used is a RAC database, this is a good point to implement the recommendation described in the Appendix A – Considerations for RAC DB Systems for using a clusterware database service for the PDB rather than the default PDB service, before continuing with the Disaster Recovery configuration. If you implement that best practice at this point, you can use the database clusterware service name in the next steps (alternatively, you can use the default PDB services names and implement the recommendation of using a clusterware service as a post step after DR setup).

7. Configure the staging mounts for WebLogic config replication
Using one of the two methods described earlier to replicate the WLS domain configuration, follow the steps described in this section to configure the desired option for your Disaster Recovery topology.

7.1 Option 1) Configure the DBFS mounts for DBFS based 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 also for replicating WebLogic domain configuration changes. 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 paper provides a script to automate this copy in primary and standby. The DBFS mount is not used for other WebLogic runtime operations related with disaster recovery, so it is not critical for the service nor has a big impact on the performance of the system.
There is no DBFS mount configured by default in the WebLogic hosts and it needs to be manually configured. 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:

```
[Select] DLP: Oracle Database 19.3.0.0 (Oracle Database In-Memory, Oracle Database Vault)
```

Then select the database client only:

```
[Select] Oracle Database 19.3.0.0
```

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

```
[Select] Oracle Database 19.3.0.0
```

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

b) **Download the script** dbfs_dr_setup_root.sh from the WLS for OCI DR utils and **upload** it to the mid-tier host. 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) This script requires some input parameters: you need to provide the connection data used to connect to the local database used by the WLS (i.e.: 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 the domain. 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>
```

**NOTE:** It is assumed and expected that the standby database is in snapshot standby at this point.

Example to run it primary midtier hosts. You have to provide primary PDB values:

```
./dbfs_dr_setup_root.sh drdbwlmrp2a.scan.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com 1521
PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com acme1234# /u01/install/V982064-01.zip
```

Example to run it secondary midtier hosts. You have to provide secondary PDB values:

```
./dbfs_dr_setup_root.sh drdbwlmrp1b.scan.wlsdrvcnfa1ad2.wlsdrvcnfa1.oraclevcn.com
1521 PDB1.wlsdrvcnfa1ad2.wlsdrvcnfa1.oraclevcn.com acme1234# /u01/install/V982064-01.zip
```

**NOTE:** If the database is a RAC Database, see Running the dbfs_dr_setup_root.sh in RAC scenarios in Appendix A for specific instructions.

d) You can 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
  dbfsdir
```

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. If you use tuples, the username is the same.
- 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 rerunning, there can be warnings because some things are already created (db user, tablespace, etc.), but these messages can be ignored.

### 7.2 Option 2) Configure the FSS mounts for FSS with rsync method

When using the FSS with rsync method to replicate the WebLogic config, FSS filesystems need to be created in the primary and secondary sites. They are separated FSS filesystems, each one is mounted by the local hosts only (no cross-region NFS access to the remote FSS). They store a copy of the domain config and are not used for runtime. These filesystem are used as staging mounts to replicate the content between them using remote rsync commands.

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

Follow these steps to configure and mount the OCI FSS mounts:

1. **Prepare FSS Mounts**
   - 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

   **Sample Values in Primary**
   - New Mount Target Name: `wlsociprefixPri_mt`
   - Availability Domain: `<same than primary wls>`
   - Virtual Cloud Network: `<same than primary wls>`
   - Subnet: `<same than primary wls>`

   **Sample Values in Secondary**
   - New Mount Target Name: `wlsociprefixStby_mt`
   - Availability Domain: `<same than secondary wls>`
   - Virtual Cloud Network: `<same than secondary wls>`
   - Subnet: `<same than secondary wls>`

2. **Create a file system (FSS) 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

   **Sample Values**
   - Name: `wlsociprefixPri_fss`
   - Availability Domain: `<same than primary wls>`
   - Export: `/fssmount`
   - Mount target: `wlsociprefixPri_mt` (previously created)

   **Sample Values**
   - Name: `wlsociprefixStby_fss`
   - Availability Domain: `<same than secondary wls>`
   - Export: `/fssmount`
   - Mount target: `wlsociprefixStby_mt` (previously created)
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3</strong></td>
<td>Validate/set the network security rules required for FSS mount. Some network rules in each subnet are required to allow the NFS traffic between hosts and mount target. Use the instructions in Configuring VCN Security Rules for File Storage to set up security rules correctly for your file systems.</td>
</tr>
</tbody>
</table>
| **4** | Mount the file system in the WLS servers. In ALL WLS hosts:
- Create the local mount point
  ```
  # sudo mkdir /fssmount
  ```
  **NOTE:** Use /fssmount as the mount point. That is the value expected as the mount point by the DR setup scripts.
- With user root, edit /etc/fstab and add the mount, using the **appropriate** 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 
  ```
  **Reference:** Mounting File Systems From Unix-Style Instances

|   | Example line to add in primary hosts’ /etc/fstab (must be in one line):
|   | 10.1.0.49:/fssmount /fssmount nfs defaults,nofail,nosuid,resvport 0 0
|   | Example line to add in secondary hosts’ /etc/fstab (must be in one line):
|   | 10.2.0.52:/fssmount /fssmount nfs defaults,nofail,nosuid,resvport 0 0 |
| **5** | Verify mounted file system. In all WLS hosts:
|   | ```
|   | # df -h | grep fssmount
|   | # ls -la /fssmount
|   | ```
|   | **Sample result in primary WLS hosts:**
|   | [root@wlsociprefix-wls-0 opc]# df -h | grep fssmount
|   | 10.1.0.49:/fssmount 8.0E o 8.0E 0% /fssmount
|   | **Sample result in secondary WLS hosts:**
|   | [root@wlsociprefix-wls-0 opc]# df -h | grep fssmount
|   | 10.2.0.52:/fssmount 8.0E o 8.0E 0% /fssmount |
| **6** | Create the folders that will be used later. - Create them with user oracle (run these commands in one of the WLS hosts in each region):
| | ```
| | # sudo chown oracle:oracle /fssmount
| | # sudo su - oracle
| | # mkdir -p /fssmount/domain_config_copy
| | ```
| | **NOTE:** Create the folder as described, with the name /fssmount/domain_config_copy. That is the value expected by the DR setup scripts.
| | **Sample result in primary WLS hosts:**
| | Run this in primary admin host (FSS mount is shared by the rest of primary WLS hosts so no need to repeat)
| | Run this in secondary admin host (FSS mount is shared by the rest of secondary WLS hosts so no need to repeat) |

8. Run DR setup script in Primary WebLogic Domain

Some actions need to be performed in the primary mid-tier Oracle WebLogic Administration Server host node. They are done by primary DR setup script **fmw_dr_setup_primary.sh**. 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) **Download** the script **fmw_dr_setup_primary.sh** from the WLS for OCI DR utils zip, **upload** to primary admin host, edit and customize.
b) **Run** the script as **oracle** user in the **primary admin host**.
It can be run interactively or non-interactively:

- **Interactive usage:**
  ```bash
  fmw_dr_setup_primary.sh
  ```
- **Non-interactive usage:**
  ```bash
  fmw_dr_setup_primary.sh <db_sys_password> <dr_method> [remote_admin_node_ip] [remote_keyfile]
  ```

Where:
- **db_sys_password:** is the password of the database sys user
- **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
- **remote_admin_node_ip:** Required only if **dr_method** is RSYNC. 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 go 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 primary WLS Administration server host and is readable by oracle user only.

Example when you are using the **DBFS based method**:
```bash
./fmw_dr_setup_primary.sh 'acme1234' DBFS
```

Example when you are using the **FSS with rsync method**:
```bash
./fmw_dr_setup_primary.sh 'acme1234' RSYNC 10.1.0.2 /u01/install/sshkeys/my_priv_key.ppk
```

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

### 9. Run DR setup script in Secondary WebLogic Domain

The **fmw_dr_setup_standby.sh** script performs the required operations to configure the secondary mid-tier cluster to act as standby for the primary one.

a) 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 db.

Note this is only required for setup and lifecycle operation. 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 DRS, depending on the network scenario:
```bash
java -classpath /u01/app/oracle/middleware/wlserver/server/lib/weblogic.jar utils.dbping ORACLE_THIN system
<system_password> <primary_db_private_or_public_ip>:1521/<primary_db_service>
```

b) **Download** the script **fmw_dr_setup_standby.sh** from the **WLS for OCI DR utils zip** and upload it to all the secondary mid-tier hosts.

c) **Stop** the secondary WebLogic servers (admin and managed) and node managers in all secondary mid-tier hosts.

d) **Connect** to primary DB and **convert** the standby database to **physical standby**:
```bash
[oracle@drdbwlmpl1a ~]$ 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)
```

e) **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. Primary db information must be provided to this script.

Interactive usage:

```
fmw_dr_setup_standby.sh
```

(Note: user will be prompted for all values)

Non-interactive usage:

```
fmw_dr_setup_standby.sh <A_DB_IP> <A_PORT> <PDB_SERVICE_PRIMARY> <SYS_DB_PASSWORD> <DR_METHOD>
```

Where:

- **A_DB_IP**: 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.
- **A_PORT**: the primary database listener port.
- **PDB_SERVICE_PRIMARY**: the primary PDB service name.
- **SYS_DB_PASSWORD**: 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

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

```
./fmw_dr_setup_standby.sh '10.0.2.2' '1521' 'PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com' 'acme1234#' DBFS
```

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

```
./fmw_dr_setup_standby.sh '10.0.2.2' '1521' 'PDB1.wlsdrvcnlon1ad2.wlsdrvcnlon1.oraclevcn.com' 'acme1234#' RSYNC
```

In DBFS DR method, the script will convert the database in snapshot standby mode so secondary servers 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.

---

**NOTE: If the database is a RAC Database, see Running the DR setup scripts in RAC scenarios in Appendix A for specific instructions**

f) **Start** the Node Manager in the node.

Example to start the node manager:

```
[oracle@wlsocprefix-wls-o -]s cd $DOMAIN_HOME/bin
[oracle@wlsocprefix-wls-o bin]$ nohup ./startNodeManager.sh $DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 &
```

g) **Start** the WebLogic servers that run in the node (admin server if this is the admin server node, and then the managed server in the node).

Example to start the Administration server using WLST:

```
[oracle@wlsocprefix-wls-o -]s cd $MIDDLEWARE_HOME/oracle_common/common/bin
[oracle@wlsocprefix-wls-o -]s ./wlst.sh
wlst> nmConnect ('weblogic','password','<admin_host_name>','5556','<domain_name>','/u01/data/domains/<domain_name>','SSL')
wlst> nmStart('admin_server_name')
```
The managed server in that node can then be started by accessing to the WLS Admin Console (Environments > Servers > Control)
h) **Verify** the server is started in RUNNING and check the sample app urls and administration consoles.
i) Repeat steps from “d)” (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, it is recommended that you **stop Admin and Managed servers** in the secondary domain and **convert** the **standby database back to physical standby**. Your secondary site is now ready for DR!

### 10. **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.

_Warning: 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 paper._

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`.

---

*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.*
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:

**a) 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:
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 of the Confidential Application is service_name + _confidential_idcs_app_ + timestamp
Example: wlsmkp7_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: 87b58e51a4744f95897a2fd7ae5c987
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:name>IDCSIntegrator</sec:name>
  ...
  <idcs:tenant>idcs-5e890de598888888888d70064c5e00718</idcs:tenant>
  <idcs:client-tenant>idcs-5e890de598888888888d70064c5e00718</idcs:client-tenant>
  <idcs:client-id>87b58e51a4744f95897a2fd7ae5c987</idcs:client-id>
</sec:authentication-provider>

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: wlsmkp7_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.

b) 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: wlsmkp7_app_gateway_2020-09-18T11:55:44.241334
Click in “Deactivate” to inactivate it.

NOTE: make sure to identify 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: 28dabe1c9de4410dab27b6e4febb2bf
Client Secret: f1f9589c-ee71-4a73-a2eb-f666f7bd6a5
Login SSH to the admin node in the secondary WLS for OCI domain.
o Stop the app gateway docker container, login as opc and run:

```
sudo docker stop appgateway
```

o 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
```

o 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.

o 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.
```

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-82507b399af654467dcdcf94029c895ab0"

o Regenerate the wallet and change the permissions 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
```

o Start the docker container of the app gateway again:

```
sudo docker start appgateway
```

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

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

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

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

To make sure that there are no more accesses 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: `wlsmkpl7enterprise_idcs_app_2020-09-1T11:55:44.241334`

- 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.

c) **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-5e890de598888e888888d70064c5e00718.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

To run this verification, 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/

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

d) **Verify the IDCS protected URLs in the secondary location**

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

At this point, the DR environment setup is completed. Oracle recommends to immediately validate that the DR setup is correct, by performing a complete Switchover (see point Switchover in next pages) or, alternatively, by opening the secondary site for validation (see point Open Secondary Site for Validation in next pages).
**LIFECYCLE PROCEDURES**

**Replicating configuration changes**

Any data residing 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. When a configuration change is done in the primary WebLogic domain (for example: a new deployment, a new datasource, a change in a deployment plan, etc.), the change must be replicated to the secondary domain in some way. Two main approaches can be used to maintain the same in both locations. The applicability of each depends on how frequently this “file-system-resident” configuration is modified:

a) For cases where the 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.

b) For cases where the domain configuration is **modified regularly**, a script that replicates the WLS domain configuration from primary to secondary can be used. The config_replica.sh script is provided for this. Depending on the method used during the DR Setup (DBFS or FSS with rsync method), the script will replicate the WLS domain config using the selected method. If the DR setup was done using the DBFS approach, Oracle Database File System (DBFS) will be used to synchronize the configuration using Data Guard. Otherwise, if the FSS with rsync method was used, rsync to remote site will be used to replicate the configuration.

Both approaches described in detail below:

**a) Apply domain configuration changes in both sites**

To maintain the file system configuration synchronized by repeating the config change in the secondary site, follow these steps:

<table>
<thead>
<tr>
<th>STEP</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apply the configuration change normally in the primary site</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Convert the standby database to a snapshot standby</td>
</tr>
<tr>
<td></td>
<td>Execute these steps as oracle user in the primary Database host:</td>
</tr>
<tr>
<td></td>
<td>[oracle@drdbwlm1a]$dgmgrlsys/your_sys_password@primary_db_unqnameDGMGRL&gt; CONVERT DATABASE secondary_db_unqname to SNAPSHOT STANDBY;</td>
</tr>
<tr>
<td></td>
<td>Converting database &quot;secondary_db_unqname&quot; to a Snapshot Standby database, please wait...</td>
</tr>
<tr>
<td></td>
<td>Database &quot;secondary_db_unqname&quot; converted successfully</td>
</tr>
<tr>
<td>3</td>
<td>Start (if it wasn’t started) the WebLogic Administration Server on the secondary site</td>
</tr>
<tr>
<td></td>
<td>Follow the steps in the Oracle Cloud documentation to start the administration server. It is important that ONLY the administration server and not the managed servers is started on the secondary location</td>
</tr>
<tr>
<td>4</td>
<td>Repeat the configuration change in the secondary site</td>
</tr>
<tr>
<td></td>
<td>Use the WLS Administration Console in the secondary location to apply the configuration change. Activate the change and verify that the change is working as expected.</td>
</tr>
<tr>
<td>5</td>
<td>Revert the database to physical standby</td>
</tr>
<tr>
<td></td>
<td>Execute this steps as oracle user in the primary Database host:</td>
</tr>
<tr>
<td></td>
<td>[oracle@drdbwlm1a]$dgmgrlsys/your_sys_password@primary_db_unqname DGMGRL&gt; CONVERT DATABASE secondary_db_unqname to PHYSICAL STANDBY;</td>
</tr>
<tr>
<td></td>
<td>Converting database &quot;secondary_db_unqname&quot; to a Physical Standby database, please wait...</td>
</tr>
<tr>
<td></td>
<td>Oracle Clusterware is restarting database “ORCL” ...</td>
</tr>
</tbody>
</table>
b) Using an script to propagate configuration changes

When the system’s lifecycle involves frequent updates to the domain file system, the process can be automated using a script (config_replica.sh) that replicates changes via DBFS or FSS with rsync, depending on the method chosen for the DR topology.

Option 1) DBFS method

In this approach, the DBFS file system created in step Configure the DBFS mount is used as an assistance file system where there is a copy of the primary site’s domain configuration. The information in this file system is automatically replicated to standby location via Data Guard. In the standby site, the DBFS file 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 a snapshot standby.

**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 in order to come up (the dependency is not only on the database but also on FUSE libraries, mount points, etc.).

Notice also that the WebLogic Server domain configuration cannot be copied “as is” in this paper’s design since each site in the domain has references to the local DB service in the JDBC connect strings. The configuration has to be modified after it is copied to each site.

The steps of this procedure are as follows:

- The primary WebLogic domain configuration directory contents are copied to the local DBFS file system. Files and folders that are irrelevant or not required (i.e. tmp 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 current production WLS domain configuration files are copied from the DBFS mount to the standby domain folder. Some file modifications are performed during this step (updating the db url to point to the secondary db).

![Figure 10 Replicating WebLogic domain configuration to standby WebLogic Cloud with DBFS based method](image.png)

The advantage of this procedure is that it relies in the robustness of the Data Guard replication to make the config updates available in the standby site. The replica direction is totally consistent with the roles of the sites and it automatically changes when there is a switchover or failover.
NOTE: The midtier mounts the dbfs mount by connecting to the local pdb database using 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 by customer if needed. Note that, if the PDB’s service is not reachable, operating system commands that retrieve info from the dbfs filesystem (like “df -h”, or an “ls” in the dbfs mount folders) will take long periods of time to return due to the retries.

Option 2) FSS with rsync method

The rsync command can be used 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 synchronized to the local FSS filesystem, then when complete, the local FSS filesystem is synchronized to the remote site’s FSS filesystem.
- On the secondary site, the domain configuration is synchronized from the local FSS filesystem to the WLS domain directory, then environment-specific configuration values are updated (the db url to point to the secondary db).

This procedure can be automated with the provided config_replica.sh script both for the DBFS and FSS with rsync methods. The same script is used in primary and standby and is valid for both methods. It checks the current role of the site and performs the required actions, depending on whether it is the primary or the secondary site and if the method is DBFS or FSS with rsync method.

Follow these steps to use the config_replica.sh script for replicating the WebLogic configuration:

1. The config_replica.sh script has the following 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 (when the script runs in standby role, it needs to convert standby database to snapshot in order 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 needs to connect to the remote admin host to perform the remote rsync copy.

   Make sure this communication is open by creating the appropriate rules. 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. Download/copy the config_replica.sh script from the WLS for OCI DR utils zip to the primary WebLogic Administration host and to the secondary WebLogic Administration host.
3. **Execute the script first in the primary WebLogic Administration host** (with oracle user). Monitor the execution and watch for any errors. The script will verify the DG status and will copy the domain configuration from the primary WebLogic domain to the secondary site (via DBFS or via FSS with rsync method).

Syntax is:
```
./config_replica.sh <DR_METHOD> [REMOTE_ADMIN_NODE_IP] [REMOTE_KEYFILE]
```

Where the parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR_METHOD</td>
<td>The method used for the config replica. 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</td>
</tr>
<tr>
<td>REMOTE_ADMIN_NODE_IP</td>
<td>Required if DR_METHOD is RSYNC, omit if DR_METHOD is DBFS. Specify the IP address of the remote site’s WLS Administration host. I.e: when you run the script in primary, provide the secondary WLS Administration host’s IP, and when you run the script in secondary, provide the primary WLS Administration host’s IP. Hostname/FQDN must NOT be provided here.</td>
</tr>
<tr>
<td>REMOTE_KEYFILE</td>
<td>Required if DR_METHOD is RSYNC, omit if DR_METHOD is DBFS. Is the complete path to the private keyfile required to ssh to remote site’s WebLogic Administration host. Make sure that the file has been uploaded to this host and is readable by oracle user only.</td>
</tr>
</tbody>
</table>

The script will prompt for the database sys password.

Example to run in the DBFS Method:
```
./config_replica.sh DBFS
```

Example to run in the FSS with rsync method:
```
./config_replica.sh 'RSYNC' '10.1.2.43' '/u01/install/MyKeyWithoutPassPhrase.ppk
```

**NOTE:** You can hardcode these values in the script, there is a section in the script for that. In that case, the sys password value must be set encrypted. To obtain an encrypted value of the sys password, run the following in the primary WLS Administration host (with oracle user):
```
cd $DOMAIN_HOME/bin
./setDomainEnv.sh
java -Dweblogic.RootDirectory=<domain_home> weblogic.security.Encrypt
```
Provide the sys password in clear, and it will return the encrypted value. Use the encrypted value when setting the password in the script.

4. **Once it completes, execute the script in the secondary middle tier WebLogic Administration host** (with oracle user). Provide the input parameters according to the method used and the site. Monitor the execution and watch for any errors. The script will verify the DG status. As it is the standby, it will copy the domain configuration from the secondary staging filesystem to the secondary WebLogic domain and will make the required replacements in the configuration that are required in the standby (db connect string in datasources, etc).

For the changes to take effect in the secondary location, the WebLogic Administration server needs to be restarted (in case it was up). Note that this needs to have the secondary DB in snapshot standby mode or use Active Data
Guard. See details in Open Secondary Site for validation to verify standby without performing a complete switchover. If the secondary WebLogic Administration server was stopped when the changes where replicated, the changes will take effect next time it is started (during the switchover, failover or open secondary site for validation operations).

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 residing 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 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 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 so that they get executed regularly. 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 performing 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 the config_replica.sh may take. Otherwise, copy jobs may overlap.

**Switchover**

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. This is known as a manual switchover (alternatively, Oracle Site Guard can be used to automate the tasks required in a switchover and reduce the RTO in such operation, refer to Appendix B for details and benefits on this).

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

<table>
<thead>
<tr>
<th>SWITTOVER STEP</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Propagate any pending configuration changes</td>
</tr>
<tr>
<td>2</td>
<td>Stop servers in primary Site</td>
</tr>
<tr>
<td>3</td>
<td>Switchover DNS name</td>
</tr>
<tr>
<td>4</td>
<td>Switchover Database</td>
</tr>
</tbody>
</table>
5 Start the servers in secondary site (new primary)

Restart secondary Admin Server if configuration changes were replicated while this was standby, so they take effect.
Start secondary managed servers (using the WebLogic Console or scripts)

Figure 12 WebLogic for OCI Disaster Recovery system AFTER a switchover (in DBFS based method)
Failover

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 perform the failover manually, or, alternatively, use Oracle Site Guard to automate the tasks required in a failover and reduce the RTO in such operation (refer to Appendix B – Using Enterprise Manager Site Guard to manage Disaster Recovery for details and benefits on this).

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

<table>
<thead>
<tr>
<th>FAILOVER STEP</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Switchover DNS name</td>
<td>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 frontend address of the system to the public IP used by LBR in site2</td>
</tr>
<tr>
<td>2 Failover Database</td>
<td>Use DB broker in secondary db host to perform the failover. As user oracle: <code>[oracle@drdbwlmpxb -]# dgmgrl sys/your_sys_password@secondary_db_unqname DGMGRL&gt; failover to &quot;secondary_db_unqname&quot;</code></td>
</tr>
<tr>
<td>3 Start the servers in secondary site</td>
<td>Restart secondary admin server if configuration changes were replicated while this was the standby, so they take effect.</td>
</tr>
</tbody>
</table>
Start secondary managed servers (use the WebLogic Console or scripts)

See Appendix D – Other lifecycle operations for additional lifecycle operations.

BEST PRACTICES

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

- Use JDBC persistent stores. In case you create custom persistent stores, be sure that you create them as JDBC persistent stores. This way, the JMS messages will be stored in database tables, so this information will replicated to secondary site via Data Guard.
- Maintain the same patch level in primary and standby sites. The software is not replicated automatically to the secondary system in any tier. If you install a patch in primary, you have to install the same patch in the standby location. When patching 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 using config_replica.sh script) 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 system. You can alternatively open the secondary site for validation without performing a complete switchover, as explained in “Appendix D Open Secondary Site for validation”.
- 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 config replica script.

See Appendix E – Additional Best Practices for details on additional best practices.
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 separate middle tier configurations, each pointing to their local database, are created to minimize the file synchronization needs across data centers. With this Disaster Recovery solution, Oracle Cloud eliminates the costs and complexity of owning and managing a standby hardware, software, and remote data center – while achieving the best 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, 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.
APPENDIX A – CONSIDERATIONS FOR RAC DB SYSTEMS

When the DB system used by the WebLogic Server is an Oracle Real Application Cluster (RAC) DB System, 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 DB System is used. When RAC DB System 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 DB System

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 DB System, which is the best practice.

During the DR setup for WebLogic on OCI, it is required that each midtier can connect to the remote RAC DB System. This communication is also 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 that are created in each midtier to point to the remote RAC DB System will use the IP address rather than the scan address name.

Data Guard Configuration for RAC DB System

The RAC Data Guard can be configured using OCI Console as explained in Option 1) Configuring the Data Guard using OCI Console and that is the recommended approach.

In case that the Data Guard can’t be configured automatically using the OCI Console for your RAC database, you can alternatively use the scripts from RAC_DG_setup_scripts.zip (note these are different from the scripts provided to manually configure DG in a single instance db) in order to configure Data Guard for a RAC DB System. Download it and follow the instructions included in the README.txt. These are scripts automate the procedure explained in Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC)(Doc ID 1617946.1) for configuring Data Guard between two OCI RAC DB Systems.

Create and use a CRS database service

Oracle WebLogic Server Suite on OCI provisions GridLink data sources in your Oracle WebLogic Server domain to connect to the selected Oracle Database cluster. GridLink provides dynamic load balancing and failover across the nodes in an Oracle Database cluster, and also receives notifications from the cluster when nodes are added or removed. 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.
NOTE: it is recommended to implement these steps in primary and secondary domain BEFORE configuring WLS for Disaster Recovery, so the replacements of the db connection string urls that are performed by setup scripts use this service name instead of the default PDB service.

e) **Create an Oracle Database service in primary RAC.**

Connect to a db node of the primary RAC and run the following with the user oracle to add, configure and start a service:

```bash
svrctl add service -db <PRIM_DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -preferred <INSTANCE_NAME1>,<INSTANCE_NAME2> -pdb <PDB_NAME> -role "PRIMARY,SNAPSHOT_STANDBY, PHYSICAL_STANDBY"
svrctl modify service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb <PDB_NAME>
svrctl start service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME>
```

Note that it is important to provide the 3 roles to the "-role" parameter when creating the service, so the service is automatically started when the database is in primary, snapshot or physical standby role.

**Example:**

```
[oracle@priracnode1 ~]$ srvctl add service -db ORCL_lhr3jg -service mydbservice.mycompany.com -preferred ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY, PHYSICAL_STANDBY"
[oracle@priracnode1 ~]$ srvctl modify service -db ORCL_lhr3jg -service mydbservice.mycompany.com -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb pdb1
[oracle@priracnode1 ~]$ srvctl start service -db ORCL_lhr3jg -service mydbservice.mycompany.com
```

**Service name:** mydbservice.mycompany.com

Server pool:
Cardinality: 2
**Service role:** PRIMARY,SNAPSHOT_STANDBY, PHYSICAL_STANDBY
Management policy: AUTOMATIC

...  
Available instances:
CSS critical: no

---

f) **Create the same Oracle Database service in the secondary RAC database.**

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

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

**Example:**

```
[oracle@secracnode1 ~]$ srvctl add service -db ORCL_fra3vb -service mydbservice.mycompany.com -preferred ORCL1,ORCL2 -pdb pbdb1 -role "PRIMARY,SNAPSHOT_STANDBY"
[oracle@secracnode1 ~]$ srvctl modify service -db ORCL_fra3vb -service mydbservice.mycompany.com -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb pbdb1
[oracle@secracnode1 ~]$ srvctl start service -db ORCL_fra3vb -service mydbservice.mycompany.com
```

**Service name:** mydbservice.mycompany.com

Server pool:
Cardinality: 2
**Service role:** PRIMARY,SNAPSHOT_STANDBY, PHYSICAL_STANDBY
Management policy: AUTOMATIC
DTP transaction: false
...
g) **Update the connection string in datasources and boot files**

Datasources and jps boot files must point to this new service name instead of the PDB default service name:

- To update the datasources, you can connect to WLS Console:
  
  Domain structure > Services > Datasources
  
  Click on a datasource > Configuration > Connection Pool > URL
  
  Within the JDBC URL, replace the value of `SERVICE_NAME` with your new database service name.
  
  An example JDBC URL is shown below:
  
  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=db-scan-address)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=mydbservice.mycompany.com)))
  ```
  
  Save and apply changes

- Update the boot database URLs in the WebLogic Server domain files.
  
  You need to update the service name also in the jps config files.
  
  Navigate to the directory `config/fmwconfig` within your domain configuration.
  
  Edit the file `jps-config.xml`.
  
  Locate this line in the file.
  
  ```
  <property name="jdbc.url" value="JDBC_URL"/>
  ```
  
  Replace the value with the updated JDBC URL. Then save your changes.
  
  Edit the file `jps-config-jse.xml`.
  
  Locate this line in the file.
  
  ```
  <property name="jdbc.url" value="JDBC_URL"/>
  ```
  
  Replace the value with the updated JDBC URL. Then save your changes.

- **A restart of the complete domain is required** (including admin server) for these changes to take effect.

It is recommended to do the same on the secondary domain BEFORE running disaster recovery setup.

---

**NOTE:** alternatively, a simple way to change this is by running the `updateDBServiceName.sh` script (you can find in the WLS for OCI DR utils zip) in the admin server of each domain. It changes the service name in the connection string in the datasources and jps files:

```
./updateDBServiceName.sh <previous_pdb_service> <new_pdb_service>
```

Example in primary domain:

```
./updateDBServiceName.sh PDB1.subnetlon3.myvcnvcnen.oraclevcn.com mydbservice.mycompany.com
```

Example in secondary domain:

```
./updateDBServiceName.sh PDB1.subnetfra3.myvcntra.oraclevcnen.com mydbservice.mycompany.com
```

---

**Running the dbfs_dr_setup_root.sh in RAC scenarios**

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. In that way, the `tnsnames.ora` alias that DBFS uses to connect to the database will use the scan address.

You can also provide the previously created db service name so dbfs will connect to the created CRS service rather than the default PDB service, and this is the recommended approach.

Example of execution in a RAC scenario:

```
./dbfs_dr_setup_root.sh ddrbracca2a.scan.subnetlon1.myvcnlon.oraclevcn.com 1521 mydbservice.mycompany.com acme1234 /u01/install/V982064-01.zip
```

---

**Running the DR setup scripts in RAC scenarios**

The `fmw_dr_setup_standby.sh` script requires that you provide some information about the primary database as input parameters. When running this script in a RAC scenario, you need to **provide one of the scan IPs of the primary**. The script will later gather the complete connection string from the database and use it for the subsequent connections.

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. If you provided the previously created database service name in the dbfs dr setup script, you must provide it also when running the dr setup script for consistency:
Example of execution in a RAC scenario:

```
./fmw_dr_setup_standby.sh '10.0.2.34' '1521' 'mydbservice.mycompany.com' 'acme1234' DBFS
```

Where '10.0.2.34' is one of the scan IPs of the primary RAC and 'mydbservice.mycompany.com' is the CRS service created in the RAC database.

**Apply other best practices**

Regardless of 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, 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”**
  Verify that the “Test Connections On Reserve” is checked in the datasources. Despite the GridLink datasources received FAN events when a RAC instances becomes unavailable, it is a best practice to enable the test connection on reserver 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 during 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.
APPENDIX B – USING ENTERPRISE MANAGER SITE GUARD TO MANAGE DISASTER RECOVERY

Oracle Site Guard is a disaster-recovery solution that enables administrators to automate complete site switchover or failover. It orchestrates the coordinated failover of Oracle Fusion Middleware, Oracle Fusion Applications, and Oracle Databases. Oracle Site Guards offers the following benefits:

- Fully automate disaster recovery operations and launch them with a single click
- Minimizes disaster-recovery time
- Reduces human errors
- Flexible and customizable
- Eliminates the need for special skills
- Use a single pane of glass to manage disaster recovery
- Assure disaster recovery readiness using on-demand or scheduled disaster recovery drills

Oracle Site Guard can be used to coordinate the WLS Disaster Recovery switchover as described in this paper for methods DBFS and FSS with RSYNC. Follow the steps in the document “Using Oracle Site Guard to manage Disaster Recovery for OCI PaaS Systems” to configure Site Guard for the WebLogic Disaster Recovery environment and use it to perform the switchovers and failovers.
APPENDIX C – DB SYSTEM BACKUPS ON MANUAL DATA GUARD

Backing up the DB System is a key aspect of any Oracle database environment. Oracle Cloud offers various approaches: you can store backups in local or cloud storage; the backup can be automatic, custom using 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](#)) the backup needs to be configured manually in order 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 just 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 a 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 archived older than 20 days
  delete noprompt archivelog all completed before 'SYSDATE-20';
  exit;
  RMAN
  echo "deleted applied old archivelogs on $ORACLE_SID"
  ```
When the Data Guard was configured using Cloud Console UI, you can enable automatic backups in the primary database and this is a good approach. 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 resuming 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 an infrequent scenario, the standby database will have to be recreated as well.

In a manual Data Guard, you can re-run the scripts that are provided in the step Option 2) Configuring the 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 the DR with this new standby system. Follow the steps described in Reassemble the WLS for OCI DR after recreating the standby DB System for more details on this.
APPENDIX D – ADDITIONAL LIFECYCLE OPERATIONS

Open Secondary Site for validation

It is possible to validate the standby site without performing a complete switchover, by converting the standby database to snapshot standby. This allows the secondary WLS servers to be started 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:

<table>
<thead>
<tr>
<th>STEPS TO OPEN THE STANDBY SITE FOR VALIDATIONS</th>
<th>DETAILS</th>
</tr>
</thead>
</table>
| 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@drdB -]$ dgmgrl  
sys/your_sys_password@primary_db_unqname  
DGMGRRL> 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','acme1234#','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) |
| 3 Validate | As this is not a switchover and the primary is still with primary role, the front-end name will be pointing to primary so any access will be redirected to primary.  
In order 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 by the secondary front-end IP.  
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 resolve the front-end name with the primary IP regardless which name is in the /etc/hosts of the client. |

Once the secondary site has been validated, follow these steps to revert it back to standby role again:

<table>
<thead>
<tr>
<th>STEPS TO REVERT BACK STANDBY TO STANDBY ROLE</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Stop managed servers and admin servers in secondary</td>
<td>You can connect to secondary WebLogic Console and shutdown managed servers and Admin server in secondary site.</td>
</tr>
</tbody>
</table>
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 back any updated /etc/hosts in clients | If you updated the virtual front-end name in the /etc/hosts file of a client, in order to point to secondary site, revert it back so the virtual front-end name points to primary front-end IP again.

---

### How to recreate the dbfs wallet

**NOTE:** This applies to DBFS based method only.

During the DR setup, a DBFS mount is created and mounted in the midtier servers when 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 in order to connect to the database and mount the dbfs. In case that the password of the "dbfsuser" is updated in the database, the wallet needs to recreated with the new password. You can follow these steps to recreate the wallet:

- 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
  ...
  ```

- 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:

  ```
  $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 acme1234#
  ```

  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
  ```
You need to 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).

**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 reinstatiation 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 it has the following implications on the disaster recovery:

- **Impact in RPO**: the domain configuration changes that are replicated from the primary site will not be pushed to the secondary domain configuration if the secondary admin server host is stopped. In case of a failover, the secondary domain can be out-of-sync from the primary configuration. To avoid this, you need to keep the secondary admin server host up at least, and stop only the other managed server compute instances.

- **Impact in RTO**: the RTO is increased if the secondary midtier hosts are stopped and you need start them and synchronize the secondary domain with primary changes before a switchover or failover.

To minimize these implications, in case you want to have some secondary WLS compute instances stopped, you can keep the secondary admin server host up and stop only the 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.*

**Scale-out and Scale-in procedures in a DR environment**

It is possible to scale-out and scale-in a WLS for OCI system following the steps described in the WLS for OCI documentation [Scale a Domain](#).

When you perform an scale-out or scale-in a WLS for OCI DR environment, there are some characteristics specific to a DR environment that must be taken into account: there are 2 WLS for OCI systems (primary and secondary), the domain configuration in secondary is a copy of the primary configuration (using primary hostnames as listen-addresses than primary), aliases are required in the midtier’s /etc/hosts, etc. The new nodes that are added in a scale-out operation do not include the aliases in its /etc/hosts file by default, and 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, follow these steps.

### Scale-out

The recommended procedure to SCALE-OUT a WLS for OCI DR environment is the following:

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

1. 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-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 the some applications may fail in secondary. See “About having different number of nodes in primary and standby”
2. Follow the steps described in Add or Remove WebLogic Server Nodes in primary stack by increasing the number of the nodes (in this example it is assumed that the number is increased by 1 node)
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, pointing to primary front-end LBR IP
   111.111.111.111 mywebapp.mycompany.com
   ```
   b. 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_midtier2_fqdn> <sec_midtier2_shortname>
   10.0.0.81 <prim_midtier2_fqdn> <prim_midtier2_shortname> <sec_midtier1_fqdn> <sec_midtier1_shortname>
   ```
   c. 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. 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, previously to proceed with the scale-out, set the listen-addresses in the secondary domain and set there the secondary wls hostnames. This makes that the scale-out procedure run without issues. Later, once the scale-out is finished, the aliases will be added too as in option 1. Detailed steps explained here:

1. Convert the standby database into snapshot standby.
2. Change the listen address of the weblogic servers in the secondary domain to use the secondary names instead the primary 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-o.wlsdrcvnlon1ad2.wlsdrcvnlon1.oraclevcn.com
     wlsociprefix-wls-1.wlsdrcvnlon1ad2.wlsdrcvnlon1.oraclevcn.com
     ```
     Primary midtier1 fqdn is wlsociprefix-wls-o.wlsdrcvnlon1ad2.wlsdrcvnlon1.oraclevcn.com, and its hostname is wlsociprefix-wls-o.
     Primary midtier2 fqdn is wlsociprefix-wls-1.wlsdrcvnlon1ad2.wlsdrcvnlon1.oraclevcn.com and its hostname is wlsociprefix-wls-1.

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

     ```
     wlsociprefix-wls-o.wlsdrcvnfra1ad2.wlsdrcvnfra1.oraclevcn.com
     wlsociprefix-wls-1.wlsdrcvnfra1ad2.wlsdrcvnfra1.oraclevcn.com
     ```
     NOTE: hostnames are expected to be the same in primary and secondary wls hosts, only the fqdn values will differ.

   - In secondary admin server node, replace primary hosts FQDN 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-o.wlsdrcvnfra1ad2.wlsdrcvnfra1.oraclevcn.com /wlsociprefix-wls-o.wlsdrcvnfra1ad2.wlsdrcvnfra1.oraclevcn.com /g' config.xml
     sed -i 's/wlsociprefix-wls-1.wlsdrcvnfra1ad2.wlsdrcvnfra1.oraclevcn.com /wlsociprefix-wls-1.wlsdrcvnfra1ad2.wlsdrcvnfra1.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 name and secondary front-end IP, 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
   ```
   - 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 section 10b of this document (Configure the primary Enterprise application and App Gateway in the secondary WebLogic domain) 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. (Optional) Now you can revert back the change done in step 2 and set again the primary FQDN names in the listen addresses by doing the same replacement in the other way:

   ```
   cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_post_scale-out
   sed -i 's/secondary_midtier1_fqdn_name/primary_midtier1_fqdn_name/g' config.xml
   sed -i 's/secondary_midtier2_fqdn_name/primary_midtier2_fqdn_name/g' config.xml
   ```
   Alternative, this will be done later when you replicate the conf from primary using config_replica.sh (step d)

   c) **Once both primary and standby are scaled-out, it is very important** to complete configuration by adding the aliases for the new node to all the midtier hosts (existing and new nodes):
   1. In primary, add it to all the existing primary midtier nodes (and also in the new one). Example:

      ```
      <primary_newnode_IP> <primary_newnode_fqdn> <primary_newnode_hostname> <secondary_newnode_fqdn>
      ```
   2. In secondary, add it to all the existing midtier nodes (and also in the new one). Example:

      ```
      <secondary_newnode_IP> <secondary_newnode_fqdn> <secondary_newnode_hostname> <primary_newnode_fqdn>
      ```

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 is the following:

a) **Scale-in primary** WLS for OCI system:
   1. Follow the steps described in Add or Remove WebLogic Server Nodes to reduce the number of servers in the primary stack.

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.
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.

**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, proceeding 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:

```
<Error> <Deployer> <BEA-149251> <Unable to set the activation state to true for the application "myCustomApp".
```

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 (or the IP of the secondary db could be used) and 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 “unknownhostexception” 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.

As a summary, **having different number of configured servers in primary and secondary can cause inconsistencies hence it is not recommended.**
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 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 to provide 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 for OCI DR with a new standby DB system:

a) Before terminating the standby DB System, note down the DB unique name ($ORACLE_UNQNAME variable in the standby DB host), 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.

e) In the standby WLS hosts:
   • Edit the file /u01/data/domains/local_CDB_jdbcurl.nodelete and update the standby DB unname with the new standby DB unique name.

f) (Only 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.

g) (Only 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 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.

h) Also, 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!

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

<table>
<thead>
<tr>
<th>DB unique name ($ORACLE_UNQNAME)</th>
<th>ORIGINAL STANDBY DB SYSTEM</th>
<th>NEW STANDBY DB SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCL_phx1kg</td>
<td>ORCL_phx1c3</td>
<td></td>
</tr>
</tbody>
</table>
Hence, in the **standby WLS hosts:**

<table>
<thead>
<tr>
<th>FILE TO UPDATE</th>
<th>ORIGINAL CONTENT</th>
<th>NEW CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/u01/data/domains/local_CDB_jdbcurl.nodelet</code></td>
<td><code>drdb6b-scan.my-subnet.region2vcn.oraclevcn.com:1521/ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com</code></td>
<td><code>drdb6b-scan.my-subnet.region2vcn.oraclevcn.com:1521/ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com</code></td>
</tr>
<tr>
<td>(only if DBFS based method is used)</td>
<td><code>ORCL_phx1kg</code></td>
<td><code>ORCL_phx1kg</code></td>
</tr>
<tr>
<td><code>$DOMAIN_HOME/dbfs/localdb.log</code></td>
<td>(other entries)</td>
<td>(other entries)</td>
</tr>
<tr>
<td>(only if DBFS based method is used)</td>
<td><code>ORCL_phx1kg = (DESCRIPTION=(SDU=65536)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=drdb6b-scan.my-subnet.region2vcn.oraclevcn.com)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com)(UR=A)))</code></td>
<td><code>ORCL_phx1kg = (DESCRIPTION=(SDU=65536)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=drdb6b-scan.my-subnet.region2vcn.oraclevcn.com)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com)(UR=A)))</code></td>
</tr>
<tr>
<td><code>$DOMAIN_HOME/dbfs/tnsnames.ora</code></td>
<td>(other entries)</td>
<td>(other entries)</td>
</tr>
</tbody>
</table>

And in the **primary WLS hosts:**

<table>
<thead>
<tr>
<th>FILE TO UPDATE</th>
<th>ORIGINAL CONTENT</th>
<th>NEW CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(only if DBFS based method is used)</td>
<td><code>ORCL_phx1kg = (DESCRIPTION=(SDU=65536)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=10.2.0.2)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com)(UR=A)))</code></td>
<td><code>ORCL_phx1kg = (DESCRIPTION=(SDU=65536)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROTOCOL=TCP)(HOST=10.2.0.2)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL_phx1kg.my-subnet.region2vcn.oraclevcn.com)(UR=A)))</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB System private IP</th>
<th>10.2.0.2</th>
<th>10.2.0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB System hostname</td>
<td>drdb6b.my-subnet.region2vcn.oraclevcn.com</td>
<td>&lt;same value&gt;</td>
</tr>
<tr>
<td>DB System scan name</td>
<td>drdb6b-scan.my-subnet.region2vcn.oraclevcn.com</td>
<td>&lt;same value&gt;</td>
</tr>
</tbody>
</table>
APPENDIX E – ADDITIONAL BEST PRACTICES

Use JDBC stores for TLOGs and JMS stores

In a DR topology, it is strongly recommended to use JDBC stores for storing the server’s TLOGs and JMS stores. The JDBC stores leverage the consistency, data protection of an Oracle database and makes resources available for all the servers in the cluster. Using JDBC storages has advantages especially for DR topologies: the information resides in the DB, is replicated to secondary via Data Guard, so this information is available in the secondary site after a switchover and coordinated with the rest of the information that is stored in the database.

Regarding the TLOGs: by default in WebLogic on OCI, the server’s TLOGs are configured to use the default store, which is the file system. In order to change the TLOG store of the managed servers to a JDBC store:

- Login in the WebLogic Console to primary domain, and click in Lock & Edit
- Navigate to the WLSSchemaDataSource: Services > Data Sources > WLSSchemaDataSource
- Click on “Targets” and in case it is only targeted to the Admin Server, target it to the cluster also. This data source will be used for the TLOGs of the managed servers.
- Navigate to the Managed Server: Environment > Servers > <manager_server_name> > Configuration > Services
- In Advanced > Transaction Log Store
  o In Type, select JDBC
  o In DataSource, select WLSSchemaDataSource
  o In the prefix name, you can use the default value

Example:

- Repeat the same in the rest of the managed servers of the cluster (this change is not required for Admin Server, it can use the default tlog)
- Apply Changes
- A restart of the managed servers is needed for the changes to be applied.

Now you can replicate the configuration change to secondary domain using the config_replica.sh script.

Regarding the JMS, there are no JMS servers or persistent stores defined by default in Oracle WebLogic on OCI. In case your application uses JMS stores, be sure that you configure them using JDBC persistent stores.
APPENDIX F – DISASTER RECOVERY BASED ON BLOCK VOLUME CROSS-REGION REPLIATION

Topology Description

The DR solution based on Block Volume Cross-Region replication is, in most of the aspects, the same than described previously in this paper. The topology is the same and only the way in which configuration is replicated varies. The following are the key aspects of the topology.

It is an active-passive model. There is a primary system consisting on an Oracle WLS for OCI domain, load balancer, and Oracle Cloud Infrastructure DB system in one region, and a standby system, consisting in Oracle WLS for OCI domain, load balancer, and Oracle Cloud Infrastructure DB system in a different region. Different regions must be used for primary and standby to take advantage of the Block Volume Cross-Region replication.

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

The secondary Oracle WebLogic Server domain is a replica of the primary domain, using the same name, schemas, passwords, etc. but pointing to the secondary database. The listener addresses of the WebLogic Servers are configured with the primary midtier host names, so in secondary midtier hosts the pertinent aliases are created in the hosts file to resolve them with the secondary IPs.

On the front-end, there is a unique name configured to access the applications running in the system. This “virtual” front-end name will point to the IP of the OCI Load Balancer of the primary site. In case of a switchover, this front-end name is updated to point to the IP of the OCI Load Balancer of the secondary site. It always must point to the LBR IP of the site that has the primary role in each time.

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. For some actions, during the DR setup and lifecycle steps described in this document, the standby database will be converted from a physical standby to a snapshot standby. A database in snapshot standby mode is fully updateable database. A snapshot standby database 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.

Like in the other approaches, all the information that resides in the database (custom schemas, TLOGs, JDBC persistent stores, OPSS information, etc.) is automatically replicated to the secondary site by Data Guard. The WebLogic Domain configuration, located on the local filesystem in each site, needs to be replicated from the primary site to the secondary as well. The difference versus the other methods described in this paper, is how the WebLogic Domain configuration is
replicated from primary to standby. In this case, it will be transferred using the **Block Volume Cross-region replication** feature. This capability allows you to perform ongoing automatic asynchronous replication of block volumes and boot volumes to other regions. See **Cross-Region Volume Replication** in Oracle documentation for more information. This can be used for Disaster Recovery scenarios like the topology described here.

![Diagram of Oracle Cloud Infrastructure Disaster Recovery topology based on Block Volume cross-region replication.](image)

**Figure 15** Detailed WLS for OCI Disaster Recovery topology based in Block Volume cross-region replication

### Advantages and Disadvantages of BV Replication DR model

The advantages of a Block Volume replication approach as compared to the other methods described in this paper can be summarized as follows:

- **It is a general-purpose solution:**
  The DR solution based in a block volumes replication is more agnostic to the specific system particularities. 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 uses a continuous and dissatended replica process:**
  The replication of the block volumes is a dissatended and continuous process performed by OCI infrastructure. It is not based on a script that needs to be manually run or scheduled.

- **The replication is not limited to the domain configuration:**
  The information of the replicated block volumes is an exact copy from the primary block volumes. Not only the
admin host WLS domain folder configuration is replicated, all the info residing in the data block volume, and in the data block volumes of the rest of the nodes will be replicated. This means that other custom files that are located outside the domain (as long as it is in a block volume that is replicated) can be also automatically replicated without need additional intervention.

In WLS for OCI compute instances, the Oracle software resides in a separate block volume (the mw block volume). It is not required to replicate the mw block volume on an ongoing basis. The content of this volume does not change frequently. Including it in the same replication strategy as the configuration volume would unnecessarily increase the cost, the RTO, and the complexity of the solution. It is possible, however, to replicate this volume sparsely to propagate patches from primary to secondary.

There are, however, some disadvantages as compared to the other methods described in this paper that need to be considered:

- **Higher Complexity:**
  Block Volume Cross-region Replication management may be complex when the number of block volumes replicated is high. It requires a good lifecycle management of the block volume and replica. Switchover and failover operations are also more complex than in the other methods, and there are additional pre and post switchover/failover steps.

- **Additional Cost implications:**
  After 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.
  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](#).

- **The amount of data replicated across regions is much higher:**
  In the other methods described in this paper (DBFS and FSS with RSYNC), the WebLogic Domain configuration that is replicated is minimal, and only from the admin server domain. In this case, the complete BV where the WebLogic Domain resides is replicated from primary and secondary, and for all the nodes. This may be considered an advantage when you require to replicate additional artifacts or files. It is a trade-off between the customer needs and the cost.

- **BV replication based DR's RTO is worse:**
  When using Block Volume replication for configuration replication the switchover and failover require additional steps (activate replicas, attach block volumes, etc.) that increment the downtime during a switchover or a failover. For a normal switchover of a 2 nodes cluster, the time is increased in about 15 minutes, and there are post-switchover tasks (detach block volumes, etc.) that take about another 15 minutes more (although, this last set of steps can be applied without affecting the recovery time objective).

- **Worse control on RPO** (referring to WebLogic Configuration, because the RPO for the database is exactly the same than in other 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 amount of write I/O operations to the volume. In the others methods described in this paper, as the information that is replicated using a script and the amount of information replicated is less, the user can have a finer control on the RPO for the WebLogic Configuration.

**Assumptions for BV Replication DR model**

**WebLogic Editions**

As described in Assumptions > WebLogic Editions for DBFS and FSS with RSYNC models.

**Authentication**

As described in Assumptions > Authentication for DBFS and FSS with RSYNC models.
Load Balancer
As described in Assumptions > Load Balancer for DBFS and FSS with RSYNC models.

Database
As described in Assumptions > Database for DBFS and FSS with RSYNC models.

Block Volumes Replicated
The BV Replication DR solution assumes that only Block Volumes will be replicated to the other site, and that the Boot Volumes are not replicated. Each mid-tier host of a WLS for OCI domain has two Block Volumes attached:

- \(<wlsociprefix>-data-block-N\), mounted in /u01/data. This block volume contains the WebLogic domain configuration in each compute instance. This block volume needs to be replicated to the standby site on an ongoing basis.
- \(<wlsociprefix>-mw-block-N\), mounted in /u01/app. This block volume contains the Oracle software homes and the inventory in each compute instance. Although these block volumes can be replicated also, it is not needed to replicate them on an ongoing basis. Their content does not change unless the WebLogic software or the JDK is patched/ altered. These are the different approaches you can use to manage these volumes:
  1. You can include these 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, higher cost, and worse RTO in a switchover operation than option #3.
  2. An optimized approach is to replicate these volumes sparsely. For example, when the primary peer has been patched. This will incur lower cost and better RTO when performing a switchover. In this model, patching is performed only in one site and copied to the secondary. However, this will cause higher management overhead, higher cost, and worse RTO in a switchover operation than option #3.
  3. As an alternative approach, this “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 a better RTO 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).

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.

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.

Requirements for BV Replication DR model

Front-end address
Same as described in Requirements > Front-end address for DBFS and FSS with RSYNC DR models.

WebLogic Resource Name Prefix
Same as described in Requirements > WebLogic Resource Name Prefix for DBFS and FSS with RSYNC models.

Network communication between sites
As described in Requirements > Network communication between sites, the primary and standby databases need to communicate with each other over their listener port for redo transport. The communication between primary and secondary sites can use Oracle Cloud’s internal networks, by using Dynamic Routing Gateway, which is the recommended approach (refer to the Dynamic Routing Gateway documentation for additional details on the network configuration). The communication between sites can also happen over an Internet Gateway (Oracle Net’s traffic is encrypted), but this is not recommended for security and reliability reasons.

In the Block Volume Replication Cross-Replication DR model, no additional network communication is needed: cross-region rsync is not used, and connectivity from mid-tiers to remote database is not needed.
Regions with Block Volume Cross-Region Replication

The primary and standby regions used in the DR topology must be different, and the Block Volume Cross-Region Replication must be available between the primary and secondary regions that are used in the DR topology. Not all the regions are interconnected for BV 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 Volume Replication > Replication Target Regions](#).
Setup Process for BV Replication DR model

The setup procedure for the BV Replication DR model is a variation of the setup process described previously in this document for DBFS and FSS with RSYNC methods, as described in the following flow chart:

Figure 16 Flow chart of the DR setup steps for the models described in this paper
As in the other models, it is assumed that, as starting point, the primary site, consisting of an OCI DB system, a WLS for OCI domain, and the Load Balancer (OCI LBR) is live. The secondary DR configuration, residing in a geographically remote site, will be created for 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).

This is a summary of the steps for the setup process based on BV replica:

1. Choose a virtual front-end name
2. Prepare Primary mid-tier for the virtual front-end
3. Setup the Database in Secondary Site
4. Provision WLS for OCI in Secondary Site
5. Prepare Secondary mid-tier for the virtual front-end
6. Create the mid-tier hosts aliases
7. Convert the standby DB into physical standby
8. Configure the Block Volume Cross-Region replication
9. Prepare the script for the environment specific replacement
10. Complete DR configuration when IDCS authentication is used
11. Verify the DR Setup

From step 1 to step 6, both included, the steps are exactly the same than explained before in Disaster Recovery setup. The rest of the steps are specific for the BV replication DR model and explained in the following points.

1. Choose a virtual front-end name
   Same than in Disaster Recovery setup > 1. Choose a virtual front-end name
2. Prepare Primary mid-tier for the virtual front-end
   Same than in Disaster Recovery setup > 2. Prepare Primary mid-tier for the virtual front-end
3. Setup the Database in Secondary Site
   Same than in Disaster Recovery setup > 3. Setup the Database in Secondary Site
4. Provision WLS for OCI in Secondary Site
   Same than in Disaster Recovery setup > 4. Provision WLS for OCI in Secondary Site
5. Prepare Secondary mid-tier for the virtual front-end
   Same than in Disaster Recovery setup > 5. Prepare Secondary mid-tier for the virtual front-end
6. Create the mid-tier hosts aliases
   Same than in Disaster Recovery setup > 6. Create the mid-tier hosts aliases
7. 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 in to 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.
   b) Convert the standby database into physical standby again
      Execute this 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
      ```
8. Configure the Block Volume Cross-Region replication

In order to replicate the block volumes of the mid-tier hosts from primary region to secondary region, you must 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, it is not required to replicate them on an ongoing basis, so the steps to replicate these block volumes are included here as optional.

Note down the names, the OCIDs and the AD where they are located. For example:
- `wlsociprefix-data-block-0`, with OCID ocid1.volume.oc1.eu-london-1.abtheljs...uhmlg2s3avjradoeoirz5q, in AD-1
- `wlsociprefix-data-block-1`, with OCID ocid1.volume.oc1.eu-london-1.abthelj....cdcayp6gw2irsri4nrgrx7q, in AD-2
- Optional:
- `wlsociprefix-mw-block-0`, with OCID ocid1.volume.oc1.eu-london-1.abtheljs...jihbuhmg2s3deoirkkz4s, in AD-1
- `wlsociprefix-mw-block-1`, with OCID ocid1.volume.oc1.eu-london-1.abthelj....wiphp6gw2irsri4nrh7jdy, in AD-2

These Block Volumes are mounted as in this example:

```
[opc@wlsociprefix-wls-0 ~]$ df -h
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  162M  560G   3% /run
tmpfs       7.2G   0  7.2G   0% /sys/fs/cgroup
/dev/sda1   39G   2G  37G   0% /u01/data
/dev/sdb    49G  14G  35G  26% /u01/app
/dev/sdc    49G  176M  47G   4% /u01/app
/dev/sda3   39G  3.7G  35G  10% /u01/data
/dev/sdb1   200M  8.6M  192M   5% /boot/efi
/dev/sdc1   49G  1.4G  46G   3% /u01/app
/dev/sdc1   49G  176M  47G   4% /u01/app
```

b) Identify the Block Volumes of the secondary mid-tier hosts

Repeat the steps described in a) to get the names, OCID and Availability domains of the block volumes of the secondary mid-tier hosts.

c) Enable the Cross-Region replication in the Primary Block Volumes

For each Block Volume of the primary WLS for OCI, enable the cross-region replica:
- Edit the Block Volume, set the Cross Region Replicaion to ON
- **Region:** this is the destination for the replica. Select the secondary region.
- **Availability Domain:** select the AD where the equivalent Block Volume is located in secondary region.
  Example: if `wlsociprefix-data-block-0` in secondary region is located in AD-3, select AD-3 as the location for the replica of the primary `wlsociprefix-data-block-0`.
- **Name:** to facilitate future automation using OCI CLI, Oracle recommends using the same name for the block volume replica object regardless in which region the replica is. Use for example: "<wlsociprefix>-data-block-N_replicated_from_remote_site"
  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 > Block Volume Replicas.

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 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 unmounted, detach the blocks volumes from the OCI Console
  Go to each Block Volume > Attached Instances > Detach from Instance
  The OCI Console will ask you to run some iscsi commands before completing the detachment.
- With root user, edit the /etc/fstab file and remove (or comment) the entries 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-ddf2b3cb0211 /u01/app ext4 auto,defaults,_netdev,nofail 0 2
  #Remove this entry:
  #UUID=9e87cf72-a75c-4dff-9825-432f16688f9 /u01/data ext4 auto,defaults,_netdev,nofail 0 2

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

  Optionally, 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.

  9. 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, some replacements need to be performed on the WebLogic Domain configuration. This is because the WebLogic Domain configuration is a copy from primary, so the DB connection url used in the datasources and other files points to primary database. It needs to be replaced with the secondary database connection url.
     In order to automate these replacements, you can use the provided script replacement_script.sh.
     a) Download the script from here and upload it to all the mid-tier hosts (primary and secondary).
     b) Store it in a folder that is NOT in the Block Volume that is replicated. You can store it, for example, in a folder under the oracle user's home (for example, /home/oracle/scripts).
     c) Change the ownership of the file to oracle user (this script will be executed by oracle user).
     d) Edit the script and customize it in each mid-tier host with the appropriate 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.

  10. Complete DR configuration when IDCS authentication is used
     Same than in Disaster Recovery setup > 10. Complete DR configuration when IDCS authentication is used.
     Note, however, that you do not need to modify the cloudgate configuration of the secondary mid-tier hosts (step b in that point), because the cloudgate configuration resides in the replicated block volume (under /u01/data), so it is already a copy of primary.

  11. Verify the DR Setup
     At this point, the DR environment setup is completed. Oracle recommends to immediately validate that the DR setup is correct, by performing a complete Switchover (see point Switchover in next page) or, alternatively, by opening the secondary site for validation (see point Open Secondary Site for Validation in next pages).
Lifecycle Operations for BV Replication DR model

Switchover

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. 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 and are required only if they were not performed as post-steps in the last switchover.

- If still attached, detach original or previously used block volumes from the mid-tier hosts in Site2 (umount, detach, and comment the entry from the /etc/fstab).
- Delete or rename the detached volumes in Site2 to prevent from mounting them by mistake. They will not be used anymore.

b) Switchover

The actual switchover procedure starts at this point:

<table>
<thead>
<tr>
<th>SWITCHOVER STEP</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop processes in primary Site</td>
</tr>
<tr>
<td>Stop WebLogic Administration Server, managed servers and node managers in Site1.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Switchover Virtual Front-end DNS name</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Activate the replicas in Site2</td>
</tr>
<tr>
<td>Until this point, the BV are being continuously replicated from of Site1 to Site2. In order to mount the replicated BV in Site2, the BV replicas of Site 2 need to be activated. When you activate a BV replica, an &quot;attachable&quot; BV is created as a clone from the replicated BV. Then, these cloned BV can be attached to the compute instance (note that the entities &quot;block volume replica&quot;, which are shown in BLOCK VOLUMES &gt; BLOCK VOLUME REPLICA, cannot be attached to a compute instance.)</td>
<td></td>
</tr>
<tr>
<td>To activate the replicas in Site2, connect to OCI Console</td>
<td></td>
</tr>
<tr>
<td>- Go to Site2. Block Volumes &gt; Block Volume Replicas</td>
<td></td>
</tr>
<tr>
<td>- Click in each BV replica and &quot;Activate&quot;</td>
<td></td>
</tr>
<tr>
<td>- In the replica name, Oracle recommends to use the same name regardless the region where they are created. Preferably, use the same name than the original Block Volume. This will facilitate automations using OCI CLI. Example: &quot;wls-prefix-data-block-N&quot; (or &quot;wlsprefix-mw-block-N&quot; if mw are also replicated)</td>
<td></td>
</tr>
<tr>
<td>- Repeat for all the BV replicas in Site2.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Attach the replicated BV to Site2 mid-tier hosts</td>
</tr>
<tr>
<td>The attachable Block Volumes created as a result of the activation must be shown in Site2, in OCI Console &gt; Storage &gt; Block Volume</td>
<td></td>
</tr>
<tr>
<td>For each mid-tier host in Site2:</td>
<td></td>
</tr>
<tr>
<td>- Attach the appropriate block volumes to the host.</td>
<td></td>
</tr>
<tr>
<td>You can do this in OCI Console, in Compute &gt; Instances &gt; (click on the instance) &gt; Attached block Volumes &gt; Attach Block Volume</td>
<td></td>
</tr>
<tr>
<td>- Once attached, click on &quot;iSCSI Commands &amp; Information&quot; of the attached block volume and run the iscsi commands provided in &quot;Commands for connecting&quot; in the mid-tier host.</td>
<td></td>
</tr>
<tr>
<td>- Once iscsi commands are run, get the UUID of the new attached block volume:</td>
<td></td>
</tr>
<tr>
<td>[root@wloscpREFIX-wls-0 opc]# sudo blkid /dev/sda3: UUID=&quot;974147f5-d731-41de-bb88-beff78edc9c&quot; TYPE=&quot;xfs&quot;</td>
<td></td>
</tr>
<tr>
<td>PARTUUID=&quot;4a95c68a-bc70-4be8-bce8-b15e995fcf46&quot;</td>
<td></td>
</tr>
</tbody>
</table>
/dev/sda1: SEC_TYPE="msdos" UUID="593B-B893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="c5ac3089-6a92-40e0-bcc1-212ba0b43418"
/dev/sda2: UUID="9ca12daa-d7ea-44a2-8680-5b676488b054" TYPE="swap" PARTUUID="682a63d1-d3ec-4019-b372-43720aaaee717"
/dev/sdb: UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4"

- **Add an entry for the appropriate UUID in */etc/*fstab**, to mount and persist the mount after reboots.
  
  UUID=35e72262-979a-4d84-85ce-a6f91e3b1250 /u01/data ext4 auto,defaults,_netdev,nofail

  **NOTE:** after the first switchover, this UUID will not change for each block volume, so you can just comment/uncomment the entry in */etc/*fstab*, depending of the role of the Site.

- **Mount** the new attached block volume in the */u01/data* and verify that it is correctly mounted.
  
  [root@wlsoicpfx-wls-0 opc]# mount -a
  [root@wlsoicpfx-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. In this case, make sure you mount the proper UUID in the correct path. You can complete one mount at a time to make sure.

  Repeat the steps all the mid-tier hosts in Site 2.

5 Run the replacement script in Site 2 mid-tier hosts

Run the script XXXXX in all the Site 2 mid-tier hosts.

6 Switchover Database

Use DG broker in primary db host to perform the switchover. As user oracle:

[oracle@drdbwlmot -]# dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> switchover to "secondary_db_unqname"

7 Start the servers in Site 2 (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',"acme1234",'wlsoicpfx-wls-0',"5556","wlsoicpfx_domain","/u01/data/domains/wlsoicpfx_domain","SSL")
wlst> nmStart("wlsoicpfx_adminserver")
```

Start secondary managed servers (use the secondary WebLogic Console or scripts)
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 (from Site2 to Site1), in the block volumes of the mid-tier hosts in Site2. Make sure you provide the appropriate Availability Domain of Site 1 for the replicas.
- Disable the replica in the Site1 block volumes.
- Detach the block volumes from Site1 mid-tier hosts to prepare them for the future switchback (umount, comment the entry in fstab, run the iscsi commands to disconnect and detach).
- Delete or rename the detached volumes from the Site1 mid-tier hosts to prevent from mounting them by mistake. They will not be used anymore.
Failover

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.

Failover steps are the same than switchover. The only 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@drdbwlmp1b ~]$ 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.

Open Secondary Site for Validation

In this model, it is also possible to validate the standby site without performing a complete switchover by converting the standby database to snapshot standby. This allows the secondary WLS servers to be started 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:
<table>
<thead>
<tr>
<th>STEPS TO OPEN THE STANDBY SITE FOR VALIDATIONS</th>
<th>DETAILS</th>
</tr>
</thead>
</table>
| 1 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. |
| 2 Activate the replicas in Site2 | Until this point, the BV are being continuously replicated from of Site1 to Site2. In order to mount the replicated BV in Site2, the BV 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, these cloned BV can be attached to the compute instance (note that the entities "block volume replica", which are shown in BLOCK VOLUMES > BLOCK VOLUME REPLICAS, cannot be attached to a compute instance.) To activate the replicas in Site2, connect to OCI Console
- Go to Site2. BLOCK VOLUMES > BLOCK VOLUME REPLICAS
- Click in each BV replica and "Activate"
- In the replica name, Oracle recommends to use the same name regardless the region where they are created. Preferably, use the same name than the original Block Volume. This will facilitate automations using OCI CLI. Example: "prefix-data-block-N" (or "prefix-mw-block-N" when the mw are also replicated)
- Repeat for all the BV replicas in Site2. |
| 3 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 For each mid-tier host in Site2:
- Attach the appropriate block volumes to the host.
- Once attached, click con "iSCSI Commands & Information" of the attached block volume and run the iSCSI commands provided in "Commands for connecting" in the mid-tier host.
- Once iSCSI commands are run, get the UUID of the new attached block volume:
  ```
  [root@wlsociprefix-wls-0 opc]# sudo blkid
  /dev/sda3: UUID="974147f5-d731-41de-bb8b-56ff8ed19c9c" TYPE="xfs" PARTUUID="4a95c68a-bc70-4b8b-84ce-b15e9578ed19c9c"
  /dev/sda1: SEC_TYPE="msdos" UUID="593b-b893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="caac3089-6a41-4000-bc11-212baa94a18"
  /dev/sda2: UUID="9c12-da-a4a2-8680-5267488b054" TYPE="swap" PARTUUID="68263d1-d3ec-4019-b372-43720a6e717"
  /dev/sdb: UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4"
  ``
- Add an entry for the appropriate UUID in /etc/fstab, to mount and persist the mount after reboots
  ```
  UUID=35e72262-979a-4d84-85ce-a6f91e3b1250 /u01/data ext4 auto,defaults,_netdev,nofail
  ```
  NOTE: after the first switchover, this UUID will not change for each block volume, so you can just comment/uncomment the entry in /etc/fstab, depending of the role of the Site.
- Mount 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%
  ```
4. Run the script that makes the replacements in Site2 mid-tier services:

Run the script XXXXX in all the Site2 mid-tier hosts.

5. Start the servers in the site2:

Start the nodemanager in all the secondary servers. Example:

```bash
$ cd $DOMAIN_HOME/bin/
$ nohup ./startNodeManager.sh > $DOMAIN_HOME/nodemanager/nodemanager.out 2>&1
```

Start the secondary admin server. Example:

```bash
$ cd /u01/app/oracle/middleware/oracle_common/common/bin
$ ./wlst.sh
wlst> nmConnect ('weblogic','acme1234#','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).

6. Validate:

Note: As this is not a swichover and the primary site is still active, the virtual front-end name will resolve to the primary site's LBR IP address, so any browser access will, by default, be redirected to the active primary site.

In order 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, and 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 the secondary site has been validated, follow these steps to revert it back to standby role again:

<table>
<thead>
<tr>
<th>STEPS TO REVERT BACK STANDBY TO STANDBY ROLE</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Stop processes in secondary Site2</td>
<td>You can connect to secondary WebLogic Console and shutdown managed servers and Admin server in secondary site. Stop the node manager processes too.</td>
</tr>
</tbody>
</table>
| 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 back any updated /etc/hosts in clients | If you updated the virtual front-end name in the /etc/hosts file of a client, in order 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 each mid-tier host in Site2, run the following:

- Unmount the replicated Block Volumes:

[opc@wlsociprefix-wls-o opc]# sudo umount /u01/data |
Scale-out and Scale-in

The procedure for scale-out and scale-in operations in the Block Volume Cross-Region DR model has slight differences with the DBFS and RSYNC models.

This is the procedure to scale-out 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 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) You can now proceed with the steps described in the point Scale-out of this document to scale-out primary and secondary. Ignore any mentions to the config_replica.sh script; it does not apply to this model.

c) Then, revert back the secondary to the standby role, by detaching the volumes, as described in the previous point Open Secondary Site for Validation.

d) For the Block Volumes 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.

NOTE: The terraform scripts used by WLS to scale-out create redundant/unnecessary BV 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 mounts pointing to the incorrect block volume. Make sure you delete the duplicated block volumes created by the scale-out job.

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 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) Disable the Cross-Region replica in the block volumes of the primary node that is going to be deleted. The scale-in job will fail to delete a block volume that has the cross-region replica enabled.

c) You can now proceed with the steps described in the point Scale-in of this document to scale-in primary and secondary. Ignore any mentions to the config_replica.sh script; it does not apply to this model.

d) Then, revert back the secondary to the standby role, by detaching the volumes, as described in the previous point Open Secondary Site for Validation.
NOTE: The terraform scripts used by WLS to scale-in create redundant/unnecessary BV 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 block volume. Make sure you delete the duplicated block volumes created by the scale-in job.

Using OCI CLI commands for automating Block Volume Replication tasks

Most of the actions related with Block Volumes (enable/disable replication, activation, attach/detach block volumes) can be performed not only using the OCI Console, also using the OCI Comman Line Interface (CLI) commands.

Here is a list of the commands for the most common actions used during the setup and switchover/failover of this DR model. For information about using the CLI, see Oracle Cloud Infrastructure CLI Command Reference and Cross-Region Volume Replication documentation.

- Enable cross-region replication for a Block Volume
  
  `oci bv volume update --volume-id <volumeId> --block-volume-replicas '[{"displayName":"<replicaDisplayName>"","availabilityDomain":"<availabilityDomain>"}]'

- Disable cross-region replication for a Block Volume
  
  `oci bv volume update --volume-id <volumeId> --block-volume-replicas '[]'

- Activate the Block Volume replication
  
  `oci bv volume create --source-volume-replica-id <volumeReplicaId> --compartment-id <compartmentId> --availability-domain <AvailabilityDomain> --display-name <displayName>

- Attach a Block Volume (iscsci type) to a compute instance
  
  `oci compute volume-attachment attach-iscsi-volume --instance-id <instanceId> --volume-id <volumeId>

- Detach a Block Volume from a compute instance
  
  `oci compute volume-attachment detach --volume-attachment-id <volumeAttachmentId>

As shown above, you need the OCIDs of the Block Volumes and volume attachments for running the OCI CLI commands. These OCIDs change after every switchover. But these values can be retrieved using OCI CLI scripts as well. As long as you use consistent "display names" for the block volumes objects, you can use the display names to get the OCID dynamically.

However, not all the steps can be automated with OCI CLI commands. Some actions, like attaching or detaching Block Volumes, require a combination of OCI Console (or OCI CLI) actions plus Operating System commands (edit fstab, mount, run iscsi commands, etc.). Here is an example of a semi-automation script for the switchover in a WLS for OCI with Block Volume replication DR model.