Oracle® Cloud

Configuring a Customer-Managed Disaster Recovery Solution for Oracle Integration 3

F91135-02
July 2024

Oracle Cloud Configuring a Customer-Managed Disaster Recovery Solution for Oracle Integration 3,

F91135-02

Primary Author: Oracle Corporation

Contents

Preface

This document describes how to configure a disaster recovery solution for Oracle Integration.

Topics:

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Resources
- Conventions

Audience

This document is intended for personnel who are responsible for configuring a disaster recovery solution for Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see the Oracle Integration documentation in the Oracle Cloud Library on the Oracle Help Center.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

Oracle Integration is available in an Oracle Cloud Infrastructure (OCI) region governed by service-level agreements (SLAs). This guide details the procedure to build a cross-region, customer-managed disaster recovery solution for Oracle Integration, specifically for the Integrations and File Server features in Oracle Integration.

Disaster Recovery Concepts

A disaster recovery (DR) solution enables you to fail over quickly from natural or human-made disasters and provide business continuity in your secondary region. In addition, you can use the DR setup for planned migrations and switch between different regions periodically.

Before you set up any DR solution, understand the following:

- Determine the recovery point objective (RPO) and recovery time objective (RTO) for your service. The RTO is the target time within which your service must be restored after a disaster occurs. The RPO is the period after a disaster occurs during which the service can tolerate lost data before the disaster begins to affect the business.
- The DR solution described in this guide replicates only design-time artifacts; therefore, the RPO is not applicable in this scenario. You can measure the RTO based on how quickly you’re able to complete the failover tasks in the secondary region.
- This DR solution only supports an active-passive configuration. Only one instance or region of the DR configuration can be active and processing transactions at a time. Other details on the scope and constraints of the solution are provided in the following sections.

2

About the Solution Scope

Learn about the scope and constraints of the DR solution.

Topics:

What's Supported?

This DR solution supports:

- Active-passive topologies.
  Active passive topologies mean that only one instance processes all the load even though both instances are expected to be up and running. An active instance is determined by which instance the traffic is directed to and is enabled to execute all functions, and not by its start or stop status. It is recommended that you not turn off your disaster recovery instance.
- Customer-managed replication of design time metadata as defined. See Automate Metadata Synchronization.
- Events.
  If both primary and secondary integrations use the same connection metadata and credentials, these integrations, if activated, are triggered.
- Integrations with scheduled parameters or polling triggers require special handling. See Key Considerations.
- Integrations (replication of the design-time artifacts only).
- File Server.

What's Not Supported?

This DR solution does not support:

- Processes.
- B2B for Oracle Integration.
- Visual Builder.
- In-flight transactions cannot be failed over. They must be retried after failover to the secondary instance is successful.

> **Note:**
>
> These are independent instances with respective audit and monitoring data.

3

Disaster Recovery for Integrations

Learn about the architectural view of the DR solution for the integrations feature in Oracle Integration. Additionally, you can learn more about the key considerations, prerequisites, and procedures to set up the solution.

Topics:

Architecture

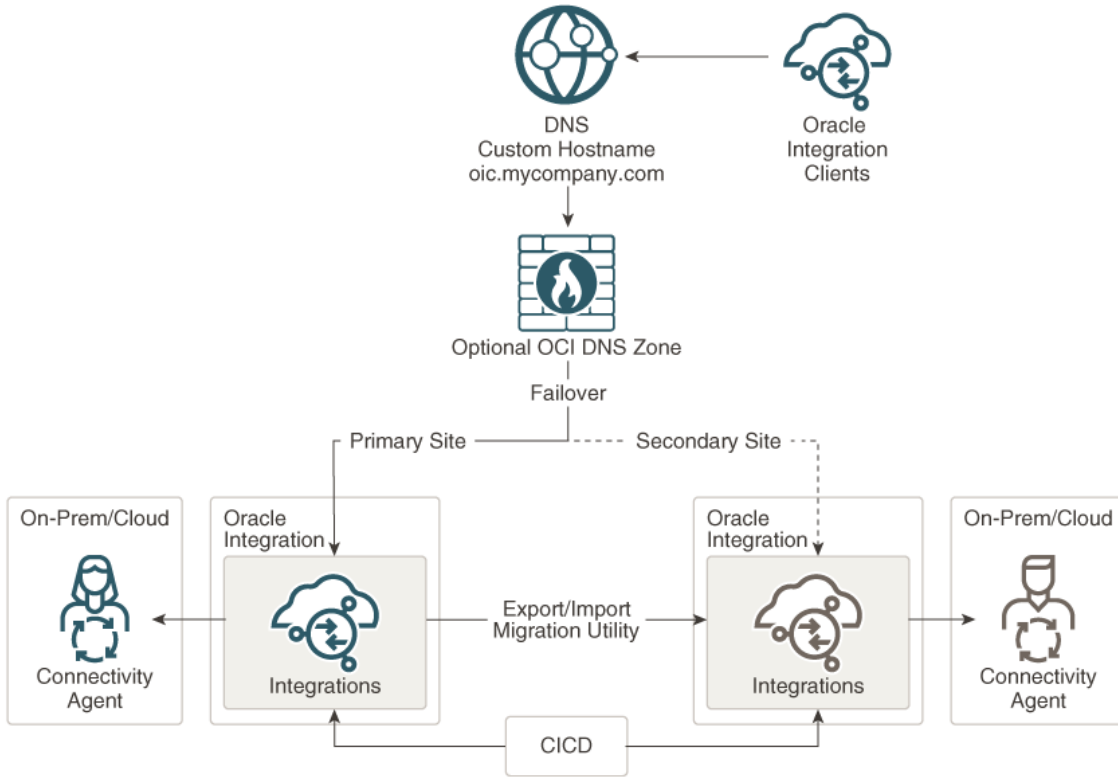
Learn about the architecture of the DR solution for the integration feature.

The DR architecture for Integrations consists of two Oracle Integration instances in two different cloud regions, which are accessed using a single custom endpoint (URL). Optionally, you can use an OCI Domain Name System (DNS) zone to resolve the custom endpoint name.

Custom endpoint URLs are a feature provided by Oracle Integration that allow you to define a custom URL in a domain that you manage. You can use these URLs as the entry point to the Oracle Integration instances. For example, prod-integration.mycompany.com. For more information about configuring custom endpoints, see Configure a Custom Endpoint for an Instance in *Provisioning and Administering Oracle Integration 3*.

The two Oracle Integration instances in the architecture are designated as primary and secondary, and both the instances run concurrently; however, only one of the instances receives traffic. Initially, it's the primary instance that receives the traffic flow. When this instance becomes unavailable, the DNS record is updated to route the traffic to the secondary instance. The following image shows this architecture in detail for a scenario using connectivity agents:

> **Note:**
>
> Custom endpoint support is provided for the runtime URL (`oic.mycompany.com` in the architecture diagram below), but not for the design-time URL. To log in to Oracle Integration at design time, you must use the URL provided during service instance provisioning.



You must update the DNS record at your DNS provider to switch between instances. Optionally, you can implement an OCI DNS zone to manage the subdomain related to the Oracle Integration custom hostname. The OCI DNS zone can reflect changes to the CNAME much faster.

In this setup, you must synchronize the Oracle Integration metadata at both the sites using continuous integration and continuous deployment (CICD).

Key Considerations

Consider the following important information before you start implementing the DR solution:

- Scheduled parameters in integrations may not be the latest in the secondary instance; this depends upon how often you update them. Therefore, you must frequently update integrations with scheduled parameters in the secondary Oracle Integration instance. See Automate Scheduled Parameters Updates.
- If a connectivity agent is required, you must provision and configure separate connectivity agents for the primary and secondary Oracle Integration instances.
- When the primary instance is active, you must stop or deactivate scheduled flows and integrations with polling triggers in the secondary instance. After a failover, activate these artifacts in the secondary instance.
- The monitoring data (message history) is not replicated. You can use the OCI Logging service to extract the message history in your data lake or warehouse to receive a global view.
- Ensure that your source applications and end users use the custom endpoint to access the Oracle Integration instance. Your administrators must use the respective original instance host names to manage the instances.

- When a failover to the secondary instance is required, you must manually change the routing by updating the DNS record at your DNS provider. Here are some additional points to note about the failover process:
  - It's quick and simple for nonscheduled and nonpolling integration flows.
  - Scheduled flows and integrations with polling triggers require special handling.

Prerequisites

Ensure that all prerequisites for the integrations DR solution are met before you begin the configuration process.

Before you begin the DR configuration process for integrations, you must:

- Provision a second Oracle Integration instance in a different OCI region with at least one message pack. However, to ensure that the secondary instance handles the usual volume, provision it with the same number of message packs as your primary instance.
- Provision an OCI Object Storage bucket for metadata migration.
- Obtain a custom host name (in a domain of your choice) and an SSL certificate for it.

Configure a Custom Endpoint

This procedure allows applications and users to access the same URL.

Configure a common custom endpoint for your Oracle Integration instances, so that applications and users can access Oracle Integration with the same URL regardless of which instance is active in the background.

> **Note:**
>
> Custom endpoint support is provided for the runtime URL, but not for the design-time URL. To log in to Oracle Integration at design time, you must use the URL provided during service instance provisioning.

See Configure a Custom Endpoint for an Instance in *Provisioning and Administering Oracle Integration 3* for the full list of tasks.

Set Up the Disaster Recovery Solution for Integrations

Perform the following one-time configuration tasks to set up the disaster recovery solution for the integrations feature. As part of the configuration, you can also set up an OCI DNS zone if necessary.

Overview of configuration tasks:

1. Create two Oracle Integration instances in two different OCI regions; for example, one in Ashburn and another in Phoenix. See Creating an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*.
2. You can do one of the following:
   - Register a custom host name in your own DNS.

- Use an OCI DNS zone. Create an OCI DNS zone to manage the subdomain related to the Oracle Integration custom host name.

  See Use an OCI DNS Management Zone.

3. Configure a common custom endpoint for both primary and secondary Oracle Integration instances.

   Note:

   If you are using a connectivity agent, you must provision a dedicated connectivity agent for each Oracle Integration instance and use the original instance host names in the agents' configurations.

   See Configure a Custom Endpoint.

4. Use REST APIs or the Oracle Integration UI to migrate the metadata from the primary to secondary instance for the first time. See Migrate Metadata from the Primary Instance.

Once you have configured the DR environment by executing these tasks, verify your system end-to-end. Access the custom endpoint and navigate through the Oracle Cloud Infrastructure Console.

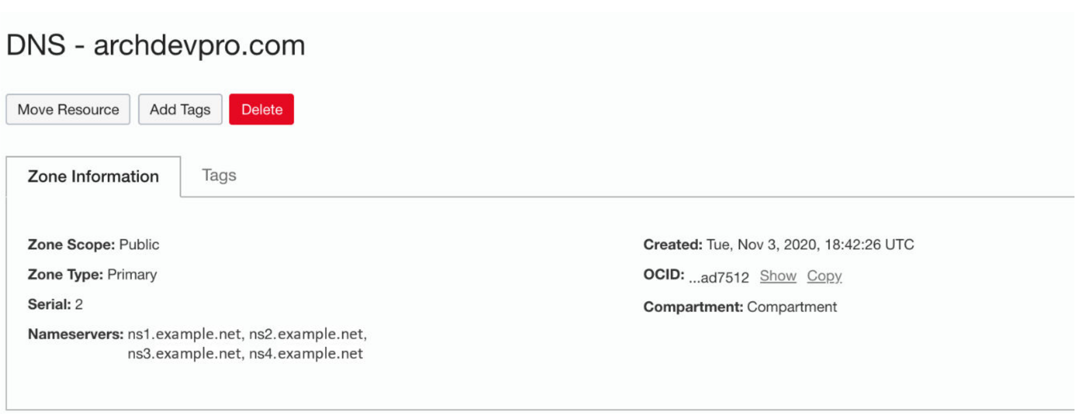
Use an OCI DNS Management Zone

Configure DNS records for your Oracle Integration instances.

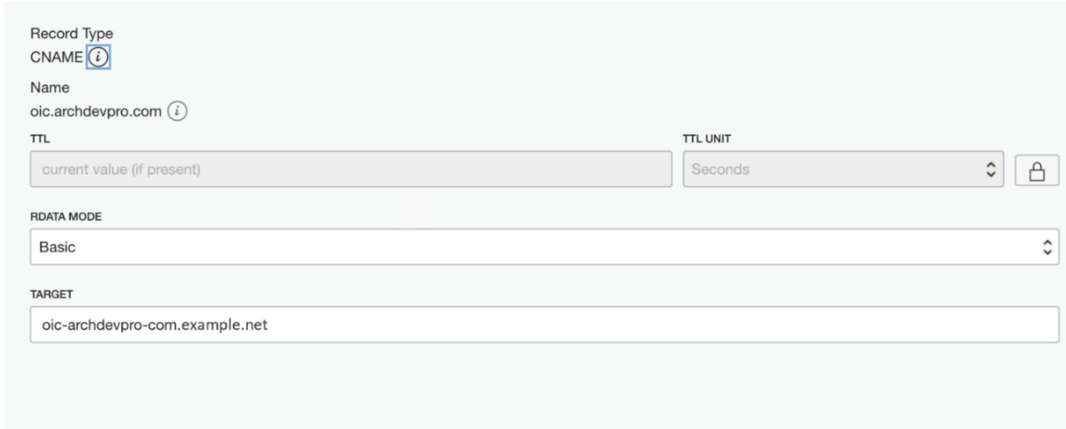
You can use an OCI DNS zone to manage DNS records and provide host name resolution for your Oracle Integration instances.

1. After you've acquired a domain (or a subdomain) for your Oracle Integration instance, add an OCI DNS zone through the Oracle Cloud Infrastructure Console or the API. For details on creating an OCI DNS zone and adding a record to it, see Managing DNS Service Zones.

   The following image shows a sample DNS zone created for the domain named `archdevpro.com`.



2. In the zone, add the Oracle Integration custom host name as a CNAME record.



3. After you've successfully published the above changes, update your domain to use the OCI DNS name servers.

Migrate Metadata from the Primary Instance

Export the metadata from the primary instance to the standby instance.

The Oracle Integration metadata consists of connections, integrations, lookups, libraries, projects, and packages. After you've configured the primary instance with all your integration deployments, you can export the metadata from the instance and import it into the standby instance. This as an initial, one-time task. You can export the metadata using any one of the following methods:

- Use REST APIs to export the metadata and import the same.
- Use the Oracle Integration UI to export and import the metadata. For details, see Export and Import Design-Time Metadata Between Instances in *Provisioning and Administering Oracle Integration 3*.

> **Note:**
>
> Before you use either of the preceding methods, you require an OCI Object Storage bucket to store the artifacts.

Subsequently, you can employ continuous integration continuous deployment (CICD) to have the metadata synchronized between instances. See Automate Metadata Synchronization.

> **Note:**
>
> - You must use the original instance host names for all administrative tasks, including metadata migration.
> - It's recommended that you import the artifacts without activating them so that the connections created in the secondary instance will not be in the *Activated* state. Once you have imported the artifacts, you can manually test the connections and activate them. Else, you can also use Connections REST Endpoints in *REST API for Oracle Integration 3* for the same.

To synchronize the metadata between the instances using REST APIs:

1. Export the metadata from the primary instance. Invoke the REST API using the following postman or curl command. This action uploads the metadata to the OCI Object Storage Cloud Service bucket instance.

   ```
   POST
   ```

   ```
   http:/host:port/ic/api/common/v1/exportServiceInstanceArchive
   ```

   Request Headers:

   ```
   Content-Type →application/json
   ```

   Request Payload:

   ```
   {
             "jobName":"Pod1_Metadata"   -  If jobName is omitted filename
   will default to
             "Service_Instance_Data-<jobId>",     "overwrite":false,      -
   defaults to false, will return error if
             archive file already exists
   "exportSecurityArtifacts":true,
             "exportAppRoleMembers":true,     "description":"Export
   description",
                 "storageName",   - name of storage configuration, this can
   be used instead of
             storageInfo, if both are defined storageInfo will take
   precedence
             "storageInfo":{
             "storageUrl":"https://swiftobjectstorage.us-
   ashburn-1.oraclecloud.com/v1/paasdevoic/<bucket
             name>",
             "storageUser":"<OCI user name>",
             "storagePassword":"<Auth Password>”     }
         }
   ```

   Response Headers:

   ```
   Location →http://host:port/ic/api/common/v1/exportServiceInstanceArchive/483
   ```

   Response Payload:

   ```
   {
             “jobId”: “483”, “location”:
             “https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/
   paasdevoic/<bucket
             name>”, “status”: “NOT_STARTED”
            }
   ```

   Response Status:

   - 202 Accepted – Export job was accepted.
   - 409 Conflict – Import or export job is already running or storage details are incorrect/missing, or the file already exists (if overwrite is set to `false`).
   - 500 Internal Server Error – Error communicating to the registry or storage.

2. Check the status of the export operation using the following command:

   ```
   GET
   ```

```
http://host:port/ic/api/common/v1/exportServiceInstanceArchive/{jobId}
```

If the status is *Completed*, the metadata has been successfully exported to the object storage bucket.

3. Now, import the metadata into the standby instance. Invoke the REST API using the following postman or curl command.

   This action retrieves the archive from the OCI Object Storage Cloud Service bucket instance where the archive was initially created.

   ```
   POST
   ```

   ```
   http://host:port/ic/api/common/v1/importServiceInstanceArchive
   ```

   Request Headers:

   ```
   Content-Type →application/json
   ```

   Request Payload:

   ```
   {
             "archiveFile":"archive_Local_Suite_Instance-483.zip",
             "importActivateMode":"importOnly",      // ImportOnly |
   ImportActivate |
             ActivateOnly | StartSchedulesOnly
             "importSecurityArtifacts":true,
             "importAppRoleMembers":true,
             "importScheduleParams":true,
             "startSchedules":false,
             "description":"Import to standby",
             "storageName",   - name of storage configuration, this can be
   used instead of
              storageInfo, if both are defined storageInfo will take
   precedence
             "storageInfo":{
               "storageUrl":"https://swiftobjectstorage.us-
   ashburn-1.oraclecloud.com/v1/paasdevoic/<bucket
             name>",
               "storageUser":"OCI cloud user name",
               "storagePassword":”Auth password"
                 }
             }
   ```

   > **Note:**
   >
   > Set the *importActivateMode* variable to *ImportOnly*, so that the integration flows are imported but aren't activated.

   Response Payload:

   ```
   {
            "jobId": "457", "status": "NOT_STARTED"
            }
   ```

   Response Status:

   - 202 Accepted – Export job was accepted.

- 409 Conflict – Import or export job is already running or storage details are incorrect/ missing, or the file already exists (if overwrite is set to `false`).
- 500 Internal Server Error – Error communicating to the registry or storage.

4. Verify the import status.

```
GET

https://host:port/ic/api/common/v1/importServiceInstanceArchive/457
```

Where *457* is the Job ID from the import response payload.

> **Note:**
>
> In this example, the integrations imported are not activated, conforming to the best practice. However, if you have many integrations, you can activate the stateless integrations while importing, but do not activate scheduled, publish-subscribe, polling, or business events integrations.

Perform Post-Configuration Tasks

Learn how to regularly update integrations with scheduled parameters between your primary and standby integration instances.

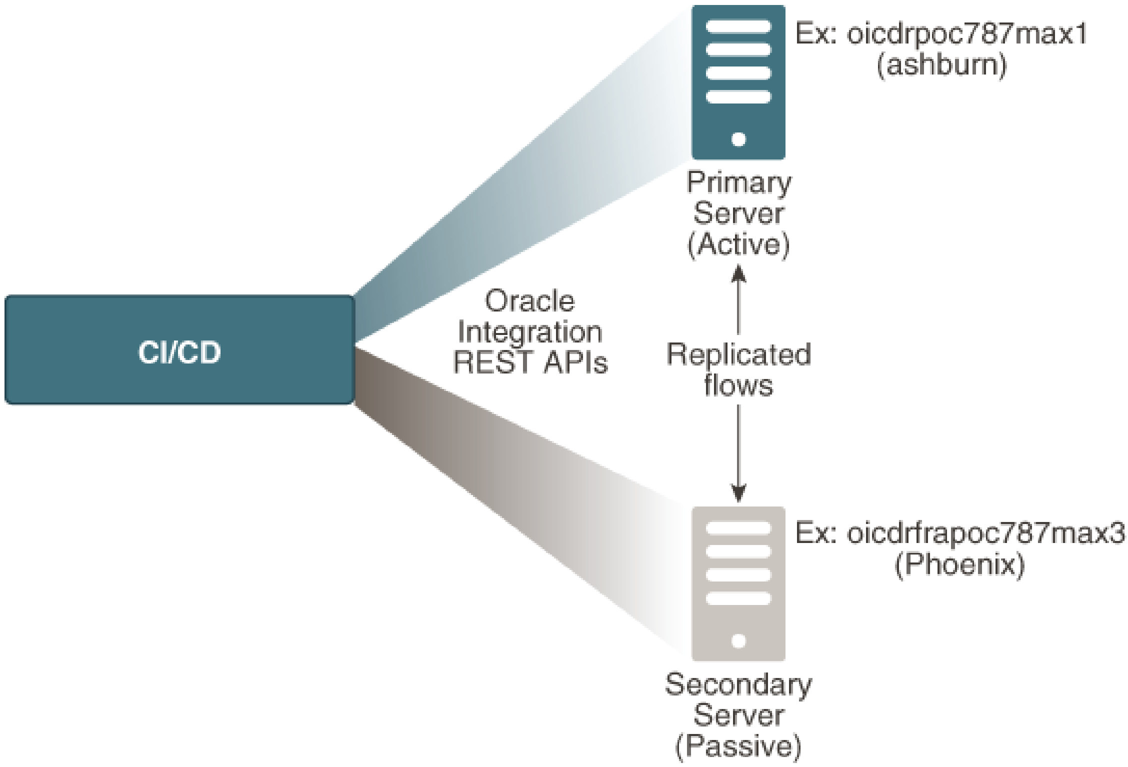
After the DR system is set up, you need to regularly update integrations with scheduled parameters in the secondary Oracle Integration instances, monitor the instances for a failover, and execute failover-handling tasks when necessary.

- Automate Metadata Synchronization
- Automate Scheduled Parameters Updates
- Monitor Your Instances

Automate Metadata Synchronization

Ensure that the metadata is regularly synchronized between the primary and the standby instance using CICD.

After the initial, one-time migration of the metadata is complete, you must keep the metadata synchronized between the instances using CICD. You can use Jenkins or a similar tool to implement CICD for your instances and have the metadata synchronized. You can also use an OCI Compute instance as the Jenkins CI server and CD hub. See Integrations REST Endpoints and Connections REST Endpoints in *REST API for Oracle Integration 3* for the REST APIs to use. The following figure shows the CICD forking to both the instances:



Automate Scheduled Parameters Updates

Update the integrations with scheduled parameters in the standby Oracle Integration instance.

You must frequently retrieve the metadata with scheduled parameters of integrations from the primary instance; for example, you can execute this every fifteen or thirty minutes. Use REST APIs to update the integrations with the extracted scheduled parameters in the standby Oracle Integration instance. Subsequently, you can choose to update the corresponding integrations in the standby instance either periodically or during a failover.

To get the details of each scheduled integration from the primary instance, see Retrieve an Integration in *REST API for Oracle Integration 3*. To update an integration in the standby instance, see Update Scheduled Integration Parameters. After a failover, you may have to manually update the parameter values to avoid reprocessing of old data.

> **Note:**
>
> Optionally, you can use the export-import APIs, see Migrate Metadata from the Primary Instance.

Monitor Your Instances

Regularly monitor the health of your active Oracle Integration instances.

You can use the OCI health check service or a third-party monitoring service. Additionally, define a process to identify outages and, subsequently, trigger failovers.

Execute Failover Tasks

Learn how to manually switch to the DR environment.

To switch from your primary instance to the standby instance during outages:

1. Stop the primary Oracle Integration instance from the Oracle Cloud Infrastructure Console.
2. Prepare your secondary instance.
   a. If you are not using CICD, then ensure that the standby (secondary) instance contains the latest versions of integrations. You can use the import API to import the latest snapshot extracted.
   b. Activate all relevant integrations.
   c. Update the scheduled parameters with the latest values to avoid reprocessing of old data.
3. Update the DNS record at your DNS provider or in the OCI DNS zone to route the traffic to the standby instance.

After the failover process, the standby instance becomes your primary instance, and the instance previously designated as primary becomes the new standby instance.

> **Note:**
>
> - If your original primary instance restarts itself after the failover process, deactivate or shut down scheduled and polling-based integrations.
> - If there are backlogs (of asynchronous transactions) in the original primary instance, these may be triggered when the instance restarts, resulting in duplicate transactions. The backlogs belong to the faulted instances of integrations. You can choose how you handle them.

4 Disaster Recovery for File Server

Learn about the architecture, prerequisites, and the procedure required to set up the DR solution for the File Server feature in Oracle Integration

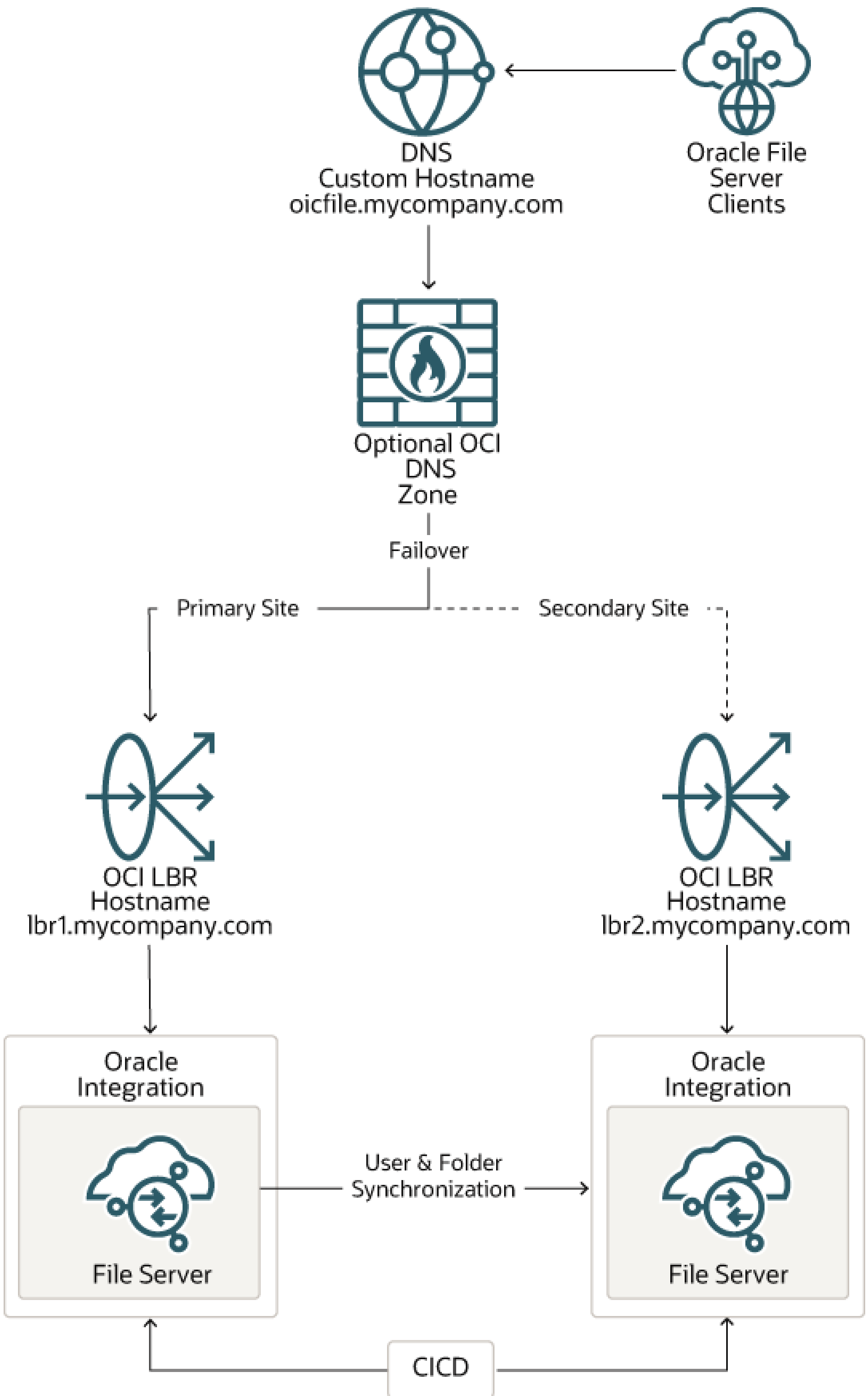
Topics:

Architecture

The DR architecture for File Server is similar to the integrations DR topology. It consists of two File Server instances (designated as primary and secondary) in two different cloud regions, which are accessed using a single custom endpoint.

It also includes two public load balancers, one for each File Server instance, that direct traffic from the internet to the respective instances. Optionally, you can use an OCI DNS zone as a front-end to route traffic to the load balancers. The following image shows the architecture in detail:

> **Note:**
>
> Custom endpoint support is provided for the runtime URL (`oicfile.mycompany.com` in the architecture diagram below), but not for the design-time URL. To log in to Oracle Integration at design time, you must use the URL provided during service instance provisioning.


DNS
Custom Hostname
oicfile.mycompany.com
Oracle File
Server
Clients
Optional OCI
DNS
Zone
Failover
Primary Site
Secondary Site
OCI LBR
Hostname
lbr1.mycompany.com
OCI LBR
Hostname
lbr2.mycompany.com
Oracle
Integration
File Server
User & Folder
Synchronization
Oracle
Integration
File Server
CICD

You must implement CICD to keep the users and folder data synchronized at all times between the two sites. During a failover, you must update the DNS record at your DNS provider or in the OCI DNS zone to switch between File Server instances.

Prerequisites

Ensure that all prerequisites for the File Server DR solution are met before you begin the configuration process.

Before you begin the disaster recovery configuration process for File Server, you must:

- Create two Oracle Integration instances in two different OCI regions. If you've already created these instances while configuring the Integrations DR, you can use the same instances.
- Enable File Server on both the Oracle Integration instances. See Enable File Server in *Using File Server in Oracle Integration 3*.
- Note the public IP address and port number of each File Server instance.
- Create the following resources for public load balancers in both the OCI regions:
  - A Virtual Cloud Network (VCN) and an internet gateway.
  - A public regional subnet in the VCN, with an associated security list and a route table.

Set Up the Disaster Recovery Solution for File Server

Perform the following one-time configuration tasks to set up the disaster recovery solution for File Server:

1. Create load balancers in the two OCI regions where your Oracle Integration instances exist. See Create Load Balancers.
2. Configure a separate custom host name for each load balancer. See Configure Custom Host Names for Load Balancers.
3. Configure a common custom endpoint for both the load balancers. See Configure a Common Custom Endpoint.

Create Load Balancers

Create a public load balancer in both primary and secondary OCI regions to direct traffic to your File Server instances.

For more information on load balancer prerequisites, see Load Balancer Management.

To create a load balancer:

1. Log in to your Oracle Cloud Infrastructure Console, and select the required region from the **Regions** drop-down list at the top of the page.
2. For the full list of tasks to create a load balancer, see Load Balancer Management . The following steps list the important configurations:
   a. On the Add Details page of the Create Load Balancer dialog:
      i. Leave **Public** and **Ephemeral IP Address** selected in the visibility type and IP address sections, respectively.
      ii. In the Bandwidth section, choose the required bandwidths.

iii. In the Choose Networking section, select the appropriate VCN and subnet.

b. On the **Choose Backends** page, specify the details for the backend set:

i. Select the required load balancing policy.

ii. Do not add a backend at this time.

iii. In the Specify Health Check Policy section:

i. In the **Protocol** field, select **TCP**.

ii. In the **Port** field, enter the port number of the File Server instance corresponding to the region.

iii. Enter the required interval, timeout, and retry values.

c. On the **Configure Listener** page:

i. Select **TCP** as the protocol.

ii. Enter **22** as the port on which to listen for incoming traffic.

d. On the **Manage Logging** page:

i. Select the compartment in which you want to store the log file.

ii. Select the log group in which to store the log file. You can either select the default log group or create a new log group. If you choose to create a new group, enter a name and description for the group.

iii. Enter a name for the log file, and choose a retention period.

iv. Click **Submit**.

After the system provisions the load balancer, the load balancer record appears on the page.

3. Add the IP address of the File Server as a backend to the load balancer. For complete details, see Backend Server Management.

a. On the **Load Balancers** page, click the name of the load balancer record to open it.

b. On the **Load Balancer Details** page, scroll down to the Resources section, and click **Backend Sets**. The backend set you configured while creating the load balancer is listed on the page.

c. Click the name of the backend set to open it.

d. On the **Backend Set Details** page, click **Backends** in the Resources section on the left.

e. Click **Add Backends**.

f. In the **Add Backends** dialog:

i. Select the **IP Addresses** option.

ii. Enter the IP address and port number of the File Server instance corresponding to the region.

iii. Click **Add**.

After the load balancer is updated, test your sFTP connection using the load balancer's IP address and port number (22).

> **Note:**
>
> A **Warning** indicator may be displayed in the **Overall Health** field of the load balancer record. If you’ve already tested the connection, no further action is required.

Configure Custom Host Names for Load Balancers

Configure a separate custom host name for both the load balancer instances, created in your primary and secondary OCI regions.

To configure a host name for a load balancer:

1. Choose a custom host name for a load balancer instance and register it with a DNS provider.
2. After you've acquired a domain, create a DNS record for the load balancer instance. You can create and manage DNS records using an OCI DNS zone.
   a. Add an OCI DNS zone through the Oracle Cloud Infrastructure Console or the API. See Managing DNS Service Zones for details on creating an OCI DNS zone and adding a record to it.
   b. After you've created a zone, update your domain to use the OCI DNS name servers.
   c. Add a record in the OCI DNS zone for the corresponding load balancer instance. Enter the following details:
      - **Record Type**: A – IPv4 Address.
      - **Name**: Optionally, enter a subdomain.
      - **Address**: Enter the IP address of the corresponding load balancer instance.

Configure a Common Custom Endpoint

Configure a common custom endpoint for your load balancer instances.

A common custom endpoint ensures that applications and users can access File Server with the same URL regardless of which instance is active in the background.

> **Note:**
>
> Custom endpoint support is provided for the runtime URL, but not for the design time URL. To log in to Oracle Integration at design time, you must use the URL provided during service instance provisioning.

See Configure a Custom Endpoint for an Instance in *Provisioning and Administering Oracle Integration 3*.

To configure a common custom endpoint:

1. Choose a parent custom endpoint for your load balancers and register it with a DNS provider.
2. After you've acquired a domain, create a DNS record with the load balancer instance in the primary OCI region as the target.

a. Add an OCI DNS zone through the Oracle Cloud Infrastructure Console or the API.
b. After you've created a zone, update your domain to use the OCI DNS name servers.
c. Add a record in the OCI DNS zone for the load balancer instance in the primary OCI region. Enter the following details:
   - **Record Type**: CNAME.
   - **Name**: Optionally, enter a subdomain.
   - **Target**: Enter the custom host name of the load balancer instance in the primary OCI region.

3. During a failover, update the CNAME record with the custom host name of the load balancer instance in the secondary OCI region.

> **Note:**
>
> - Using a separate custom host name for each load balancer makes it easier to identify which load balancer is being used by the final custom endpoint. However, if you do not want to create parent-child custom host names for load balancers, you can create one custom host name and add a DNS A-Record for the load balancer instance in the primary OCI region initially. During a failover, you can update the A-Record with the IP address of the load balancer instance in the secondary OCI region.
> - You can also create and manage DNS records for your DR setup at a DNS provider of your choice.

Perform Post-Configuration Tasks

Learn how to regularly synchronize data between your File Server instances.

After the DR system is set up, you must regularly synchronize data between your File Server instances, monitor the instances for a failover, and execute failover-handling tasks when necessary.

- Automate Metadata Synchronization
- Monitor Your Instances

Automate Metadata Synchronization

Keep the user and folder data synchronized between the instances using CICD.

You can use Jenkins or a similar tool to implement CICD for your instances and have the data synchronized. You can also use an OCI Compute instance as the Jenkins CI server and CD hub. See File Server REST Endpoints in *REST API for File Server in Oracle Integration 3* for the REST APIs to use.

Before you begin the data synchronization, you must perform the following tasks in the secondary File Server instance, either manually or using a script:

- Replicate the primary instance’s file structure.
- Replicate the primary instance’s permission grants.
- Upload all necessary certificates.

Monitor Your Instances

Regularly monitor the health of your active File Server instances.

You can use the OCI health check service or a third-party monitoring service. Additionally, you can define a process to identify outages and, subsequently, trigger failovers.

Execute Failover Tasks

Switch from your primary instance to the standby instance during outages.

During outages, you can switch from your primary instance to the standby instance by updating the DNS record of the parent or common custom endpoint in the OCI DNS zone (or at your DNS provider). The traffic routes to the secondary File Server instance and file replication starts.

After the failover process, the standby instance becomes your primary instance, and the instance previously designated as primary becomes the new standby instance.