Oracle® Cloud

Migrating to the Cloud and Side-by-Side Upgrade in the Cloud for Oracle SOA Suite on Marketplace, Oracle SOA Cloud Service, and Oracle MFT Cloud Service

E89823-08
March 2021

ORACLE

Oracle Cloud Migrating to the Cloud and Side-by-Side Upgrade in the Cloud for Oracle SOA Suite on Marketplace, Oracle SOA Cloud Service, and Oracle MFT Cloud Service,

E89823-08

Primary Author: Oracle Corporation

Contents

3 Migration/Side-by-Side Upgrade for SOA Cloud Service

4 Migration/Side-by-Side Upgrade for MFT Cloud Service

Preface

This guide describes how to migrate on-premises SOA and MFT applications to the cloud. It also describes how to do side-by-side upgrade in the cloud for SOA Cloud Service and MFT Cloud Service.

Audience

This guide is intended for administrators who want to migrate on-premises SOA and MFT applications to the cloud. It is also intended for administrators who want to do side-by-side upgrade of SOA Cloud Service or MFT Cloud Service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud:

  http://cloud.oracle.com
- *Administering Oracle SOA Cloud Service in a Customer-Managed Environment*
- *Developing SOA Applications with Oracle SOA Suite*
- *Using Oracle Managed File Transfer Cloud Service*
- *Using Oracle Managed File Transfer*

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 to Migration and Upgrade for Oracle SOA Suite on Marketplace, Oracle SOA Cloud Service, and Oracle Managed File Transfer Cloud Service

SOA Suite on Marketplace, SOA Cloud Service, and MFT Cloud Service support manual migration from on-premises to the cloud and side-by-side upgrade from one version to another (or an older instance to a newer instance in the same version) in the cloud.

Notes:

- The steps in this guide describe migration to the following target environments:
  - SOA Suite on Marketplace 12.2.1.4
  - SOA Cloud Service 12.2.1.3 (SOA+OSB+B2B+ESS topology) or later
  - MFT Cloud Service 12.2.1.3 or later
- You can perform migration/side-by-side upgrade even if there is no version upgrade. For example, you can migrate from a subscription account to a universal credits account.

Topics:

Prerequisites for Migration and Side-by-Side Upgrade

Before migration or side-by-side upgrade, ensure the following:

- For SOA Suite on Marketplace, the source on-premises version for migration to the cloud is 11.1.1.7 or later. The source version for side-by-side upgrade in the cloud is 12.2.1.4, as Oracle SOA Suite on Marketplace supports only 12.2.1.4.

- For SOA Cloud Service and MFT Cloud Service, the source on-premises version for migration to the cloud is 11.1.1.7 or later. The source version for side-by-side upgrade in the cloud is 12.1.3 or later.
- For SOA Cloud Service and MFT Cloud Service, it is assumed that disaster recovery is not configured for the source environment. Note that appropriate changes have to be made to the instructions if disaster recovery is configured.
- For SOA Suite on Marketplace and SOA Cloud Service, it is assumed that the production environment has a load balancer. Otherwise, the steps have to be modified and adapted accordingly.

Understand Migration and Side-by-Side Upgrade for Oracle SOA Suite on Marketplace

The tasks that you'll perform for migrating on-premises Oracle SOA Suite applications to the cloud are same as the tasks that you'll perform for side-by-side upgrade in the cloud. The approach for migration and the side-by-side upgrade is the same — you'll provision a new Oracle SOA Suite on Marketplace instance, migrate or re-create configurations from the old source environment, and then transition to the newly provisioned cloud instance.

> **Note:**
>
> Oracle SOA Suite on Marketplace allows you to create a separate instance for each service type (SOA with SB & B2B Cluster, MFT Cluster, and BAM Cluster).

Keep in mind the following differences between a migration to the cloud and a side-by-side upgrade in the cloud:

Migration	Side-by-Side Upgrade
The source version of your on-premises Oracle SOA Suite application must be 11.1.1.7 or later.	Oracle SOA Suite on Marketplace supports only 12.2.1.4. For example, you can migrate projects and artifacts from an older 12.2.1.4 instance to a newer 12.2.1.4 instance.
Your on-premises Oracle SOA Suite application may use internal Lightweight Directory Access Protocol (LDAP), third party, or Oracle IDM.	Oracle SOA Suite on Marketplace uses internal LDAP.
Your on-premises Oracle SOA Suite application may use Oracle Traffic Director (OTD), Oracle HTTP Server (OHS), or a third party load balancer.	Oracle SOA Suite on Marketplace uses the Oracle Cloud Infrastructure load balancer.
Your on-premises Oracle SOA Suite application may use Java Keystore (JKS) or Keystore Services (KSS).	Oracle SOA Suite on Marketplace uses KSS.

Migration	Side-by-Side Upgrade
You cannot copy security information directly between an on-premises Oracle SOA Suite application and Oracle SOA Suite on Marketplace during the migration. You have to first export the security information from your on-premises Oracle SOA Suite application to a local file, and then copy the file and import it to the target environment.	You can directly copy and import security information between the source and the target Oracle SOA Suite on Marketplace instances.

For the tasks you will need to complete to migrate an on-premises Oracle SOA Suite application to Oracle SOA Suite on Marketplace or perform a side-by-side upgrade, see Migration/Side-by-Side Upgrade for Oracle SOA Suite on Marketplace.

Understand Migration and Side-by-Side Upgrade for Oracle SOA Cloud Service

The tasks that you'll perform for migrating on-premises Oracle SOA Suite applications to the cloud are same as the tasks that you'll perform for side-by-side upgrade in the cloud. The approach for migration and the side-by-side upgrade is the same — you'll provision a new cloud instance of Oracle SOA Cloud Service, migrate or re-create configurations from the old source environment, and then transition to the newly provisioned cloud instance.

> **Note:**
>
> Oracle SOA Cloud Service allows you to create a separate instance for each service type (SOA with SB & B2B Cluster, MFT Cluster, and BAM Cluster).

Keep in mind the following differences between a migration to the cloud and a side-by-side upgrade in the cloud:

Oracle SOA Cloud Service

Migration	Side-by-Side Upgrade
The source version of your on-premises Oracle SOA Suite application must be 11.1.1.7 or later.	The source version of the Oracle SOA Cloud Service instance that you want to upgrade must be 12.1.3 or later.
Your on-premises Oracle SOA Suite application may use internal Lightweight Directory Access Protocol (LDAP), third party, or Oracle IDM.	Oracle SOA Cloud Service uses internal LDAP.
Your on-premises Oracle SOA Suite application may use Oracle Traffic Director (OTD), Oracle HTTP Server (OHS), or a third party load balancer.	Oracle SOA Cloud Service uses OTD.

Migration	Side-by-Side Upgrade
Your on-premises Oracle SOA Suite application may use Java Keystore (JKS) or Keystore Services (KSS).	Oracle SOA Cloud Service uses KSS.
You cannot copy security information directly between an on-premises Oracle SOA Suite application and Oracle SOA Cloud Service during the migration. You have to first export the security information from your on-premises Oracle SOA Suite application to a local file, and then copy the file and import it to the target environment.	You can directly copy and import security information between the source and the target Oracle SOA Cloud Service instances.

For the tasks you will need to complete to migrate an on-premises Oracle SOA Suite application to Oracle SOA Cloud Service or perform a side-by-side upgrade, see Migration/Side-by-Side Upgrade for SOA Cloud Service.

Understand Migration and Side-by-Side Upgrade for MFT Cloud Service

The tasks that you'll perform for migrating on-premises MFT applications to the cloud are same as the tasks that you'll perform for side-by-side upgrade in the cloud for MFT Cloud Service. The approach for migration and the side-by-side upgrade is the same — you'll provision a new cloud instance of MFT Cloud Service, migrate or recreate configurations from the old source environment and then transition to the newly provisioned cloud instance.

However, you have to keep in mind the following differences between migration to the cloud and side-by-side upgrade in the cloud:

Migration	Side-by-Side Upgrade
The source version of your on-premises MFT application must be 11.1.1.7 or later.	The source version of the cloud MFT instance that you want to upgrade must be 12.1.3 or later.
Your on-premises MFT application may use internal Lightweight Directory Access Protocol (LDAP), third party, or Oracle IDM.	MFT Cloud Service uses internal LDAP.
Your on-premises MFT application may use Oracle Traffic Director (OTD) or a third party load balancer.	MFT Cloud Service uses OTD.
You cannot copy security information directly between an on-premises MFT application and MFT Cloud Service during the migration. You have to first export the security information from your on-premises MFT application to a local file, and then copy the file and import it to the target environment.	You can directly copy and import security information between the source and the target MFT Cloud Service instances.

For the tasks you will need to complete to migrate an on-premises Oracle SOA Suite application to MFT Cloud Service or perform a side-by-side upgrade, see Migration/Side-by-Side Upgrade for MFT Cloud Service.

2

Migration/Side-by-Side Upgrade for Oracle SOA Suite on Marketplace

Learn how to migrate your on-premises SOA application to the cloud or how to do a side-by-side upgrade in the cloud for Oracle SOA Suite on Marketplace.

Topics:

Provision Oracle SOA Suite on Marketplace

Provision a new Oracle SOA Suite on Marketplace instance before starting the other migration and side-by-side upgrade related tasks. You'll migrate configurations from your old source environment into this newly provisioned instance of Oracle SOA Suite on Marketplace.

Create a simple hello world application (SOA composite/OSB proxy service/B2B agreement) and test to check that it works.

Prepare Clients for Migration/Side-by-Side Upgrade

Configure and prepare your clients such that the transition of HTTP clients from the old deployment to the new deployment is smooth and happens by switching the Domain Name System (DNS) entry.

These changes can be done gradually over time because after these changes are completed everything continues to work as before the changes. This includes some changes to the source environment.

To prepare clients:

1. Get a DNS name issued from DNS issuing authority. Point this DNS name to the source environment load balancer.

   If you are already using a DNS name in clients skip this step.

2. Create a new port in the source environment load balancer that matches the target Oracle SOA Suite on Marketplace port number. Add routing rule to this new port to route to the original load balancer port in the source environment.

   Note that on-premises SOA applications can use a different port number than the target Oracle SOA Suite on Marketplace environment. This makes it impossible to switch clients during transition from the old to the new deployment by switching the DNS.

3. Change all clients to use the DNS name and new port.

   For SSL, it might be required that the trust certificate for the target environment server has to be pre-configured at the client so that transition from the source to the target environment works smoothly.

4. If you were already set up to use a global DNS name, ensure that the Oracle WebLogic Server front end host points to the load balancer and not the DNS name.

   Ensure that the loopback HTTP invokes point to the Oracle WebLogic Server front end host/port. For SOA, also ensure that the loopback abstract WSDL/Schema references point to the WebLogic FE host/port.

   These changes ensure that:

   - Callbacks come back to the source domain that issued the request, after transitioning to the target.
   - Loopbacks in the source domain come back to the source domain after transitioning to the target.

Prepare Your Source for Migration/Side-by-Side Upgrade

You have to migrate the Integrated Development Environment (IDE) projects and export or capture needed artifacts from the source environment to prepare your source for Oracle SOA Suite on Marketplace migration/side-by-side upgrade.

To prepare your source:

1. Migrate IDE projects to the 12*c* IDE that matches the Oracle SOA Suite on Marketplace version for SOA/Oracle Service Bus.
2. Export metadata from the source environment for B2B/Oracle Service Bus, where the IDE is not used.

   For B2B, export the full repository. See "Importing and Exporting the Design-Time Repository" in *Using Oracle B2B* (12.2.1.4 | 12.2.1.3 | 12.2.1.2 | 12.1.3).

3. Grab the domain file system artifacts such as custom XPath functions, B2B Java callouts, SOA token mapping file and any script scheduled with Oracle Enterprise Scheduler (ESS) for Oracle Service Bus/SOA.
4. Ensure that there are no hard-coded URLs in the job definitions for Oracle Enterprise Scheduler. Use tokens instead.

   See "Using Token Substitution" in *Developing Applications for Oracle Enterprise Scheduler* (12.2.1.4 | 12.2.1.3 | 12.2.1.2 | 12.1.3).

5. Export the shared artifacts that are stored in MetaData Services (MDS) schemas in the source environment by using the offline WLST command:

   ```
   sca_exportSharedData(serverURL, JARfile, pattern, username,
   password)
   ```

6. Export `/oracle/apps/ess/custom namespace` in the ESS partition `essUserMetadata` for Oracle Enterprise Scheduler. Do this export using MDS export in the application essnativehostingapp in MDS.
7. Export `/oracle/as/ess/essapp/custom namespace` in the ESS partition `essapp-internal-partition` for Oracle Enterprise Scheduler. Do this export using MDS export in the application ESSAPP in MDS.
8. Note the token values in URLs for Oracle Enterprise Scheduler. You will need this later.

   Note the token values to be used for cloud, if different.
9. For SOA, if there are references to schemas and abstract WSDLs that are different in the cloud, change the source environment for composites or capture it in the configuration plan.
10. For Oracle Service Bus/SOA, adjust the customization files/configuration plans for deployment to the target instance. Change URLs to values appropriate for Oracle SOA Suite on Marketplace.
11. Use the T3 syntax `cluster:t3://clustername` for local loopback T3 references in configurations.

    Some products like Oracle Service Bus do not support this syntax. For loopback WebLogic JMS URL, use `jms://connection_factory/.....`
12. If you are migrating to an Autonomous Transaction Processing database (ATP-D), to ensure successful export of the SOAINFRA schema from your on-premises database, run the following command to unlock the schema before export:

    ```
    ALTER USER schema_name IDENTIFIED BY password ACCOUNT UNLOCK;
    ```

Prepare Your Target Environment

Prepare your target environment by importing or recreating all the configurations of your source. This will ensure successful deployment of the target Oracle SOA Suite on Marketplace instance.

To prepare your target environment:

1. Create the required WLS artifacts.

   WLS artifacts can be: Java Message Service (JMS) queue, Java EE Connector Architecture (JCA) adapter configurations, data source, work managers, J2EE app deployment, JMS servers, JMS topics, and so on.
2. Implement any security configurations.

   Security configurations can be: custom Oracle Web Service Manager (OWSM) policies, Credential Store Framework (CSF) keys, certificates, users, groups, custom Oracle Platform Security Service (OPSS) roles, custom OPSS permissions, group memberships, role memberships, enterprise roles, OPSS credentials and so on.

For information on OPSS commands to migrate keystores , see Managing Keystores with WLST in *Securing Applications with Oracle Platform Security Services*.

For information on OPSS commands to migrate credentials, see Managing Credentials with WLST in *Securing Applications with Oracle Platform Security Services*.

For information on OWSM commands to migrate custom policies, see Migrating Policies in *Administering Web Services*.

3. Test that your security configurations work.
   - Create a simple application comprising of SOA composite, OSB proxy service, B2B agreement, ESS job.
   - Ensure that the application uses at least one of the keys/certificates/credentials.
   - Test to check if the application works.
   - Check if you can view an imported user in LDAP.
4. Import shared artifacts in MetaData Services (MDS) schemas for SOA.
5. Deploy projects from the console for SOA/Oracle Service Bus.

   Use the prepared customization file/configuration plan. Ensure loopback abstract WSDL/Schema references and loopback HTTP invokes point to the target environment Load Balancer and not the DNS name.

   For inbound adapters, if the address for both deployments is the same, ensure that it doesn't start processing production messages by externally blocking it from accessing inbound endpoints. Then, if possible, you can deactivate the SOA adapter.
6. Import artifacts for B2B.

   The inbound channels are disabled by default. If required, add URLs in the console for the cloud and deploy all artifacts.
7. Import `/oracle/apps/ess/custom namespace` and `/oracle/as/ess/essapp/custom namespace` for Oracle Enterprise Scheduler.
8. Enter the token values noted earlier for Oracle Enterprise Scheduler.
9. Rebind work assignments to the cluster or managed server for Oracle Enterprise Scheduler.

   See Managing Work Assignments and Workshifts in *Administering Oracle Enterprise Scheduler*.
10. Add file system artifacts captured from the source environment, such as custom XPath functions, SOA token mapping file, B2B java callouts.
11. Test the endpoints.

    Use the endpoints in the application (SOA composite, OSB proxy service, B2B agreement, ESS job) created for testing and check if it works. After testing, change it back to the original endpoints.
12. Add scripts scheduled with Oracle Enterprise Scheduler.
13. Set your tuning settings if they are available.
14. Redo all the SOA Composer customizations manually.

15. Redo any Enterprise Manager configuration steps manually.

    For details, see Reconfigure Tuning and Configuration Parameters.

16. If Oracle SOA Suite on Marketplace is going to access endpoints on-premises then you may need VPN.

    You can setup VPN through IPSec VPN.

17. Apply UMS configuration manually to the target environment.

Test Your Target Environment

You can test your target environment at this point to check if everything is working as expected after the migration. It is assumed that you have already tested in a stage system (test environment).

To test your target environment:

1. Use endpoints to test in the configuration plans of the steps that you have completed till now.
2. Test and check if everything is working as expected.
3. Switch to production endpoints.

   This may require projects to be redeployed with appropriate configuration plans.

Transition from Old Deployment to New Deployment

After you have prepared your source and target environments for the migration, you can transition your production system from old deployment to new deployment. You can do this by transitioning: HTTP Clients, inbound adapters where address is the same for old and new, clients of inbound adapters where address is different for old and new, and clients who are reading from the old environment (such as a JMS queue) but now need to read from the target environment.

Note that the transition from old to new deployment will not work if the following are used:

- BPEL correlation sets or message ordering.
- Mid-process receives from clients in BPEL.
- If SOA composites have human workflow elements.

To transition from old to new deployment:

1. De-activate the inbound composite/adapter/channel/transport in the old deployment if the inbound address in both old and new deployment is same.

   For FTP inbound, delete any processed file left behind after processing.

2. Switch the DNS.

   The DNS switch is not instantaneous and may take a while (depending on TTL settings in routers) to propagate across the internet.

3. Enable inbound composite/adapter/channel/transport in the target environment system.

   For some inbound adapters like WLS Java Messaging Service (JMS), the address is different and clients have to change the address and switch.

4. Terminate all ESS jobs in the source environment and schedule them in the target environment.
5. Ensure that callback and loopback invokes in SOA must come to the domain that initiated it. So the old deployment continues processing callbacks/loopbacks while new requests are processed by the new deployment.

   When all callbacks/loopbacks are processed and all backlog messages are processed and there is no need for a rollback, then you can destroy the old deployment. External clients who read from, for example, local weblogic JMS queues in the source deployment will switch to the target deployment after all messages are processed.

Reconfigure Configuration Parameters and Tune in Oracle SOA Suite on Marketplace

Reconfigure any Enterprise Manager tuning and configuration parameters that you had previously set in the source environment or you need to change in the target environment.

You'll perform these steps as part of preparing your target environment for transitioning from the old to the new environment.

SOA

- Lazy loading
- Modularity profile
- Autopurge
- Timeouts (transaction, Enterprise JavaBeans, HTTP)
- Work managers
- SOA data source connection pool
- Resiliency
- In-memory
- EDN
- Instance tracking

ESS

- Dispatcher
- Processor thread pool
- Attach ESS web service OWSM policy
- Scheduled purge

OSB

- Results cache
- Work managers

B2B

- For information on Enterprise Manager Parameters, see Setting B2B Configuration Properties in Fusion Middleware Control in *Using Oracle B2B*.
- For information on B2B interface parameters, see Configuring B2B System Parameters in *Using Oracle B2B*.

Migrate Data Components

Migrate your data components such as LDAP, OPSS, OWSM, ESS, B2B, OSB and SOA from the source to the target environment.

You'll perform these tasks as part of preparing your target environment for transitioning from the old environment to the new.

Migrate your data components in the following order:

1. Migrate LDAP Data
2. Migrate OPSS Data
3. Migrate OWSM Data
4. Migrate the remaining data (ESS, B2B, OSB and SOA) in any order.

Move LDAP Data

LDAP data includes the Oracle WebLogic Server specified user, group, enterprise role and security policies (predefined Oracle WebLogic configurations and configurations that users have added to internal LDAP). Import and move the LDAP data from your source to your target environment.

For migrating LDAP data from your on-premises SOA instance to the cloud, refer to the WebLogic LDAP documentation or refer to your on-premises IDM documentation. Check if any migration is possible or you will need to manually re-enter everything.

The WebLogic console has commands to export and import internal LDAP. This can be used to move users/groups/group memberships/enterprise roles etc. By default, LDAP import will not overlay users and groups, and other artifacts that are already there. This is the desired behavior. For details, see Exporting and Importing Information in the Embedded LDAP Server in *Administering Security for Oracle WebLogic Server*.

When you export the whole LDAP, information which the integration does not use such as XACML policies and default credential mapper, also gets exported. This information may get seeded by WebLogic and exporting/importing this information can have issues. So do not export/import this information.

For information on how to handle the WebLogic OOTB security provider data migration, see:

- Security Data Migration in *Developing Security Providers for Oracle WebLogic Server*.
- Migrating Security Data in *Administering Security for Oracle WebLogic Server*.

You can navigate to any security provider that supports the migration functions and invoke the import( ) and/or export ( ) MBean operation such that this security provider's

data can be addressed outside of any other security provider data. See Migrating Data with WLST in *Administering Security for Oracle WebLogic Server*.

Here is an example with direct lookup vs navigation:

```
$ java weblogic.WSLT
% connect()
% serverConfig()
% realm = cmo.getSecurityConfiguration().getDefaultRealm()
% atn = realm.lookupAuthenticationProvider('DefaultAuthenticator')
% atn.exportData('DefaultAtn', 'myFile', None)
% disconnect()
```

You can use WLST if you decide that you need any data beyond the default Authenticator (Embedded LDAP users/groups). It is recommended that you also export roles.

Exporting LDAP Data

This is an example of commands to export LDAP data from on-premises Oracle SOA Suite 12.1.3.

Before exporting LDAP data, enter the following commands in the source 12.1.3 environment.

```
ssh -i opc_rsa opc@<source_admin_host_ip>
sudo -su oracle
cd /u01/
cd /app/oracle/middleware/oracle_common/common/bin/
./wlst.sh


connect('weblogic','welcome1','t3s://
<source_admin_host_ip>:<admin_port>')
<currentDomainName>=cmo.getName()
```

1. Export users and groups.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/AuthenticationProviders/DefaultAuthenticator')
   cmo.exportData('DefaultAtn', '<filename>', Properties())
   ```

   Example:

   ```
   cmo.exportData('DefaultAtn','/tmp/ldapdata/
   DefaultAuthenticator.dat', Properties())
   ```

2. Export security roles.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/RoleMappers/XACMLRoleMapper')
   cmo.exportData('XACML','<filename>', Properties())
   ```

3. Export credential mapper.

```
cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
'/Realms/myrealm/CredentialMappers/DefaultCredentialMapper')
cmo.exportData('DefaultCreds','<filename>', Properties())
```

4. Export XACML Authorizer.

```
cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
'/Realms/myrealm/Authorizers/XACMLAuthorizer')
cmo.exportData('XACML','<filename>', Properties())
```

   Where `<filename>` is the file path where data needs to be exported.

5. Copy the exported file to a local computer.

   Create a directory where you can copy the exported data and enter the following commands:

```
scp DefaultAuthenticator.dat <username:source_host_ip>:/
<local_export_dir_path>
scp DefaultCredentialMapper.dat <username:source_host_ip>:/
<local_export_dir_path>
scp XACMLAuthorizer.dat <username:source_host_ip>:/
<local_export_dir_path>
scp XACMLRoleMapper.dat <username:source_host_ip>:/
<local_export_dir_path>
```

6. Copy the SOA LDAP data from the 12.1.3 host to the target SOA Suite on Marketplace host. For example:

   Create a directory on the target environment.

   Go to the target directory folder where exported files should be copied and enter the following commands:

```
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
DefaultAuthenticator.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
DefaultCredentialMapper.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
XACMLAuthorizer.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
XACMLRoleMapper.dat
```

Importing LDAP Data

This is an example of commands to import LDAP data into SOA Suite on Marketplace.

Before importing LDAP data, enter the following commands in the target SOA Suite on Marketplace environment:

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2, but are also applicable to Oracle SOA Suite on Marketplace.

```
ssh -i opc_rsa opc@<target_admin_host_ip>
sudo -su oracle
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh


connect('weblogic','welcome1','t3s://
<target_admin_host_ip>:<target_host_port>')
currentDomainName=cmo.getName()
```

1. Import users and groups.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/AuthenticationProviders/DefaultAuthenticator')
   cmo.importData('DefaultAtn','<filename>', Properties())


   cmo.importData('DefaultAtn','/tmp/temp_usera/
   DefaultAuthenticator.dat', Properties())
   ```

2. Import security RoleMapper.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/RoleMappers/XACMLRoleMapper')
   cmo.importData('XACML','<filename>', Properties())
   ```

3. Import credential mapper.

   SOA does not use the WebLogic credential mapper. In general, it is recommended not to import data that SOA does not use. This is because this data may have WebLogic seeded data which might conflict with seeded data in the target environment. However it is provided for completeness.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/CredentialMappers/DefaultCredentialMapper')
   cmo.importData('DefaultCreds','<filename>', Properties())
   ```

4. Import XACML Authorizer.

   SOA does not use WebLogic XACML authorization. In general, it is recommended not to import data that SOA does not use. This is because this data may have WebLogic seeded data which might conflict with seeded data in the target environment. However it is provided for completeness.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/Authorizers/XACMLAuthorizer')
   cmo.importData('XACML','<filename>', Properties())
   ```

Where `filename` is the directory in which the imported data needs to be placed.

Move OPSS Data

Move OPSS data by exporting from the source (on-premises Oracle SOA Suite application in case of migration to the cloud and SOA Suite on Marketplace in case of side-by-side upgrade). Then copy the exported file to the newly provisioned target environment and import.

OPSS consists of the following:

- OPSS policies application roles and permissions

  These are mostly seeded automatically but in some cases customers can create their own roles and policies. Also, customers will define role memberships.

- Keys, certificates and trust certificates

  These are used for authentication, signing, encryption and SSL. Trust certificates are public certificates of certificate issuing authorities to establish the trust chain.

- Credentials

Note the following when you move OPSS data:

- Bootstrap credentials and bootstrap keys must be preserved in the target environment domain and should not be overlayed with import and export.

  If nothing was done to specifically import/export keys into the system keystore in the source system, it is recommended that you do not migrate the source system keystore since the same contents will get seeded when the destination domain is created.

- Migration of the OPSS audit service is not required.
- Server SSL key must be preserved in the target environment domain and should not be overlayed with import and export.

> **Note:**
>
> Source environment deployment server certificates with host names in the certificates cannot be reused.

OPSS Policies

Integration products typically do not permit users to create custom policies (roles and permissions) with the exception of Oracle Enterprise Scheduler (ESS). You cannot move custom policies because they are intermingled with seeded policies and OPSS does not support moving just custom policies.

For ESS, the custom policies are in the native hosting application stripe and this stripe is empty out of the box. So, it is possible to move these policies intact. For details on how to do this, see Migrating Policies with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.

Keystores

OPSS supports two types of keystores: JKS (old) and KSS. By default, 12c uses KSS.

If the keys and certificates are in JKS, there is no OPSS involvement. You can simply copy the file to the other domain and move it to KSS.

There are two options available for KSS migration:

- Use the command `MigrateSecurityStore` to migrate across domains. This means that for side-by-side upgrade (cloud to cloud), it may be possible to migrate directly, that is, database to database across domains.
- Use the command `MigrateSecurityStore` to migrate in a two-phased approach: migrate from KSS to file, copy over the file, and then migrate back to KSS.

For information on the commands and examples to migrate keys to file or across domains, see Migrating Keys and Certificates across Different Domains in *Securing Applications with Oracle Platform Security Services*. From a migration perspective, LDAP or DB should not make a difference. The steps to migrate are the same.

For information on importing and exporting individual keys in the keystores, see Tasks 4 and 5 of Managing Keystores with WLST in *Securing Applications with Oracle Platform Security Services*. You can use this in the target environment to export the server SSL key before the move and import it back after the move.

Server certificates typically have DNS names in them and are not usable in the target environment.

Credentials

You can use the `migrateSecurityStore` command to migrate credentials.

There are two variants of the `migrateSecurityStore` command:

- You can use it to migrate all the credentials in the DB store. See Migrating All Credentials with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.
- You can migrate only the specific map that you're interested in. See Migrating One Credential Map with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.

You'll need to invoke migration in two phases:

- Run the migration on the first domain to migrate credentials (all or a specific map) from DB to file. This will create a file `cwallet.sso`.
- Carry over the file `cwallet.sso` to the second domain, and run the migration to migrate credentials from file `cwallet.sso` (carried over from phase 1) to DB in the second domain.

The export is done in the source environment domain. The exported file is carried over and then the import is done in the target environment domain.

The OWSM bootstrap CSF key in KSS should not be overlaid. The bootstrap CSF key entry is specified in `wsm-config.xml` under `$DOMAIN_CONFIG_DIR/fmwconfig`.

Steps for Migrating OPSS Data

This is an example of commands to migrate OPSS data from on-premises Oracle SOA Suite 12.1.3 to SOA Suite on Marketplace.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to Oracle SOA Suite on Marketplace.

1. Download the `opc_rsa` file to the on-premises Oracle SOA Suite 12.1.3 host.

   Example commands:

   ```
   scp -i opc_rsa opc_rsa opc@10.252.159.68:/tm
   ssh -i opc_rsa opc@10.252.159.68
   ```

2. Export preparation.

   Example commands:

   ```
   mkdir -p /u01/data/opss/export/

   cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/system-jazn-
   data.xml
   /u01/data/opss/export

   cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/keystores.xml
   /u01/data/opss/export

   cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/cwallet.sso
   /u01/data/opss/export/bootstrap_cwallet.sso

   cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/jps-config-
   jse.xml
   /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/export-jps-
   config.xml
   ```

   Add configurations into `export-jps-config.xml`.

   Example:

   ```
   <serviceInstance name="policystore.dest"
   provider="policystore.xml.provider"
   location="/u01/data/opss/export/system-jazn-data.xml">
   <description>File Based Policy Store Service Instance</description>
   </serviceInstance>

   <serviceInstance name="credstore.dest" provider="credstoressp"
   location="/u01/data/opss/export">
   <description>File Based Credential Store Service Instance</
   ```

```
description>
</serviceInstance>

<serviceInstance name="keystore.dest" provider="keystore.provider">
<description>Default JPS Keystore Service</description>
<property name="keystore.file.path" value="/u01/data/opss/export"/>
</serviceInstance>

<jpsContext name="dest1213">
<serviceInstanceRef ref="policystore.dest"/>
<serviceInstanceRef ref="pdp.service"/>
<serviceInstanceRef ref="credstore.dest"/>
<serviceInstanceRef ref="keystore.dest"/>
</jpsContext>
```

3. Export OPSS data.

   Example commands:

```
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh

migrateSecurityStore(type="credStore",
configFile="/u01/data/domains/SOAOSBLS_domain/config/fmwconfig/
export-jps-config.xml",
src="default", dst="dest1213")

migrateSecurityStore(type="appPolicies", src="default",
dst="dest1213",
configFile="/u01/data/domains/SOAOSBLS_domain/config/fmwconfig/
export-jps-config.xml",
srcApp="EssNativeHostingApp", overWrite="true")

migrateSecurityStore(type="keyStore",configFile="/u01/data/domains/
SOAOSBLS_domain/config/
fmwconfig/export-jps-config.xml", src="default",dst="dest1213")
```

4. Copy SOA, OSB, and ESS OPSS data from the on-premises Oracle SOA Suite 12.1.3 host to the SOA Suite on Marketplace host.

   Example commands:

```
scp -i /u01/data/opc_rsa /u01/data/opss/export/system-jazn-data.xml
opc@10.252.157.25:/tmp

scp -i /u01/data/opc_rsa /u01/data/opss/export/keystores.xml
opc@10.252.157.25:/tmp

scp -i /u01/data/opc_rsa /u01/data/opss/export/cwallet.sso
opc@10.252.157.25:/tmp

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/bootstrap/
cwallet.sso /u01/data/opss/export/bootstrap_cwallet.sso
```

```
scp -i /u01/data/opc_rsa /u01/data/opss/export/
bootstrap_cwallet.sso opc@10.252.157.25:/tmp
```

5. Save SOA, OSB, and ESS OPSS data.

   Example commands:

   ```
   ssh -i opc_rsa opc@10.252.157.25
   chmod a+r /tmp/system-jazn-data.xml /tmp/keystores.xml /tmp/
   cwallet.sso
   /tmp/bootstrap_cwallet.sso
   ```

6. Migrate preparation.

   Example commands:

   ```
   mkdir -p /u01/data/opss/bootstrap

   cd /u01/data/opss/

   cp /tmp/system-jazn-data.xml /tmp/keystores.xml /tmp/
   cwallet.sso /u01/data/opss

   cp /u01/data/domains/SOAOSBta_domain/config/fmwconfig/jps-config-
   jse.xml
   /u01/data/domains/SOAOSBta_domain/config/fmwconfig/migrate-jps-
   config.xml
   ```

   Add configuration into `migrate-jps-config.xml`.

   Example:

   ```
   <serviceInstance name="policystore.src"
   provider="policystore.xml.provider"
   location="/u01/data/opss/system-jazn-data.xml">
   <description>File Based Policy Store Service Instance</description>
   </serviceInstance>

   <serviceInstance name="credstore.src" provider="credstoressp"
   location="/u01/data/opss">
   <description>File Based Credential Store Service Instance</
   description>
   </serviceInstance>

   <serviceInstance name="keystore.src" provider="keystore.provider">
   <description>Default JPS Keystore Service</description>
   <property name="keystore.file.path" value="/u01/data/opss"/>
   </serviceInstance>

   <jpsContext name="soamp">
   <serviceInstanceRef ref="policystore.src"/>
   <serviceInstanceRef ref="pdp.service"/>
   <serviceInstanceRef ref="credstore.src"/>
   </jpsContext>
   ```

```
<jpsContext name="soampkss">
<serviceInstanceRef ref="keystore.src"/>
</jpsContext>
```

7. Migrate OPSS data to SOA Suite on Marketplace.

   Example commands:

   ```
   cd /u01/app/oracle/middleware/oracle_common/common/bin/
   ./wlst.sh

   migrateSecurityStore(type="credStore",
   configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
   migrate-jps-config.xml", src="soamp",
   dst="default")

   migrateSecurityStore(type="appPolicies", src="soamp", dst="default",
   configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
   migrate-jps-config.xml",
   srcApp="EssNativeHostingApp", overWrite="true")
   ```

   If SOA, OSB, and ESS have keystores other then system keystore in the customer test environment then run the following example command to migrate OPSS and keystore data.

   ```
   //migrateSecurityStore(type="keyStore", srcConfigFile="/u01/data/
   opss/migrate-jps-config.xml",
   configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
   migrate-jps-config.xml",
   src="soampkss", dst="default", srcStripe="<SOURCE_STRIPE>")
   ```

Move OWSM Data

Move OWSM data by exporting it from the source and importing it to the target environment.

OWSM has the following artifacts of interest:

- CSF keys: There are references to CSF keys in OWSM policies/policy overrides. There is no change required as long as actual values are available in the credential store owned by OPSS. CSF keys must be available in the target environment.
- certs and keys: OWSM supports two types of keystores: JKS (file based) and KSS (owned by OPSS). The certificates/aliases in the source environment should be made available in the target environment. There are references to keys/certificates in OWSM policies/policy overrides.
- Custom OWSM authorization policies: These are same as custom policies.
- Custom OWSM policies

  See Exporting Documents from the Repository Using WLST in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  See Importing Documents into the Repository Using WLST in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Additional configurations that may be required: trust config and OAuth config

In 12c, `exportWSMRepository` exports all custom policies from the repository, the trust configuration, OAuth configuration, and any other configuration documents.

In 11g, the specific custom policies have to be enumerated to export them. Note that it may not be as simple as moving the documents from 11g to 12c because as part of upgrade, the OWSM upgrade plugin takes care of adding/updating 12c specific changes in the artifacts. There may be no easy way to automate policy movement from 11g to 12c as part of migration.

Steps for Migrating OWSM Data

This is an example of commands to migrate OWSM data from on-premises Oracle SOA Suite 12.1.3 to SOA Suite on Marketplace.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to Oracle SOA Suite on Marketplace.

When migrating OWSM data, note that only custom policies and policy sets are migrated.

1. Export OWSM data.

   On the source Oracle SOA Suite 12.1.3 environment, enter the following commands before exporting OWSM data:

   ```
   ssh -i opc_rsa opc@<source_admin_host_ip>
   sudo -su oracle
   cd /u01/
   cd /app/oracle/middleware/oracle_common/common/bin/
   ./wlst.sh
   ```

   Then:

   ```
   connect('weblogic','welcome1','t3s://
   <source_admin_host_ip>:<admin_port>')
   wls:/SOAOSB12_domain/serverConfig> exportWSMRepository('/tmp/
   policies.zip',[''])
   ```

   This will export only the custom/cloned policies and custom policy sets into the `policies.zip` file under the `/tmp` folder.

2. Copy the exported file to a local computer.

   Go to the directory where the exported data is saved, then move the file from the source to the local computer.

   ```
   scp policies.zip <username:source_admin_host_ip>:/
   <local_export_dir_path>
   ```

Example:

```
scp policies.ip asmathur@adc01jjt.us.oracle.com:/scratch/exp_dat
```

3. Copy OWSM data from the Oracle SOA Suite 12.1.3 host to the SOA Suite on Marketplace host.

   Create a folder on the target environment. Change to the target directory folder where exported files should be copied.

   ```
   scp <username:TARGET_SOAMP_HOST_IP>:/<local_export_dir>/policies.zip
   ```

   Example:

   ```
   scp asmathur@adc01jjt.us.oracle.com:/scratch/export_data/
   policies.zip
   ```

4. Import OWSM data.

   On the target SOA Suite on Marketplace environment, enter the following commands to import OWSM data:

   ```
   ssh -i opc_rsa opc@host_adminip_target
   sudo -su oracle
   cd /u01/app/oracle/middleware/oracle_common/common/bin/
   ./wlst.sh


   connect('weblogic','welcome1','t3s://
   <target_host_ip>:<target_host_port>')
   wls:/SOAB2B12_domain/serverConfig> importWSMArchive("/tmp/
   policies.zip")
   ```

   This will import all the custom policies and the global policy sets exported in the `policies.zip` file into the target SOA Suite on Marketplace server.

Move ESS Metadata

Since we need to export tip versions of metadata in MDS in a specific package, we can use the `exportMetadata` WLST command with docs parameter as `"/oracle/apps/ess/custom/**"` and `"/oracle/as/ess/essapp/custom/**"` to an archive. Then we can import from the archive to the target MDS repository using the `importMetadata` WLST command.

To ensure the metadata is independent of environment, we need to tokenize URLs in job definitions first. Users have to define the new token values in the target environment (if required).

For MDS `importMetadata` and `exportMetadata` commands, see exportMetadata and importMetadata in *WLST Command Reference for Infrastructure Components*.

Steps for Migrating ESS Metadata

This is an example of commands to migrate ESS metadata from Oracle SOA Suite 12.1.3 to SOA Suite on Marketplace.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2, but are also applicable to Oracle SOA Suite on Marketplace.

1. On the source Oracle SOA Suite 12.1.3 environment, enter the following commands:

```
exportMetadata(application='EssNativeHostingApp',server='SOAOSBLS_se
rver_1',toLocation='/u01/data/artifacts
/Custom.mar',docs='/oracle/apps/ess/custom/**')


exportMetadata(application='ESSAPP',
server='SOAOSBLS_server_1',toLocation='/tmp/
ESSAPP_custom.mar',docs='/oracle/as/ess/essapp/custom/**')
```

2. On the target SOA Suite on Marketplace environment, enter the following commands:

```
importMetadata(application='EssNativeHostingApp',server='SOAOSBta_se
rver_1',fromLocation='/u01/data/artifacts
/Custom.mar',docs='/oracle/apps/ess/custom/**')


importMetadata(application='ESSAPP',
server='SOAOSBta_server_1',fromLocation='/tmp/
ESSAPP_custom.mar',docs='/oracle/as/ess/essapp/custom/**')
```

Move B2B Metadata

Move B2B metadata from your source to your target environment.

For detail instructions, see Importing and Exporting Data in *Using Oracle B2B*.

Move Oracle Service Bus Projects

The easiest way to export and import Oracle Service Bus metadata is through the console. You can export all the projects with one export.

See How to Export Resources to a Configuration JAR File in the Console in *Developing Services with Oracle Service Bus*.

Move SOA Projects

The SOA composite SAR archive can be generated easily in JDeveloper by generating a SAR archive (instead of deploying to the server). This can be deployed to the target Oracle SOA Suite on Marketplace server from the console, ant script, or WLST script.

See Deploying SOA Composite Application in Oracle JDeveloper in *Developing SOA Applications with Oracle SOA Suite*.

Transition Inbound Adapters/Transports

For successful migration/side-by-side upgrade, you need to transition inbound adapters/transports.

There are two use cases to consider for transitioning inbound adapters/transports. During transition, you disable the inbound adapters/transport at the source and enable it on the target environment. Also, when you first deploy the projects to the target environment, you do not want inbound adapters/transports to process production messages right away until you are ready for the transition. To solve both the use cases, you can do any of the following:

- Change the etc/host file or add/remove permissions for the file directory.
- Change to composite or adapter activate/deactivate.

  SOA supports adapter activate/deactivate only in 12.1.3. In B2B, the inbound channel is disabled by default on import. Oracle Service Bus does not support this.
- Change the inbound endpoints to test or true endpoints.

  This requires a redeployment.

3

Migration/Side-by-Side Upgrade for SOA Cloud Service

Learn how to migrate your on-premises SOA application to the cloud or how to do a side-by-side upgrade in the cloud for SOA Cloud Service.

Topics:

Provision SOA Cloud Service

Provision a new SOA Cloud Service instance before starting the other migration and side-by-side upgrade related tasks. You'll migrate or recreate configurations from your old source environment into this newly provisioned instance of SOA Cloud Service.

Create a simple hello world application (SOA composite/OSB proxy service/B2B agreement) and test to check that it works.

Prepare Clients for Migration/Side-by-Side Upgrade

Configure and prepare your clients such that the transition of HTTP clients from the old deployment to the new deployment is smooth and happens by switching the Domain Name System (DNS) entry.

These changes can be done gradually over time because after these changes are completed everything continues to work as before the changes. This includes some changes to the source environment.

To prepare clients:

1. Get a DNS name issued from DNS issuing authority. Point this DNS name to the source environment load balancer.

   If you are already using a DNS name in clients skip this step.

2. Create a new port in the source environment load balancer that matches the target SOA Cloud Service port number. Add routing rule to this new port to route to the original load balancer port in the source environment.

   Note that on-premises SOA applications can use a different port number than the target SOA Cloud Service environment. This makes it impossible to switch clients during transition from the old to the new deployment by switching the DNS.

3. Change all clients to use the DNS name and new port.

   For SSL, it might be required that the trust certificate for the target environment server has to be pre-configured at the client so that transition from the source to the target environment works smoothly.

4. If you were already set up to use a global DNS name, ensure that the Oracle WebLogic Server front end host points to the load balancer and not the DNS name.

   Ensure that the loopback HTTP invokes point to the Oracle WebLogic Server front end host/port. For SOA, also ensure that the loopback abstract WSDL/Schema references point to the WebLogic FE host/port.

   These changes ensure that:

   - Callbacks come back to the source domain that issued the request, after transitioning to the target.
   - Loopbacks in the source domain come back to the source domain after transitioning to the target.

Prepare Your Source for Migration/Side-by-Side Upgrade

You have to migrate the Integrated Development Environment (IDE) projects and export or capture needed artifacts from the source environment to prepare your source for Oracle SOA Cloud Service migration/side-by-side upgrade.

To prepare your source:

1. Migrate IDE projects to the 12*c* IDE that matches the Oracle SOA Cloud Service version for SOA/Oracle Service Bus.
2. Export metadata from the source environment for B2B/Oracle Service Bus, where the IDE is not used.

   For B2B, export the full repository. See "Importing and Exporting the Design-Time Repository" in *Using Oracle B2B* (12.2.1.4 | 12.2.1.3 | 12.2.1.2 | 12.1.3).

3. Grab the domain file system artifacts such as custom XPath functions, B2B Java callouts, SOA token mapping file and any script scheduled with Oracle Enterprise Scheduler (ESS) for Oracle Service Bus/SOA.
4. Ensure that there are no hard-coded URLs in the job definitions for Oracle Enterprise Scheduler. Use tokens instead.

   See "Using Token Substitution" in *Developing Applications for Oracle Enterprise Scheduler* (12.2.1.4 | 12.2.1.3 | 12.2.1.2 | 12.1.3).

5. Export the shared artifacts that are stored in MetaData Services (MDS) schemas in the source environment by using the offline WLST command:

   ```
   sca_exportSharedData(serverURL, JARfile, pattern, username,
   password)
   ```

6. Export `/oracle/apps/ess/custom namespace` in the ESS partition `essUserMetadata` for Oracle Enterprise Scheduler. Do this export using MDS export in the application essnativehostingapp in MDS.
7. Export `/oracle/as/ess/essapp/custom namespace` in the ESS partition `essapp-internal-partition` for Oracle Enterprise Scheduler. Do this export using MDS export in the application ESSAPP in MDS.
8. Note the token values in URLs for Oracle Enterprise Scheduler. You will need this later.

   Note the token values to be used for cloud, if different.
9. For SOA, if there are references to schemas and abstract WSDLs that are different in the cloud, change the source environment for composites or capture it in the configuration plan.
10. For Oracle Service Bus/SOA, adjust the customization files/configuration plans for deployment to the target instance. Change URLs to values appropriate for Oracle SOA Cloud Service.
11. Use the T3 syntax `cluster:t3://clustername` for local loopback T3 references in configurations.

    Some products like Oracle Service Bus do not support this syntax. For loopback WebLogic JMS URL, use `jms://connection_factory/.....`
12. If you are migrating to an Autonomous Transaction Processing database (ATP-D), to ensure successful export of the SOAINFRA schema from your on-premises database, run the following command to unlock the schema before export:

    ```
    ALTER USER schema_name IDENTIFIED BY password ACCOUNT UNLOCK;
    ```

Prepare Your Target Environment

Prepare your target environment by importing or recreating all the configurations of your source. This will ensure successful deployment of the target instance.

Test Your Target Environment

You can test your target environment at this point to check if everything is working as expected after the migration. It is assumed that you have already tested in a stage system (test environment).

To test your target environment:

1. Use endpoints to test in the configuration plans of the steps that you have completed till now.
2. Test and check if everything is working as expected.
3. Switch to production endpoints.

This may require projects to be redeployed with appropriate configuration plans.

Transition from Old Deployment to New Deployment

After you have prepared your source and target environments for the migration/side-by-side upgrade, you can transition your production system from old deployment to new deployment. You can do this by transitioning: HTTP Clients, inbound adapters where address is the same for old and new, clients of inbound adapters where address is different for old and new, and clients who are reading from the old environment (such as a jms queue) but now need to read from the target environment.

Note that the transition from old to new deployment will not work if the following are used:

- BPEL correlation sets or message ordering.
- Mid-process receives from clients in BPEL.
- If SOA composites have human workflow elements.

To transition from old to new deployment:

1. De-activate the inbound composite/adapter/channel/transport in the old deployment if the inbound address in both old and new deployment is same.

   For FTP inbound, delete any processed file left behind after processing.

2. Switch the DNS.

   The DNS switch is not instantaneous and may take a while (depending on TTL settings in routers) to propagate across the internet.

3. Enable inbound composite/adapter/channel/transport in the target environment system.

   For some inbound adapters like WLS Java Messaging Service (JMS), the address is different and clients have to change the address and switch.

4. Terminate all ESS jobs in the source environment and schedule them in the target environment.

5. Ensure that callback and loopback invokes in SOA must come to the domain that initiated it. So the old deployment continues processing callbacks/loopbacks while new requests are processed by the new deployment.

   When all callbacks/loopbacks are processed and all backlog messages are processed and there is no need for a rollback, then you can destroy the old deployment. External clients who read from, for example, local weblogic JMS queues in the source deployment will switch to the target deployment after all messages are processed.

Reconfigure Tuning and Configuration Parameters

Reconfigure any Enterprise Manager tuning and configuration parameters that you had previously set in the source environment or you need to change in the target environment.

SOA

- Lazy loading

- Modularity profile
- Autopurge
- Timeouts (transaction, Enterprise JavaBeans, HTTP)
- Work managers
- SOA data source connection pool
- Resiliency
- In-memory
- EDN
- Instance tracking

ESS

- Dispatcher
- Processor thread pool
- Attach ESS web service OWSM policy
- Scheduled purge

Oracle Service Bus

- Results cache
- Work managers

B2B

Refer to the following topics in *Using Oracle B2B*:

- For information about Enterprise Manager Parameters, see Setting B2B Configuration Properties in Fusion Middleware Control.
- For information about B2B interface parameters, see Configuring B2B System Parameters.

Migrate Data Components

Migrate your data components such as LDAP, OPSS, OWSM, ESS, B2B, OSB and SOA from the source to the target environment.

You'll perform these tasks as part of preparing your target environment for transitioning from the old environment to the new.

> **Note:**
>
> The side-by-side upgrade steps described here are for SOA Could Service 12.1.3 as source environment and SOA Cloud Service 12.2.1.2 as target environment. All the steps described here are certified only for target SOA Cloud Service deployment of 12.2.1.2.

Migrate your data components in the following order:

1. Migrate LDAP Data
2. Migrate OPSS Data
3. Migrate OWSM Data
4. Migrate the remaining data (ESS, B2B, OSB and SOA) in any order.

Move LDAP Data

LDAP data includes the Oracle WebLogic Server specified user, group, enterprise role and security policies (predefined Oracle WebLogic configurations and configurations that users have added to internal LDAP). Import and move the LDAP data from your source to your target environment.

For migrating LDAP data from your on-premises SOA instance to the cloud, refer to the WebLogic LDAP documentation or refer to your on-premises IDM documentation. Check if any migration is possible or you will need to manually re-enter everything.

If you are doing a side-by-side upgrade in the cloud, note that SOA Cloud Service uses internal LDAP.

The WebLogic console has commands to export and import internal LDAP. This can be used to move users/groups/group memberships/enterprise roles etc. By default, LDAP import will not overlay users and groups, and other artifacts that are already there. This is the desired behavior. For details, see Exporting and Importing Information in the Embedded LDAP Server in *Administering Security for Oracle WebLogic Server*.

When you export the whole LDAP, information which the integration does not use such as XACML policies and default credential mapper, also gets exported. This information may get seeded by WebLogic and exporting/importing this information can have issues. So do not export/import this information.

For information on how to handle the WebLogic OOTB security provider data migration, see:

- Security Data Migration in *Developing Security Providers for Oracle WebLogic Server*.
- Migrating Security Data in *Administering Security for Oracle WebLogic Server*.

You can navigate to any security provider that supports the migration functions and invoke the import( ) and/or export ( ) MBean operation such that this security provider's data can be addressed outside of any other security provider data. See Migrating Data with WLST in *Administering Security for Oracle WebLogic Server*.

Here is an example with direct lookup vs navigation:

```
$ java weblogic.WSLT
% connect()
% serverConfig()
% realm = cmo.getSecurityConfiguration().getDefaultRealm()
% atn = realm.lookupAuthenticationProvider('DefaultAuthenticator')
% atn.exportData('DefaultAtn', 'myFile', None)
% disconnect()
```

You can use WLST if you decide that you need any data beyond the default Authenticator (Embedded LDAP users/groups). It is recommended that you also export roles.

Exporting LDAP Data

This is an example of commands to export LDAP data from SOA Cloud Service 12.1.3.

Before exporting LDAP data, enter the following commands in the source SOA Cloud Service 12.1.3 environment.

```
ssh -i opc_rsa opc@<source_admin_host_ip>
sudo -su oracle
cd /u01/
cd /app/oracle/middleware/oracle_common/common/bin/
./wlst.sh


connect('weblogic','welcome1','t3s://
<source_admin_host_ip>:<admin_port>')
<currentDomainName>=cmo.getName()
```

1. Export users and groups.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/AuthenticationProviders/DefaultAuthenticator')
   cmo.exportData('DefaultAtn', '<filename>', Properties())
   ```

   Example: `cmo.exportData('DefaultAtn','/tmp/ldapdata/DefaultAuthenticator.dat', Properties())`

2. Export security roles.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/RoleMappers/XACMLRoleMapper')
   cmo.exportData('XACML','<filename>', Properties())
   ```

3. Export credential mapper.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/CredentialMappers/DefaultCredentialMapper')
   cmo.exportData('DefaultCreds','<filename>', Properties())
   ```

4. Export XACML Authorizer.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/Authorizers/XACMLAuthorizer')
   cmo.exportData('XACML','<filename>', Properties())
   ```

   Where `filename` is the file path where data needs to be exported.

5. Copy the exported file to a local computer.

Create a directory where you can copy the exported data and enter the following commands:

```
scp DefaultAuthenticator.dat <username:source_host_ip>:/
<local_export_dir_path>
scp DefaultCredentialMapper.dat <username:source_host_ip>:/
<local_export_dir_path>
scp XACMLAuthorizer.dat <username:source_host_ip>:/
<local_export_dir_path>
scp XACMLRoleMapper.dat <username:source_host_ip>:/
<local_export_dir_path>
```

6. Copy the SOA LDAP data from SOA Cloud Service 12.1.3 host to the target SOA Cloud Service host (for example, SOA Cloud Service 12.2.1.2).

   Create a directory on the target environment.

   Go to the target directory folder where exported files should be copied and enter the following commands:

```
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
DefaultAuthenticator.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
DefaultCredentialMapper.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
XACMLAuthorizer.dat
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/
XACMLRoleMapper.dat
```

Importing LDAP Data

This is an example of commands to import LDAP data into SOA Cloud Service 12.2.1.2.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to 12.2.1.3 and 12.2.1.4.

Before importing LDAP data, enter the following commands in the target SOA Cloud Service 12.2.1.2 environment:

```
ssh -i opc_rsa opc@<host_adminip_target>
sudo -su oracle
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh


connect('weblogic','welcome1','t3s://
<target_host_ip>:<target_host_port>')
currentDomainName=cmo.getName()
```

1. Import users and groups.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/AuthenticationProviders/DefaultAuthenticator')
   cmo.importData('DefaultAtn','<filename>', Properties())
   ```

   Example:

   ```
   cmo.importData('DefaultAtn','/tmp/temp_usera/
   DefaultAuthenticator.dat', Properties())
   ```

2. Import security RoleMapper.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/RoleMappers/XACMLRoleMapper')
   cmo.importData('XACML','<filename>', Properties())
   ```

3. Import credential mapper.

   SOA does not use the WebLogic credential mapper. In general, it is recommended not to import data that SOA does not use. This is because this data may have WebLogic seeded data which might conflict with seeded data in the target environment. However it is provided for completeness.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/CredentialMappers/DefaultCredentialMapper')
   cmo.importData('DefaultCreds','<filename>', Properties())
   ```

4. Import XACML Authorizer.

   SOA does not use WebLogic XACML authorization. In general, it is recommended not to import data that SOA does not use. This is because this data may have WebLogic seeded data which might conflict with seeded data in the target environment. However it is provided for completeness.

   ```
   cd('serverConfig:/SecurityConfiguration/' + <currentDomainName> +
   '/Realms/myrealm/Authorizers/XACMLAuthorizer')
   cmo.importData('XACML','<filename>', Properties())
   ```

   Where `<filename>` is the directory in which the imported data needs to be placed.

Move OPSS Data

Move OPSS data by exporting from the source (on-premises SOA application in case of migration to the cloud and SOA Cloud Service in case of side-by-side upgrade). Then copy the exported file to the newly provisioned target environment and import.

OPSS consists of the following:

- OPSS policies application roles and permissions

  These are mostly seeded automatically but in some cases customers can create their own roles and policies. Also, customers will define role memberships.

- Keys, certificates and trust certificates

These are used for authentication, signing, encryption and SSL. Trust certificates are public certificates of certificate issuing authorities to establish the trust chain.

- Credentials

Note the following when you move OPSS data:

- Bootstrap credentials and bootstrap keys must be preserved in the target environment domain and should not be overlayed with import and export.

  If nothing was done to specifically import/export keys into the system keystore in the source system, it is recommended that you do not migrate the source system keystore since the same contents will get seeded when the destination domain is created.
- Migration of the OPSS audit service is not required.
- Server SSL key must be preserved in the target environment domain and should not be overlayed with import and export.

> **Note:**
>
> Source environment deployment server certificates with host names in the certificates cannot be reused.

OPSS Policies

Integration products typically do not permit users to create custom policies (roles and permissions) with the exception of Oracle Enterprise Scheduler (ESS). You cannot move custom policies because they are intermingled with seeded policies and OPSS does not support moving just custom policies.

For ESS, the custom policies are in the native hosting application stripe and this stripe is empty out of the box. So, it is possible to move these policies intact. For details on how to do this, see Migrating Policies with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.

Keystores

OPSS supports two types of keystores: JKS (old) and KSS. By default, 12c uses KSS.

If the keys and certificates are in JKS, there is no OPSS involvement. You can simply copy the file to the other domain and move it to KSS.

There are two options available for KSS migration:

- Use the command `MigrateSecurityStore` to migrate across domains. This means that for side-by-side upgrade (cloud to cloud), it may be possible to migrate directly, that is, database to database across domains.
- Use the command `MigrateSecurityStore` to migrate in a two-phased approach: migrate from KSS to file, copy over the file, and then migrate back to KSS.

For information on the commands and examples to migrate keys to file or across domains, see Migrating Keys and Certificates across Different Domains in *Securing Applications with Oracle Platform Security Services*. From a migration perspective, LDAP or DB should not make a difference. The steps to migrate are the same.

For information on importing and exporting individual keys in the keystores, see Tasks 4 and 5 of Managing Keystores with WLST in *Securing Applications with Oracle Platform Security Services*. You can use this in the target environment to export the server SSL key before the move and import it back after the move.

Server certificates typically have DNS names in them and are not usable in the target environment.

Credentials

You can use the `migrateSecurityStore` command to migrate credentials.

There are two variants of the `migrateSecurityStore` command:

- You can use it to migrate all the credentials in the DB store. See Migrating All Credentials with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.
- You can migrate only the specific map that you're interested in. See Migrating One Credential Map with migrateSecurityStore in *Securing Applications with Oracle Platform Security Services*.

You'll need to invoke migration in two phases:

- Run the migration on the first domain to migrate credentials (all or a specific map) from DB to file. This will create a file `cwallet.sso`.
- Carry over the file `cwallet.sso` to the second domain, and run the migration to migrate credentials from file `cwallet.sso` (carried over from phase 1) to DB in the second domain.

The export is done in the source environment domain. The exported file is carried over and then the import is done in the target environment domain.

The OWSM bootstrap CSF key in KSS should not be overlaid. The bootstrap CSF key entry is specified in `wsm-config.xml` under `$DOMAIN_CONFIG_DIR/fmwconfig`.

Steps for Migrating OPSS Data

This is an example of commands to migrate OPSS data from SOA Cloud Service 12.1.3 to SOA Cloud Service 12.2.1.2.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to 12.2.1.3 and 12.2.1.4.

1. Download `opc_rsa` file to the SOA Cloud Service 12.1.3 host.

   Example commands:

   ```
   scp -i opc_rsa opc_rsa opc@10.252.159.68:/tm
   ssh -i opc_rsa opc@10.252.159.68
   ```

2. Export preparation.

Example commands:

```
mkdir -p /u01/data/opss/export/

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/system-jazn-
data.xml
/u01/data/opss/export

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/keystores.xml
/u01/data/opss/export

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/cwallet.sso
/u01/data/opss/export/bootstrap_cwallet.sso

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/jps-config-
jse.xml
/u01/data/domains/SOAOSBLS_domain/config/fmwconfig/export-jps-
config.xml
```

Add configurations into `export-jps-config.xml`.

Example:

```
<serviceInstance name="policystore.dest"
provider="policystore.xml.provider"
location="/u01/data/opss/export/system-jazn-data.xml">
<description>File Based Policy Store Service Instance</description>
</serviceInstance>

<serviceInstance name="credstore.dest" provider="credstoressp"
location="/u01/data/opss/export">
<description>File Based Credential Store Service Instance</
description>
</serviceInstance>

<serviceInstance name="keystore.dest" provider="keystore.provider">
<description>Default JPS Keystore Service</description>
<property name="keystore.file.path" value="/u01/data/opss/export"/>
</serviceInstance>

<jpsContext name="dest1213">
<serviceInstanceRef ref="policystore.dest"/>
<serviceInstanceRef ref="pdp.service"/>
<serviceInstanceRef ref="credstore.dest"/>
<serviceInstanceRef ref="keystore.dest"/>
</jpsContext>
```

3. Export OPSS data.

   Example commands:

```
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh

migrateSecurityStore(type="credStore",
```

```
configFile="/u01/data/domains/SOAOSBLS_domain/config/fmwconfig/
export-jps-config.xml",
src="default", dst="dest1213")

migrateSecurityStore(type="appPolicies", src="default",
dst="dest1213",
configFile="/u01/data/domains/SOAOSBLS_domain/config/fmwconfig/
export-jps-config.xml",
srcApp="EssNativeHostingApp", overWrite="true")

migrateSecurityStore(type="keyStore",configFile="/u01/data/domains/
SOAOSBLS_domain/config/
fmwconfig/export-jps-config.xml", src="default",dst="dest1213")
```

4. Copy SOA, OSB, and ESS OPSS data from SOA Cloud Service 12.1.3 host to SOA Cloud Service 12.2.1.2 host.

   Example commands:

```
scp -i /u01/data/opc_rsa /u01/data/opss/export/system-jazn-data.xml
opc@10.252.157.25:/tmp

scp -i /u01/data/opc_rsa /u01/data/opss/export/keystores.xml
opc@10.252.157.25:/tmp

scp -i /u01/data/opc_rsa /u01/data/opss/export/cwallet.sso
opc@10.252.157.25:/tmp

cp /u01/data/domains/SOAOSBLS_domain/config/fmwconfig/bootstrap/
cwallet.sso /u01/data/opss/export/bootstrap_cwallet.sso

scp -i /u01/data/opc_rsa /u01/data/opss/export/
bootstrap_cwallet.sso opc@10.252.157.25:/tmp
```

5. Save SOA, OSB, and ESS OPSS data.

   Example commands:

```
ssh -i opc_rsa opc@10.252.157.25
chmod a+r /tmp/system-jazn-data.xml /tmp/keystores.xml /tmp/
cwallet.sso
/tmp/bootstrap_cwallet.sso
```

6. Migrate preparation.

   Example commands:

```
mkdir -p /u01/data/opss/bootstrap

cd /u01/data/opss/

cp /tmp/system-jazn-data.xml /tmp/keystores.xml /tmp/
cwallet.sso /u01/data/opss

cp /u01/data/domains/SOAOSBta_domain/config/fmwconfig/jps-config-
jse.xml
```

```
/u01/data/domains/SOAOSBta_domain/config/fmwconfig/migrate-jps-
config.xml
```

Add configuration into `migrate-jps-config.xml`.

Example:

```
<serviceInstance name="policystore.src"
provider="policystore.xml.provider"
location="/u01/data/opss/system-jazn-data.xml">
<description>File Based Policy Store Service Instance</description>
</serviceInstance>

<serviceInstance name="credstore.src" provider="credstoressp"
location="/u01/data/opss">
<description>File Based Credential Store Service Instance</
description>
</serviceInstance>

<serviceInstance name="keystore.src" provider="keystore.provider">
<description>Default JPS Keystore Service</description>
<property name="keystore.file.path" value="/u01/data/opss"/>
</serviceInstance>

<jpsContext name="src12212">
<serviceInstanceRef ref="policystore.src"/>
<serviceInstanceRef ref="pdp.service"/>
<serviceInstanceRef ref="credstore.src"/>
</jpsContext>

<jpsContext name="src12212kss">
<serviceInstanceRef ref="keystore.src"/>
</jpsContext>
```

7. Migrate OPSS data to SOA Cloud Service 12.2.1.2.

   Example commands:

```
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh

migrateSecurityStore(type="credStore",
configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
migrate-jps-config.xml", src="src12212",
dst="default")

migrateSecurityStore(type="appPolicies", src="src12212",
dst="default",
configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
migrate-jps-config.xml",
srcApp="EssNativeHostingApp", overWrite="true")
```

If SOA, OSB, and ESS have keystores other then system keystore in the customer test environment then run the following example command to migrate OPSS and keystore data.

```
//migrateSecurityStore(type="keyStore", srcConfigFile="/u01/data/
opss/migrate-jps-config.xml",
configFile="/u01/data/domains/SOAOSBta_domain/config/fmwconfig/
migrate-jps-config.xml",
src="src12212kss", dst="default", srcStripe="<SOURCE_STRIPE>")
```

Move OWSM Data

Move OWSM data by exporting it from the source and importing it to the target environment.

OWSM has the following artifacts of interest:

- CSF keys: There are references to CSF keys in OWSM policies/policy overrides. There is no change required as long as actual values are available in the credential store owned by OPSS. CSF keys must be available in the target environment.
- certs and keys: OWSM supports two types of keystores: JKS (file based) and KSS (owned by OPSS). The certificates/aliases in the source environment should be made available in the target environment. There are references to keys/certificates in OWSM policies/policy overrides.
- Custom OWSM authorization policies: These are same as custom policies.
- Custom OWSM policies

  See Exporting Documents from the Repository Using WLST in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  See Importing Documents into the Repository Using WLST in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Additional configurations that may be required: trust config and OAuth config

In 12c, `exportWSMRepository` exports all custom policies from the repository, the trust configuration, OAuth configuration, and any other configuration documents.

In 11g, the specific custom policies have to be enumerated to export them. Note that it may not be as simple as moving the documents from 11g to 12c because as part of upgrade, the OWSM upgrade plugin takes care of adding/updating 12c specific changes in the artifacts. There may be no easy way to automate policy movement from 11g to 12c as part of migration.

Steps for Migrating OWSM Data

This is an example of commands to migrate OWSM data from SOA Cloud Service 12.1.3 to SOA Cloud Service 12.2.1.2.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to 12.2.1.3 and 12.2.1.4.

When migrating OWSM data, note that only custom policies and policy sets are migrated.

1. Export OWSM data.

   On the source SOA Cloud Service 12.1.3 environment, enter the following commands before exporting OWSM data:

   ```
   ssh -i opc_rsa opc@<source_admin_host_ip>
   sudo -su oracle
   cd /u01/
   cd /app/oracle/middleware/oracle_common/common/bin/
   ./wlst.sh
   ```

   Then:

   ```
   connect('weblogic','welcome1','t3s://
   <source_admin_host_ip>:<admin_port>')
   wls:/SOAOSB12_domain/serverConfig> exportWSMRepository('/tmp/
   policies.zip',[''])
   ```

   This will export only the custom/cloned policies and custom policy sets into the `policies.zip` file under the `/tmp` folder.

2. Copy the exported file to a local computer.

   Go to the directory where the exported data is saved, then move the file from the source to the local computer.

   ```
   scp policies.zip <username:source_admin_host_ip>:/
   <local_export_dir_path>
   ```

   Example:

   ```
   scp policies.ip asmathur@adc01jjt.us.oracle.com:/scratch/exp_dat
   ```

3. Copy OWSM data from the SOA Cloud Service 12.1.3 host to the SOA Cloud Service 12.2.1.2 host.

Create a folder on the target environment. Change to the target directory folder where exported files should be copied.

```
scp <username:TARGET_SOACS_HOST_IP>:/<local_export_dir>/policies.zip
```

Example:

```
scp asmathur@adc01jjt.us.oracle.com:/scratch/export_data/
policies.zip
```

4. Import OWSM data.

   On the target SOA Cloud Service 12.2.1.2 environment, enter the following commands to import OWSM data:

```
ssh -i opc_rsa opc@host_adminip_target
sudo -su oracle
cd /u01/app/oracle/middleware/oracle_common/common/bin/
./wlst.sh


connect('weblogic','welcome1','t3s://
<target_host_ip>:<target_host_port>')
wls:/SOAB2B12_domain/serverConfig> importWSMArchive("/tmp/
policies.zip")
```

   This will import all the custom policies and the global policy sets exported in the `policies.zip` file into the target SOA Cloud Service server.

Move ESS Metadata

Since we need to export tip versions of metadata in MDS in a specific package, we can use the `exportMetadata` WLST command with docs parameter as `"/oracle/apps/ess/custom/**"` and `"/oracle/as/ess/essapp/custom/**"` to an archive. Then we can import from the archive to the target MDS repository using the `importMetadata` WLST command.

To ensure the metadata is independent of environment, we need to tokenize URLs in job definitions first. Users have to define the new token values in the target environment (if required).

For MDS `importMetadata` and `exportMetadata` commands, see exportMetadata and importMetadata in *WLST Command Reference for Infrastructure Components*.

Steps for Migrating ESS Metadata

This is an example of commands to migrate ESS metadata from SOA Cloud Service 12.1.3 to SOA Cloud Service 12.2.1.2.

> **Note:**
>
> These commands have been certified only on SOA Cloud Service 12.2.1.2 as a target environment, but are also applicable to 12.2.1.3 and 12.2.1.4.

1. On the source SOA Cloud Service 12.1.3 environment, enter the following commands:

```
exportMetadata(application='EssNativeHostingApp',server='SOAOSBLS_se
rver_1',toLocation='/u01/data/artifacts
/Custom.mar',docs='/oracle/apps/ess/custom/**')


exportMetadata(application='ESSAPP',
server='SOAOSBLS_server_1',toLocation='/tmp/
ESSAPP_custom.mar',docs='/oracle/as/ess/essapp/custom/**')
```

2. On the target SOA Cloud Service 12.2.1.2 environment, enter the following commands:

```
importMetadata(application='EssNativeHostingApp',server='SOAOSBta_se
rver_1',fromLocation='/u01/data/artifacts
/Custom.mar',docs='/oracle/apps/ess/custom/**')


importMetadata(application='ESSAPP',
server='SOAOSBta_server_1',fromLocation='/tmp/
ESSAPP_custom.mar',docs='/oracle/as/ess/essapp/custom/**')
```

Move B2B Metadata

Move B2B metadata from your source to your target environment.

For detail instructions, see Importing and Exporting Data in *Using Oracle B2B*.

Move Oracle Service Bus Projects

The easiest way to export and import Oracle Service Bus metadata is through the console. You can export all the projects with one export.

See How to Export Resources to a Configuration JAR File in the Console in *Developing Services with Oracle Service Bus*.

Move SOA Projects

The SOA composite SAR archive can be generated easily in JDeveloper 12.2.1.2 by generating a SAR archive (instead of deploying to the server). This can be deployed to the target SOA Cloud Service 12.2.1.2 server from the console, ant script or WLST script.

See Deploying SOA Composite Application in Oracle JDeveloper in *Developing SOA Applications with Oracle SOA Suite*.

Transition Inbound Adapters/Transports

For successful migration/side-by-side upgrade, you need to transition inbound adapters/transports.

There are two use cases to consider for transitioning inbound adapters/transports. During transition, you disable the inbound adapters/transport at the source and enable it on the target environment. Also, when you first deploy the projects to the target environment, you do not want inbound adapters/transports to process production messages right away until you are ready for the transition. To solve both the use cases, you can do any of the following:

- Change the etc/host file or add/remove permissions for the file directory.
- Change to composite or adapter activate/deactivate.

  SOA supports adapter activate/deactivate only in 12.1.3. In B2B, the inbound channel is disabled by default on import. Oracle Service Bus does not support this.
- Change the inbound endpoints to test or true endpoints.

  This requires a redeployment.

4

Migration/Side-by-Side Upgrade for MFT Cloud Service

Learn how to migrate your on-premises MFT application to the cloud or how to do a side-by-side upgrade in the cloud for MFT Cloud Service.

Topics

Provision MFT Cloud Service

Provision a new MFT Cloud Service instance before starting the other migration and side-by-side related tasks. You'll migrate or recreate configurations from your old source environment into this newly provisioned instance of MFT Cloud Service.

Perform the following steps:

- Provision a new Oracle MFT Cloud Service instance and configure the Oracle Cloud Infrastructure environment. Use the Oracle SOA Cloud Service provisioning wizard and select **MFT Cluster** as the service type. See Provision Oracle SOA Cloud Service Instances in Oracle Cloud Infrastructure in *Administering Oracle SOA Cloud Service in a Customer-Managed Environment*.
- Create a simple hello world application (MFT transfer) and test to make sure it works.

Prepare Clients for Migration/Side-by-Side Upgrade (MFT)

Configure and prepare your clients such that the transition of HTTP clients from the old deployment to the new deployment is smooth and happens by switching the DNS entry.

These changes can be done gradually over time because after these changes are completed everything continues to work as before the changes. This includes some changes to the source environment.

To prepare clients:

1. Get a DNS name issued from DNS issuing authority. Point this DNS name to the source environment load balancer.

   If you are already using a DNS name in clients skip this step. This step is required only if you are using an Oracle Traffic Director (OTD) IP address.

2. Create a new port in the source environment load balancer that matches the target MFT Cloud Service port number. Add routing role to this new port to route to the original load balancer port in the source environment.

   Note that on-premises MFT applications can use a different port number than the target MFT Cloud Service environment. This makes it impossible to switch clients during transition from the old to the new deployment by switching the DNS.

3. Change all clients to use the DNS name and new port.

   Note that the DNS name will also be used by clients of the embedded FTP and SFTP server that write files.

   For SSL, it might be required that the trust certificate for the target environment server has to be pre-configured at the client so that transition from the source to the target environment works smoothly.

Prepare Your Source for Migration/Side-by-Side Upgrade (MFT)

Prepare your source for migration/side-by-side upgrade by exporting or capturing the needed artifacts from the source environment.

To prepare your source:

1. Export all metadata from the source environment.

   See Importing and Exporting the MFT Configuration in *Using Oracle Managed File Transfer*.

   In earlier versions of MFT, a configuration plan (that the user can edit) is not generated during export. However, you'll require a configuration plan for import. In such cases, the easiest way to generate a configuration plan is to import it into a test 12.2.1.2 deployment and generate the configuration plan from the test deployment.

2. Grab the domain file system artifacts such as Java callouts.

3. Adjust the configuration plans for deployment to MFT Cloud Service.

   Change URLs to values appropriate for MFT Cloud Service.

Prepare Your Target Environment (MFT)

Prepare your target environment by importing or recreating all the configurations of your source. This will ensure successful deployment of the target MFT Cloud Service instance.

To prepare your target environment:

1. Implement any security configurations.

   Security configurations can be: custom Oracle Web Service Manager (OWSM) policies, Credential Store Framework (CSF) keys, certificates, users, groups,

custom Oracle Platform Security Service (OPSS) roles, custom OPSS permissions, group memberships, enterprise roles, OPSS credentials and so on.

For information on OPSS commands to migrate keystores , see Managing Keystores with WLST in *Securing Applications with Oracle Platform Security Services*.

For information on OPSS commands to migrate credentials, see Managing Credentials with WLST in *Securing Applications with Oracle Platform Security Services*.

For information on OWSM commands to migrate custom policies, see Migrating Policies in *Administering Web Services*.

> **Note:**
>
> If the source environment is MFT Cloud Service, the internal LDAP data can be migrated into the target environment MFT Cloud Service instance. Migration is supported for MFT Cloud Service 12.1.3, 12.2.1 and 12.2.1.2.

2. Import artifacts with the configuration plan. Do not deploy yet.
3. Add file system artifacts captured from source environment – Java callouts.
4. Test by creating a simple hello world application (MFT transfer). Ensure that it works.
5. Set your tuning settings if they are available.
6. Redo any Enterprise Manager configuration steps manually.

   For details, see Reconfigure Configuration Parameters and Tuning in MFT Cloud Service.
7. If MFT Cloud Service is going to access endpoints on-premises then you may need VPN.

   You can setup VPN through VPNaaS.
8. Apply UMS configuration manually to the target environment.
9. Enable the embedded SFTP server by making any documented changes.

Test Your Production Environment (MFT)

You can test your production environment at this point to check if everything is working as expected after the migration. It is assumed that you have already tested in a stage system (test environment).

To test your production environment:

1. Use endpoints to test in the configuration plans of the steps that you have completed till now. Deploy and enable everything.
2. Test and check if everything is working as expected.
3. Apply tuning settings.
4. Switch to production endpoints.

   This may require projects to be redeployed with appropriate configuration plans.

Transition from Old Deployment to New Deployment (MFT)

After you have prepared your source and target environments for the migration/side-by-side upgrade, you can transition your production system from old deployment to new deployment.

To transition from old to new deployment:

1. Disable inbound sources in the source environment, if the inbound address in both old and new deployment is same.

   For inbound sources which are remote FTP/SFTP servers, ensure that already processed files in the directory are not processed again after transitioning to the target environment by removing them from the directory.

2. Complete deployment and enable everything in the target environment.

   Note that for some inbound sources, the address is different and clients have to change the address in the source and switch.

3. Switch the DNS.

   The DNS switch is not instantaneous and may take a while (depending on TTL settings in routers) to propagate across the internet.

4. Note that the source environment will continue processing backlogged transfers while new messages are processed by the target environment. When all backlogged transfers are processed and there is no need to rollback, you can destroy the source environment.

   Switch external clients of the embedded FTP and SFTP server that read files to the new deployment, after all files in the old deployment have been processed.

5. If you have directories in the embedded FTP server for storing data that is not processed by MFT, it is up to you to either copy these files or deal with them as you see fit.

Reconfigure Configuration Parameters and Tuning in MFT Cloud Service

Re-configure any tuning and configuration parameters that you had previously set in the source environment or you need to change in the target environment.

You'll perform these steps as part of preparing your target environment for transitioning from the old to the new environment.

- Schedule Purge
- See Tuning Oracle Managed File Transfer in *Fusion Middleware Tuning Performance Guide*.

Migrate Data Components in MFT Cloud Service

Migrate your data components such as LDAP, OPSS and OWSM from your source to the target environment.

You'll perform these tasks as part of preparing your target environment for transitioning from the old to the new environment.

> **Note:**
>
> The migration steps described here are for target version of MFT Cloud Service 12.2.1.2. If you want to do the migration for other versions, refer to the product documentation.

Move LDAP Data

The tasks for migrating LDAP data in MFT Cloud Service are similar to the tasks for migrating LDAP data in SOA Cloud Service.

See Move LDAP Data.

Move OPSS Data

The tasks for migrating OPSS data in MFT Cloud Service are similar to the tasks for migrating OPSS data in SOA Cloud Service.

See Move OPSS Data.

Move OWSM Data

The tasks for migrating OWSM data in MFT Cloud Service are similar to the tasks for migrating OWSM data in SOA Cloud Service.

See Move OWSM Data.