## Contents

### Preface

- Audience vi
- Documentation Accessibility vi
- Related Resources vii
- Conventions vii

### 1  Learn About Migrating to Oracle Cloud Infrastructure

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Migrate to Oracle Cloud Infrastructure</td>
<td>1-1</td>
</tr>
<tr>
<td>About the Migration Scope</td>
<td>1-1</td>
</tr>
<tr>
<td>Oracle Integration Classic Administration Restrictions</td>
<td>1-2</td>
</tr>
<tr>
<td>Integrations Migration Scope</td>
<td>1-3</td>
</tr>
<tr>
<td>Integrations Migration Restrictions</td>
<td>1-3</td>
</tr>
<tr>
<td>Processes Migration Scope</td>
<td>1-4</td>
</tr>
<tr>
<td>Processes Migration Restrictions</td>
<td>1-4</td>
</tr>
<tr>
<td>Visual Builder Migration Scope</td>
<td>1-4</td>
</tr>
<tr>
<td>Visual Builder Migration Restrictions</td>
<td>1-6</td>
</tr>
<tr>
<td>About Oracle Cloud Infrastructure</td>
<td>1-6</td>
</tr>
<tr>
<td>About the Migration Task Flow</td>
<td>1-7</td>
</tr>
<tr>
<td>About the Migration Tooling</td>
<td>1-8</td>
</tr>
</tbody>
</table>

### 2  Prepare to Migrate Oracle Integration Classic to Oracle Cloud Infrastructure

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrations and Processes Tasks</td>
<td>2-1</td>
</tr>
<tr>
<td>Plan Your Migration</td>
<td>2-1</td>
</tr>
<tr>
<td>Prepare to Migrate</td>
<td>2-2</td>
</tr>
<tr>
<td>Size Oracle Integration Instances</td>
<td>2-2</td>
</tr>
<tr>
<td>Size Processes</td>
<td>2-3</td>
</tr>
<tr>
<td>Whitelist IP Addresses</td>
<td>2-4</td>
</tr>
<tr>
<td>Visual Builder Tasks</td>
<td>2-5</td>
</tr>
<tr>
<td>Create and Validate Oracle Integration</td>
<td>2-6</td>
</tr>
</tbody>
</table>
3 Migrate Your Oracle Integration Classic Instances to Oracle Cloud Infrastructure

Create a Cloud Storage Compartment, User, Group, and Policies 3-1
Create an Object Storage Bucket 3-3
Migrate Integrations Design-Time Metadata into Oracle Integration 3-6
   Export the Integrations Design-Time Metadata 3-7
   Import Integrations Design-Time Metadata into Oracle Integration 3-8
   Check the Integrations Design-Time Metadata Import Status 3-9
Migrate Processes Design-Time Metadata into Oracle Integration 3-10
   Import Processes Design-Time Metadata into Oracle Integration 3-10
   Import Using the Import Command Line Tool 3-11
Migrate Your Visual Builder Design-Time Metadata 3-13
   Export Visual Builder Design-Time Metadata 3-13
   Import Visual Builder Design-Time Metadata 3-14
   Import Visual Builder Design-Time Metadata from a Git Repository 3-14

4 Complete the Post-Migration Tasks

Verify Your Environment and Activate Your Integrations 4-1
Complete Post-Migration Tasks for Processes 4-1
   Perform Post Import Steps for Processes 4-1
   Handle In-Flight Process Instances 4-3
   Follow Migration Best Practices for Processes 4-4
Complete Post-Migration Tasks for Visual Builder Applications 4-4
   Lock the Live Visual Builder Application 4-4
   Export Application Data From the Live Database 4-5
   Import the Live Application Data into the New Application 4-5
   Publish the New Application with the Live Application Data 4-6

A Migrate Integrations and Processes Together

Perform Oracle Cloud Infrastructure Prerequisites A-2
Create an Oracle Cloud Infrastructure Compute Classic Instance A-2
Configure the Oracle Cloud Infrastructure Compute Classic Instance A-6
Update the Default Profile for Your Oracle Cloud Infrastructure Classic Services A-8
Update the Oracle Cloud Infrastructure Configuration File A-10
Learn About Available Commands A-11
Discover Instances in Your Source Environment A-12
List Your Oracle Integration Instances A-13
Export Design-Time Metadata A-13
Create the Target Instance on Oracle Cloud Infrastructure A-17
Edit the Target Configuration File A-18
Import Design-Time Metadata A-19
Preface

*Migrating Oracle Integration Classic Instances to Oracle Cloud Infrastructure* describes how to migrate Oracle Integration Classic to Oracle Integration on Oracle Cloud Infrastructure.

**Note:**

The information in this guide applies to all of your Oracle Integration instances. It doesn’t matter which edition you’re using, what features you have, or who manages your cloud environment. You’ll find what you need here, including notes about any differences between the various flavors of Oracle Integration when necessary.

Topics:

- Audience
- Documentation Accessibility
- Related Resources
- Conventions

Audience

*Migrating Oracle Integration Classic Instances to Oracle Cloud Infrastructure* is intended for users who need to migrate Oracle Integration Classic to Oracle Integration on Oracle Cloud Infrastructure.

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 Integration documentation in the Oracle Cloud Library on the Oracle Help Center.

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Learn About Migrating to Oracle Cloud Infrastructure

Learn about the benefits to migrating your existing Oracle Integration Classic instances to Oracle Integration on Oracle Cloud Infrastructure, and receive an overview of the migration process and tools.

Topics:
- Why Migrate to Oracle Cloud Infrastructure
- About the Migration Scope
- About Oracle Cloud Infrastructure
- About the Migration Task Flow
- About the Migration Tooling

Why Migrate to Oracle Cloud Infrastructure

Oracle encourages you to migrate your existing cloud resources from Oracle Cloud Infrastructure Classic regions. You can gain several advantages by doing so.

In Oracle Cloud, you provision resources in specific regions, which are localized to geographic locations. Certain regions support the Oracle Cloud Infrastructure platform.

Oracle Cloud Infrastructure is Oracle's modern cloud platform that's based on the latest cloud technologies and standards. It provides more consistent performance and better features at lower costs. Oracle continues to invest in Oracle Cloud Infrastructure, including the addition of new regions, services, and features. See Data Regions for Platform and Infrastructure Services.

You can benefit from these additional administrative features when you migrate your cloud resources to Oracle Cloud Infrastructure:
- Organize cloud resources into a hierarchy of logical compartments.
- Create fine-grained access policies for each compartment.

About the Migration Scope

Before migrating your existing Oracle Integration Classic instances to Oracle Integration on Oracle Cloud Infrastructure, consider the scope and constraints of this migration path. There are restrictions to consider at the Oracle Integration Classic level and at the individual Integrations, Processes, and Visual Builder levels.
- Oracle Integration Classic Administration Restrictions
- Integrations Migration Scope
Oracle Integration Classic Administration Restrictions

Understand the following restrictions when migrating Oracle Integration Classic to Oracle Integration on Oracle Cloud Infrastructure. In addition to Oracle Integration Classic administration restrictions, there are also Integrations, Processes, and Visual Builder restrictions described in subsequent sections. Once migration is complete, you no longer manage your instances. Oracle manages your instances, including performing all backups and patching/upgrading. This enables you to concentrate on developing your integrations, processes, and applications.

Restrictions By Design

With the move to Oracle-managed instances, the following restrictions are by design.

- Any custom configurations done outside of the Oracle Integration Classic My Services Console are not migrated. For example:
  - Configurations performed through direct VM access
  - Configurations performed through consoles such as Oracle WebLogic Server Console, Oracle Enterprise Manager Console, and others
  - Any external processes you created (cron jobs, Javascripting, and so on)

There are additional customizations that are not supported. See Unsupported Tasks of Administering Oracle Integration.

- You cannot log in to and access virtual machines (VMs).
- There is no log management through VM access.
- SSH access is not supported.
- IP addresses change between Oracle Integration Classic and Oracle Integration, meaning you must update any client applications with which your integrations communicate (for example, Oracle ERP Cloud, Oracle HCM Cloud, and others).
- You cannot log in and manage the Oracle Cloud Infrastructure database (and therefore, application data). Oracle performs all backup and patching/upgrading tasks. The only administration tasks you can perform are those accessible from the Oracle Integration My Services Console.
- Custom XPath functions and custom Javascript code are not supported.
- Life cycle management REST APIs (for example, backing up, restoring, and patching) are not supported.
- Transport Level Security versions 1.0 and 1.1 are not supported. Oracle Integration endpoints support only TLS 1.2 as a trigger connection. Ensure that you configure your client to use TLS 1.2 when invoking Oracle Integration services.
**Additional Restrictions**

- Private endpoints are not supported.
- There is no virtual private network (VPN) or FastConnect access.
- Integration Analytics (which consists of Stream Analytics and Integration Insight) is not supported.

**Integrations Migration Scope**

You export Integrations design-time metadata into an archive file to then import into Oracle Integration on Oracle Cloud Infrastructure. The archive file consists of the following design-time metadata.

- Integrations, connections, lookups, agent groups, and so on. Note that:
  - Integrations, connections, or objects in any state (in-progress, activated, and so on) are exported.
  - All resources such as lookups and connections that are not currently referenced by integrations are exported.

- Endpoint configurations

- User-defined credentials. Note the following details:
  - Credentials are exported into cwallet.sso.
  - The oracle.cloud.adapter map and oracle.wsm.security maps are imported.

- User-defined certificates (not the seeded certificates). Only user-uploaded trusted certificates (whose alias begins with icsuser_||_) from the following keystores are exported:
  - system/trust
  - owsm/keystore
  - ics/keystore

- All security policies. Existing policies are not overwritten.
- Connection passwords stored in the CSF store.
- Settings such as database settings, notification settings, and so on.
- Recommendations engine details and API Platform connection details.

**Integrations Migration Restrictions**

Understand the following restrictions when migrating Integrations to Oracle Integration on Oracle Cloud Infrastructure.

- Logging settings that you configured in are not migrated.
- A best effort is made to migrate database settings.
- After importing a scheduled integration (scheduled is started) from Oracle Integration Classic into Oracle Integration, the integration is imported and the schedule is started automatically. You must manually stop the schedule in Oracle Integration, if necessary.
If you modify the default value of the recovery job in Oracle Integration Classic, the upgrade to Oracle Integration resets the value to the default value.

Only one export at a time can be started. Subsequent export requests are rejected if one is currently running.

If an integration uses the on-premises connectivity agent, those integrations have to be manually activated after registering the agents manually.

If a parent integration calls a child integration, the child integration must be manually activated. This is because the child must be activated after the parent.

Data of the same name is overwritten. For example, if an integration of the same name and version exists in Oracle Integration, it is overwritten by the integration of the same name and version imported from Oracle Integration Classic.

Instance runtime data such as monitoring, tracking, and error details is not migrated.

Custom adapters and their integrations are not migrated. File a service request to have your custom adapters and their integrations included in Oracle Integration.

Processes Migration Scope

Use the Process Import tool to import Processes design-time metadata into Oracle Integration on Oracle Cloud Infrastructure. You can import the following design-time metadata.

- Process applications
- Decision models

Processes Migration Restrictions

Understand the following restrictions when migrating to Oracle Integration on Oracle Cloud Infrastructure.

- **Application user role (swimlane) mapping**: Process role mappings are not migrated. You must remap Process user roles (swim lanes) for all process applications after activation.

- **Running instances**: Instances cannot be moved between environments.
  - Running (in flight) process instances and tasks are not migrated to Oracle Cloud Infrastructure.
  - Completed process instances and tasks are not migrated to Oracle Cloud Infrastructure.

After importing process applications into Oracle Integration, you must activate them and create new running instances.

Visual Builder Migration Scope

You export the design-time metadata for each visual application into an archive file to then import into the new instance on Oracle Cloud Infrastructure.

The archive file contains the design-time metadata for the applications in your visual application, plus a variety of other files that your visual application needs once it is imported in the new instance. The following directory structure for a visual application
that contains a web and mobile app, plus a business object provides an illustrative example of the type of metadata that an archive file contains.

```
VisualApplicationArchiveDirectory
+----businessObjects
    |   \---Department
+----mobileApps
    |   \---hmobileapp
    |       +----flows
    |       |   \---main
    |       |       \---pages
    |       +----pages
    |       |   \---resources
    |       |       \---strings
    |       |           \---app
    |       |               \---nls
    |       |                   \---root
    ...+
    \---settings
        \---mobile-build-templates
+----process
    |   \---pcs
+----services
+----settings
+----webApps
    |   \---hrwebapp
    |       +----flows
    |       |   \---main
    |       |       \---pages
    |       +----pages
    |       |   \---resources
    ...+
    \---resources
        +----css
        \---strings
            \---app
                \---nls
                \---root
            \---settings
```

When you export the visual application you can choose if you want the archive to include the development data contained in the application's custom business objects. Some information, such as the user credentials to access external REST end points, is removed when you export a visual application. Also, mobile build configurations are not exported. A mobile configuration comprises of artifacts like keystore, iOS provisioning profiles, and passwords. Export this information and artifacts separately so that you can provide it after the archive is imported into the new instance on Oracle Cloud Infrastructure.

Apart from the design-time metadata for each visual application, you also need to export the application data for live applications. Before you migrate this data, you need to lock the live application which prevents end users from accessing the application.
Visual Builder Migration Restrictions

Understand the following restrictions when migrating to a new instance on Oracle Cloud Infrastructure.

You can only migrate the latest version of an existing visual application from the Oracle Cloud Infrastructure Classic instance to Oracle Cloud Infrastructure. Visual Builder does not have a mechanism to import previous versions of the visual application into the new instance. Also, you can only export one version of the visual application from the Oracle Cloud Infrastructure Classic instance at a time. You cannot export all versions of a visual application in one action.

Post-migration, you need to communicate the new URLs that web app end users will use to access the applications that are hosted on the new instance of Oracle Cloud Infrastructure. For mobile app end users, you need to rebuild the mobile app on the Oracle Cloud Infrastructure instance, publish it to the appropriate app store, and inform your end users that they need to update to the newer version of the mobile app.

About Oracle Cloud Infrastructure

Get familiar with basic Oracle Cloud Infrastructure security, network, and storage concepts, and their equivalent concepts in Oracle Cloud Infrastructure Classic.

Cloud resources in Oracle Cloud Infrastructure are created in logical compartments. You also create fine-grained policies to control access to the resources within a compartment.

You create instances within an Oracle Cloud Infrastructure region. You also specify an availability domain (AD), if supported in the selected region. Oracle Cloud Infrastructure Classic does not use availability domains.

A virtual cloud network (VCN) is comprised of one or more subnets, and an instance is assigned to a specific subnet. In Oracle Cloud Infrastructure Classic, you assign instances to IP networks or the shared network. Typically, you create one subnet for the shared network, and create a separate subnet for each IP network in Oracle Cloud Infrastructure Classic. Note that unlike Oracle Cloud Infrastructure Classic, Oracle Cloud Infrastructure does not allow you to reserve IP addresses for platform services.

A subnet's security lists permit and block traffic to and from specific IP addresses and ports. In Oracle Cloud Infrastructure Classic, an instance's access rules provide similar capabilities, although security lists are configured at the subnet level.

Instances can communicate with resources outside of Oracle Cloud by using Oracle Cloud Infrastructure FastConnect, which provides a fast, dedicated connection to your on-premises network. This service is equivalent to Oracle Cloud Infrastructure FastConnect Classic. Alternatively, use IPSec VPN in Oracle Cloud Infrastructure as a replacement for VPN as a Service (VPNaaS) or Corente in Oracle Cloud Infrastructure Classic.

A bucket in Oracle Cloud Infrastructure Object Storage can be used to store files and share them with multiple instances. A user's generated authentication token (auth token) is required to access the bucket. Oracle Cloud Infrastructure Object Storage Classic provides the same service in Oracle Cloud Infrastructure Classic, but does not use auth tokens.
To learn more, see Key Concepts and Terminology in the Oracle Cloud Infrastructure documentation.

About the Migration Task Flow

Get an overview of the process that you use to migrate your existing Oracle Integration Classic instances to Oracle Integration on Oracle Cloud Infrastructure.

The following diagram shows the migration flow of Oracle Integration Classic instances that include Integrations, Processes, and Visual Builder design-time metadata to Oracle Integration on Oracle Cloud Infrastructure. The components to migrate in your instances may vary.

Where:

- **1** - Use the `curl` or `postman` command to export and import Integrations design-time metadata.
- **2** - Use the import tool to import Processes design-time metadata (user interface recommended or command line).
- **3** - Use the `opcmigrate` tool to export and import all Integrations and Processes design-time metadata together.
- **4** - Use the Export and Import menu options in Visual Builder's Home page to export and import design-time metadata.
- **5** - Use the Visual Builder Data Manager screen to export and import application data.

At a high level, the migration process is comprised of these tasks:

1. Plan and prepare for the migration and perform any prerequisite tasks in Oracle Cloud Infrastructure, if necessary.
2. Provision the target Oracle Integration instance in an Oracle Cloud Infrastructure region.

3. If you are migrating Integrations, use **curl** or **postman** commands to export and import Integrations design-time metadata into Oracle Integration on Oracle Cloud Infrastructure.

4. If you are migrating Processes, use the import tool (user interface or command line) to import Processes design-time metadata into Oracle Integration on Oracle Cloud Infrastructure.

5. If you are migrating Integrations and Processes together, use the **opcmigrate** tool to import any Integrations and Processes design-time metadata together into Oracle Integration on Oracle Cloud Infrastructure. See Migrate Integrations and Processes Together.

6. If you are migrating Visual Builder, use the Import menu option on the Visual Builder Home page to import Visual Builder design-time metadata (visual applications) into Oracle Integration on Oracle Cloud Infrastructure.

7. Test your applications on the target instance, and perform any other post-migration tasks.

### About the Migration Tooling

You use the following migration tooling to migrate Oracle Integration Classic and instances to Oracle Integration on Oracle Cloud Infrastructure.

- **Integrations only**: Use **curl** or **postman** commands to export and import design-time metadata. When you export metadata, an archive file is asynchronously created. You import the archive file into Oracle Integration.

- **Processes only**: Use the Import tool (user interface or command line). Using the user interface is strongly recommended.

- **Integrations and Processes together**: Use the Oracle Cloud Infrastructure Classic Discovery and Translation Tool (**opcmigrate**) to export and import any Integrations and Processes design-time metadata in your instance. This tool helps you to discover Oracle Integration Classic service instances in your source environment, create Oracle Integration service instances on Oracle Cloud Infrastructure, export design-time metadata from an Oracle Integration Classic service instance to an archive file, and import an archive file into an Oracle Integration service instance.

- **Visual Builder**: Use Visual Builder's menu options to export the design-time metadata (visual applications) to an archive file that you then import into Oracle Integration.
Prepare to Migrate Oracle Integration Classic to Oracle Cloud Infrastructure

Before you migrate Oracle Integration Classic instances to Oracle Cloud Infrastructure, plan and prepare for migration.

Topics:
• Integrations and Processes Tasks
• Visual Builder Tasks
• Create and Validate Oracle Integration

Integrations and Processes Tasks

Plan and prepare for Oracle Integration Classic migration to Oracle Cloud Infrastructure.

Topics:
• Plan Your Migration
• Prepare to Migrate

Plan Your Migration

Review the following considerations when planning your migration from Oracle Integration Classic to Oracle Integration on Oracle Cloud Infrastructure.

• Migration considerations:
  – Time sensitive migration: You have a window in which to migrate Oracle Integration Classic instances to an Oracle Integration instance. Contact your customer support representative for details.
  – Side-by-side migration: You pay to run Oracle Integration Classic side-by-side with Oracle Integration. This option is recommended for high traffic Oracle Integration Classic environments running business-critical integrations and processes in which a window to migrate is not feasible. This option enables you to migrate integrations and processes in a phased fashion.
• How many instances (development, test, and production) do you have. This enables you to determine the number of Oracle Integration instances to provision.
• Which data centers host the instances. This is important because Oracle Integration may not be available in the same data center regions as your Oracle Integration Classic instances. This means you need to identify other regions in which to run your instances. See https://cloud.oracle.com/data-regions.
• Migration time line: Understand why and when to migrate to Oracle Integration.
• Big bang versus staggered: If you are running business critical integrations, you cannot migrate everything at once. Create a new Oracle Integration instance for some integrations while continuing to run other instances on Oracle Integration Classic.

• Devise a cut over strategy: Determine the best time for your company to migrate to Oracle Integration.

• Work out your financial details with the customer sales team.

Prepare to Migrate

Prepare to migrate your Oracle Integration Classic instances to Oracle Integration.

Review the following options when preparing your migration:

• **Size Oracle Integration Instances**
• **Size Processes**
• **Whitelist IP Addresses**

Size Oracle Integration Instances

Size your Oracle Integration instances to determine the number of message packs required for your environment.

The concept of sizing is different between Oracle Integration Classic and Oracle Integration.

• Oracle Integration Classic is based on CPUs.

• Oracle Integration sizing is based on message packs.

There is no 1:1 correspondence between the number of CPUs and the number of message packs. Therefore, you must estimate the relationship. Follow these sizing best practices to determine the number of message packs needed:

• Identify the number of Oracle Integration Classic messages by monitoring the Dashboard page. For example:
  – How many messages were processed over a specific time period (for example, three days).
  – What are the message types being processed (for example, file-based messages or SOAP/REST-based messages).

For specific questions, contact your customer support representative for advice on sizing.

• Configure the development environment with different message pack numbers than test and production environments. For example, it is recommended that you configure development to use a minimal number of message packs because you are not running load or performance tests in that environment.

• Size the test environment similar to the development environment, unless you know that you have regular usages large enough to require more message packs. A test environment may only run a fraction of the production environment volume most of the time. Typically, a performance test environment would run regular production volumes or more for a short period of time (such as two weeks).
• Always estimate the number of messages before deciding on the number of message packs to use.
• Guesstimates are fine. The message packs can be increased or decreased, if necessary, from within the Oracle Cloud Infrastructure Console.

Size Processes

Determine message packs needed for Processes for your environment, then add that number to those needed for standalone Integrations and other Oracle Integration features.

Oracle Integration Process sizing is based on concurrent users, which are converted to message packs, and added to message packs needed for integrations.

Each concurrent user (which is a distinct user during that hour) per hour is counted as 400 messages per user. This number is added to the integration volume towards the 5,000 messages per hour. If you have 1,000 messages per hour and 10 distinct users, these would count as 1,000 integration messages + (400)*10 users = 5,000, so 1 message pack of 5,000 messages per hour would suffice.

Another way to visualize Process sizing: 5,000 message packs per hour equate to 12.5 distinct concurrent users performing tasks.

What's counted?

A logged in user is counted for a minimum of one hour when performing any write operations that update a task or process instance, which includes:
• Updating or processing tasks (approve/reject a task, add an attachment/comment, re-assign, or request for information)
• Creating process instances
Within each hour of use, a distinct user can perform an unlimited number of write operations.

Oracle Integration has a 1 message pack minimum charge per hour to keep the system available, even with no usage. Note that you can turn off your Oracle Integration instance for billing purposes, but no instances are processed while the instance is stopped.

What's NOT counted?

This count doesn't include:
• Logged in users performing read-only only (query or read) operations.
• Integrations triggered from the process (integrations are waived).

Example 1

Between 9am and 10am, 20 employees access Workspace. Within the one hour timeframe:
• 5 users (user1 through user5) create a total of 100 new process instances.
• 10 other users (user6 through user15) process different tasks created by user1 through user5, and complete them.
• The remaining 5 users (user16 through user20) only check the task and process instance status, but do not perform any update/write operations.
Result: The 9am-10 am hour block reports 15 concurrent users (5 created new instances and 10 processed tasks).

Example 2

Between 10 and 11am, 10 users access Workspace and 5 access the Oracle Process Mobile app. Within the one hour timeframe:

- 10 users (user1 through user10) create new process instances and also approve at least 1 task total.
- 5 users (user11 through user15) log into the mobile app: 3 of them create new instances, and the other 2 perform only read-only operations.

Result: The 10am-11am hour block reports 13 concurrent users (10 workspace users plus 3 mobile users performed update/write operations, while 2 mobile users did not perform any update/write operations).

Example 3

Between 11am and 12pm, 5 users access Oracle Integration from a Visual Builder application and 5 other users access Workspace.

- 2 of the 5 Visual Builder users access Visual Builder, and interact with a Visual Builder app that in turn triggers execution of an API that creates new process instances and processes tasks.
- The other 3 Visual Builder users access the Visual Builder app and read and access task and process instance status.
- The 5 users access Workspace and approve a minimum of 1 task each within the hour timeframe.

Result: The 11am-12pm hour block reports 7 concurrent users (2 Visual Builder users and 5 Workspace users performed update/write operations). This result does not include the Visual Builder concurrent user licenses. Visual Builder concurrent users need to be sized separately.

Whitelist IP Addresses

You may have whitelisted your Oracle Integration Classic IP addresses. For example, to access an FTP server or perhaps Oracle ERP Cloud calls back to Oracle Integration Classic and you white listed some of the Oracle Integration Classic IP addresses with Oracle ERP Cloud.

Perform the following steps:

- Provision your instances. This is because whitelisting IP addresses can take approximately three weeks.
- File a service request to whitelist IP addresses. Include information from the About button of your Oracle Integration instance. To access the About button:
  1. Go to the Oracle Integration Home page.
  2. In the upper right corner, click the **username** icon.
  3. From the list, select **About**.
Visual Builder Tasks

Review the following list to understand the tasks that you need to complete to migrate Visual Builder applications from your Oracle Integration Classic instance to Oracle Cloud Infrastructure.

1. Review the entries for users and roles in Oracle Identity Cloud Service to ensure that the instance of Oracle Identity Cloud Service that the new instance uses matches the entries in the instance of Oracle Identity Cloud Service used by Oracle Integration Classic. This task is not necessary if both old and new instances use the same instance of Oracle Identity Cloud Service. If you use a new instance of Oracle Identity Cloud Service, make sure that your application users are granted access to the new instance with the appropriate roles.

2. For each visual application (and for each version of each visual application that you want to migrate), export the visual application from the Oracle Integration Classic instance.

3. If you use build tools and pipelines from Oracle Developer Cloud Service, configure build jobs to point to your new instance and rebuild. Otherwise, follow the next steps to manually export and import visual applications.

4. If you use Oracle Developer Cloud Service with a connection to a Git repository to manage visual applications, use Visual Builder's Import menu in the new instance to import the visual application. Each user who collaborates on a Git-managed visual application needs to perform this import. Otherwise, follow the next steps to manually export and import visual applications.

5. For each application (and for each version of each visual application you want to migrate), export the visual application from the old instance.

6. For each visual application, import the visual application into the Oracle Integration instance.

   **Note:**
   It is not possible today to import a version of a visual application; import always creates a new visual application in the Oracle Integration instance.

7. For each visual application, ensure proper mapping of virtual roles to Oracle Identity Cloud Service groups and app roles.

8. If using team collaboration features, add team members to the newly-imported visual application as necessary.

9. For each visual application, re-enter security-related details for your visual application that are not captured during export from the old instance. This includes any client IDs and basic authentication details, along with build configurations entries needed to build mobile apps.

10. For each visual application, stage and publish the visual application in the Oracle Integration instance.

11. Test the behavior of the migrated visual application. To perform testing, you may want to export application data from the visual application on the classic instance and import it to the new visual application on Oracle Integration. When you
complete testing, perform the following post-migration steps to finish the migration
of your visual applications.
If your application is embedded with an iFrame in a Fusion Application, update and
test your Fusion Application to ensure your embedded application continues to
function post-migration.

12. For each migrated visual application, lock the visual application on the old
instance.

13. For each migrated visual application, use the Data Manager screen to export live
data from the old instance.

14. For each migrated visual application, use the Data Manager screen to import live
data to your new instance.

15. Inform end user clients to use the URL for the new service instance and to update
any bookmarks they may have.

Note: This applies to accessing apps through a browser and client applications
that may access business object REST APIs hosted in the new instance.

16. Mobile apps that do not use business object REST APIs and bypass Oracle Visual
Builder authentication proxy should continue working the same post-migration.
Otherwise, build a new version of your mobile app on the new instance and submit
it to the appropriate app store for distribution to end users.

Create and Validate Oracle Integration

Create and validate new Oracle Integration instances. Once creation and validation
are complete, you can migrate your Oracle Integration Classic instances to Oracle
Integration.

Perform the following options:

• Create Oracle Integration instances (development, test, and production). There
are two creation options:
  – Through the My Services Console. See Create an Oracle Integration Instance in Administering Oracle Integration.
  – Through opcmigrate. This tool enables you to discover Oracle Integration
Classic instances, provision an Oracle Integration instance, and export and
import Integrations and Processes design-time metadata. See Migrate
Integrations and Processes Together.

• Validate development, test, and production instance setup.
Migrate Your Oracle Integration Classic Instances to Oracle Cloud Infrastructure

Migrate Oracle Integration Classic instances to Oracle Integration on Oracle Cloud Infrastructure.

Topics:
• Create a Cloud Storage Compartment, User, Group, and Policies
• Create an Object Storage Bucket
• Migrate Integrations Design-Time Metadata into Oracle Integration
• Migrate Processes Design-Time Metadata into Oracle Integration
• Migrate Your Visual Builder Design-Time Metadata
• Migrate Integrations and Processes Together

Note:
No manual migration of users and roles is required for Oracle Integration Classic to Oracle Integration on Oracle Cloud Infrastructure migration. Both environments already use Oracle Identity Cloud Service to manage users and roles.

Create a Cloud Storage Compartment, User, Group, and Policies

Oracle Integration instances use the Oracle Cloud Infrastructure as their underlying infrastructure. To export design-time metadata into an Oracle Integration instance, you must first create a compartment (if one does not exist), user, group, and policies.

Note:
You must have an Oracle Storage Cloud Service account to perform these steps.

1. Log in to the Oracle Cloud Infrastructure Console as the Administrator user.
2. Open the navigation menu.
3. Under Governance and Administration, go to Identity, then click Compartments.
   A list of the compartments in your tenancy is displayed.
4. Click **Create Compartment** to create the compartment to use for upgrading.

5. Enter the following:
   - **Name**: Enter a name that is unique across all compartments in your tenancy (maximum 100 characters, including letters, numbers, periods, hyphens, and underscores).
   - **Description**: Enter a description for this compartment.
   - **Tags**: Enter tags to organize and list resources based on your business needs.

6. Click **Create Compartment**.

7. Return to the navigation pane.

8. Go to **Governance and Administration > Identity > Users** to create the user to use for migration.

9. Click **Create User**.

10. Enter a name and description.

11. Click **Create**.

12. Under **Governance and Administration**, go to **Identity**, then click **Groups**.

13. Click **Create Group** to export your design-time metadata.

14. Enter a name and description.

15. Click **Create Group**.

   You are now ready to add the user to the group. When configuration is complete, users in this group can export design-time metadata into Oracle Integration.

16. Locate the group you created in the list.

17. Click the group. Its details are displayed.

18. Click **Add User to Group**.

19. Select the user you created from the list, then click **Add User**.

20. Return to the navigation menu.

21. Under **Governance and Administration**, go to **Identity**, then click **Policies**.

22. Select the compartment you previously created.

23. Click **Create Policy**.

24. Enter a name and optional description.

25. Leave **KEEP POLICY CURRENT** as the **Policy Versioning** selection.

26. Add the following statements. You can either add three statements or condense them into two statements.

   - As three statements:
     - allow group Upgrade_Group to manage buckets in compartment Upgrade_Compartment
     - allow group Upgrade_Group to manage objects in compartment Upgrade_Compartment where any
allow group Upgrade_Group to manage objects in compartment Upgrade_Compartment where any
{request.permission='OBJECT_CREATE', request.permission='OBJECT_INSPECT'}

As two statements:
- allow group Upgrade_Group to manage buckets in compartment Upgrade_Compartment
- allow group Upgrade_Group to manage objects in compartment Upgrade_Compartment where any
{request.permission='OBJECT_CREATE', request.permission='OBJECT_DELETE', request.permission='OBJECT_INSPECT'}

where:
- Upgrade_Group is the group you created.
- Upgrade_Compartment is the compartment you created.

This action enables users in this group to export design-time metadata to Oracle Integration.

27. Click Create.

Create an Object Storage Bucket

To export design-time metadata into an Oracle Integration instance, you must create a storage bucket if one does not already exist.

1. Log in to the My Services Console with the user that the administrator added to the group for exporting design-time metadata. You must have the admin role to access this page.

2. If the Compute section is not displayed, select Customize Dashboard in the upper right corner of the My Services Dashboard.
   a. Click Show in the Compute section. This enables this service to appear on the My Services page.

3. Click Compute to access the Oracle Cloud Infrastructure page.

4. In the upper right corner, click Open Service Console.

5. In the upper left corner of the Oracle Cloud Infrastructure page, click .


7. Select the compartment in which to create the storage bucket. The compartment to use for exporting design-time metadata was created in Create a Cloud Storage
Compartment, User, Group, and Policies. You can also click Learn more about Compartments to find the appropriate compartment for your environment.

8. After determining the compartment to use, click Create Bucket.

![Object Storage and Buckets in Integration Compartment]

9. In the Bucket Name field, enter a name.

10. In the Storage Tier section, select Standard.

11. Click Create Bucket.

The new bucket is displayed in the list.

![Buckets in Integration Compartment]

12. If you want to view bucket details, click the name (for this example, cloneRepo).

Note:

Ensure that you click the Update Visibility button. This action ensures that the storage bucket you create is publicly visible. Otherwise, the export utility cannot see the storage.

Visibility: Public

13. In the upper left corner of the page, click ☐.

14. Select Identity > Users. This opens the Oracle Identity Service Console to generate a password.

15. Select the appropriate user.
16. From the **Resources** list on the left side of the page, select **Auth Tokens**.

17. Click **Generate Token**.

18. Provide a token description and click **Generate Token**.

This token is used as part of your JSON payload when upgrading.

19. Copy and save the generated token.

   a. The storage payload format is as follows with object storage bucket details:

   ```json
   {"storageInfo":
   {
       "storageUrl":"https://swiftobjectstorage.region.oraclecloud.com/v1/
       namespace/bucket",
       "storageUser":"my.email@oracle.com",
       "storagePassword":"generated_token"
   }
   }
   
   For example:

   ```json
   {
   "storageInfo":
   {
       "storageUrl":"https://swiftobjectstorage.us-ashburn-1.oraclecloud.
       com/v1/paasdevoic/cloneRepo",
       "storageUser":"my.email@oracle.com",
       "storagePassword":"generated_token"
   }
   }
   ```

20. Construct the storage URL.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter the Swift API as the base for the URL because you are dealing with object storage.</td>
<td><a href="https://swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket">https://swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket</a></td>
</tr>
<tr>
<td>2</td>
<td>Enter a required data center. To find a data center, navigate to <strong>Menu &gt; Administration &gt; Tenancy</strong> Details &gt; <strong>Home Region</strong> (for example: us-ashburn-1).</td>
<td><a href="https://swiftobjectstorage.us-ashburn-1">https://swiftobjectstorage.us-ashburn-1</a></td>
</tr>
<tr>
<td>3</td>
<td>Enter the domain name. This is typically <code>oraclecloud.com</code>.</td>
<td><a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com</a></td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Enter the version: v1.</td>
<td><a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1</a></td>
</tr>
<tr>
<td>5</td>
<td>Enter the automatically generated object storage namespace. Open the User menu and click Tenancy: your_tenancy_name. The namespace string is listed under Object Storage Settings. See Understanding Object Storage Namespaces.</td>
<td>paasdevoic</td>
</tr>
<tr>
<td>6</td>
<td>Enter the bucket name as the last part of the URL.</td>
<td><a href="https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/BUCKET_NAME">https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/BUCKET_NAME</a></td>
</tr>
</tbody>
</table>

Examples of the full storage URL to use later in the payload are as follows:

https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/cloneRepo

https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/paasdevoic/migrationartifacts_bucket-20190717-1223

Migrate Integrations Design-Time Metadata into Oracle Integration

You can export an archive of design-time metadata from an Oracle Integration Classic instance and import that archive into an Oracle Integration instance. All design-time metadata for Integrations gets migrated during import, including security and credential settings, IDCS role mappings, connection endpoints and password, and other dependencies. Integration activations occur automatically upon import if ImportActivateMode is **set** to ImportActivate.

- Export the Integrations Design-Time Metadata
- Import Integrations Design-Time Metadata into Oracle Integration
- Check the Integrations Design-Time Metadata Import Status
Export the Integrations Design-Time Metadata

You must export the Integrations design-time metadata.

1. For the Integrations design-time metadata of the Oracle Integration Classic instance to export to Oracle Integration, invoke the REST API. This action asynchronously creates an archive that includes all above-mentioned objects.

   Headers
   Authorization : Basic
   Content-Type : application/json
   payload
   {
     "storageInfo": {
       "storageUrl": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo",
       "storageUser": "myemail@company.com",
       "storagePassword": "generated_token"
     }
   }

   For example, use a postman or curl (as shown below) command to export all Integrations design time metadata to an Oracle Storage Cloud Service instance that you specify:

   curl -k -v -H "Content-Type: application/json" -X POST -d '{"storageInfo":{"storageUrl": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo","storageUser": "myemail@company.com","storagePassword": "generated_token"}}' -u admin:password https://host/ic/api/common/v1/exportServiceInstanceArchive

   where:
   • storageInfo: Is the URL of the storage container.
   • storageUser: Is the storage user name.
   • storagePassword: Is the storage password.

   Possible sample output from this command is as follows:

   {
     "archiveFilename": "archive_Local_Suite_Instance-d1e4295f-e17a-498a-a96e-44dcb417dfb4.zip",
     "jobID": "d1e4295f-e17a-498a-a96e-44dcb417dfb4",
     "location": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo",
     "status": "Starting"
2. Check the status of the export operation using a `postman` or `curl` (as shown below) command:

```
curl -k -v  -X GET -u admin:password https://host/ic/api/common/v1/exportServiceInstanceArchive/\{jobId\}
```

Possible sample output from this command is as follows:

```
{
   "status": "COMPLETED"
}
```

The archive is created in the Oracle Storage Cloud Service instance of Oracle Integration.

3. If the status is completed, you are now ready to import the archive.

### Import Integrations Design-Time Metadata into Oracle Integration

After exporting Integrations design-time metadata to a zip archive file, import the file to Oracle Integration on Oracle Cloud Infrastructure.

```
{
   "archiveFile": "archive_file_name",
   "importActivateMode": "ImportActivate",
   // options are "ImportOnly" || "ActivateOnly" || "ImportActivate"
   "storageInfo": {
      // storageUrl points to the storage container
      "storageUrl": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo",
      "storageUser": "myemail@company.com",
      "storagePassword": "generated_token"
   }
}
```

1. To import the archive, go to the Oracle Integration instance and invoke the REST API. This action retrieves the archive from the Oracle Storage Cloud Service instance where the archive was created.

Headers

- **Authorization**: Basic
- **Content-Type**: application/json

```
payload
{
   "archiveFile": "archive_Local_Suite_Instance-67e7358b-077b-420f-9e04-e9b9e8374b68.zip",
   "importActivateMode": "ImportActivate",
   // options are "ImportOnly" || "ActivateOnly" || "ImportActivate"
   "storageInfo": {
      "storageUrl": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo",
      "storageUser": "myemail@company.com",
      "storagePassword": "generated_token"
   }
}
```
For example, use a `postman` or `curl` (as shown below) command to import the archive of design time objects into Oracle Integration. You can set `importActivateMode` to the following values:

- **ImportOnly**: Imports, but does not activate, integrations.
- **ActivateOnly**: Activates previously imported integrations. This enables you to update connection parameters before activating integrations.
- **ImportActivate**: Imports and activates integrations.

```bash
curl -k -v -H "Content-Type: application/json" -X POST -d '{"archiveFile": "archive_Local_Suite_Instance-67e7358b-077b-420f-9e04-e9b9e8374b68.zip", "importActivateMode": "ImportOnly", "storageInfo": { "storageUrl": "https://swiftobjectstorage.us-region-1.oraclecloud.com/v1/paasdevoic/cloneRepo", "storageUser": "myemail@company.com", "storagePassword": "generated_token" }}' -u admin:password https://host/ic/api/common/v1/importServiceInstanceArchive
```

Possible sample output from this command is as follows:

```
{
  "jobId": "554",
  "status": "NOT_STARTED"
}
```

**Check the Integrations Design-Time Metadata Import Status**

Verify the design-time metadata import status after completing the import task.

1. Check the status of the import operation using a `postman` or `curl` (as shown below) command:

```bash
curl -k -v -X GET -u admin:password https://host/ic/api/common/v1/importServiceInstanceArchive/554
```

Possible sample output from this command is as follows. Other potential output includes **RUNNING**, **COMPLETED**, and **FAILED**.

```
{
  "jobId": "5108",
  "overallStatus": "RUNNING",
  "componentStatus": [
    { "component": "Integration" },
  ]
}
```
2. Log in to your Oracle Integration instance.
3. Browse the pages and note that the design-time metadata you exported from your Oracle Integration Classic instance are now visible. For example, for Integrations, look for integrations, connections, lookups, and more.

Migrate Processes Design-Time Metadata into Oracle Integration

As an administrator, you can import your Processes design-time metadata (process applications and decision models) into Oracle Integration. The import tool provides migration flexibility: choose to migrate the entire instance, a selected space, or individual items.

- Import Processes Design-Time Metadata into Oracle Integration
- Import Using the Import Command Line Tool

Import Processes Design-Time Metadata into Oracle Integration

Use the Process import tool to automatically move Process and decision design-time metadata from one instance to another. The tool moves process applications and active decision models from a selected source (an Oracle Integration instance) to a selected Oracle Integration destination, and activates the decision models.

Alternatively, you can import process applications and decision models from the command line instead of the user interface. See Import Using the Import Command Line Tool. Note that using the import tool is highly recommended.

⚠️ Important:

You can specify an import scope, which allows you to import all Process assets for an instance, for a space, or a specific process application or decision model. Be sure not to import assets more than once, because doing so can lead to unpredictable results. Avoid refreshing or closing the tool during the import.
1. On the Oracle Integration Home page, click **Processes** in the navigation pane.

2. Click **Settings** in the navigation pane, then **Import** from the top options on the Administration page.

3. On the Import Administration page, specify a source for the import.
   a. In the **Identify the instance to import from** fields, choose an Oracle Integration instance.
   b. In the **Host Name** field, enter the host, using the format `https://host:port`.
   c. Enter a user name and password to sign in to the instance as an administrator.

4. Identify a destination for the import.
   - To import to your current instance, skip the **Import to the current Oracle Integration instance** field.
   - To import to another Oracle Integration instance than your current instance, click **Edit** and complete the host, user name, and password fields for the instance you want to import to.
   - To select importing to the current instance, click **Reset**.

5. In the **Scope** field, identify what you want to import.
   - Choose **Entire Instance** to import all process applications and decision models from all spaces located on the instance.
   - Choose **Space** to import a selected source only. Select the space in the **Source Space** field that displays.
   - Choose **Process Application or Decision Model** to import a selected application or decision model only. In the additional fields that display, select the source space, process application or decision model to import, and the destination space.

6. Click **Import**.

7. Review the import log.

   **Note:**
   
   You must check the logs and make sure there are no errors before proceeding with post migration steps.

   Click **Download Import Log** in the import dialog to download a zip file containing the import log. It lists any errors that might have occurred along with a summary of the number of spaces or items (process applications or decision models) imported.

8. In Oracle Integration, change space settings as needed.

   Return to the design-time Administration screen, and share the space(s) you migrated to Oracle Integration and change their permissions.

### Import Using the Import Command Line Tool

Optionally, use an import command line tool to automatically move Process and decision design-time metadata from one instance to another instead of the user
interface. (The user interface tool is recommended.) The tool moves process applications and active decision models from a selected source (an Oracle Integration instance) to a selected Oracle Integration destination, and activates the decision models.

**Note:**

You can specify an import scope, which allows you to import all Process assets for an instance, for a space, or a specific process application or decision model. Be sure not to import assets more than once, because doing so can lead to unpredictable results.

**Note:**

The import utility requires Java version 8 or later. In addition, both the source and target instances must return a ping response.

1. On the Oracle Integration Home page, click **Processes** in the navigation pane.
2. Click **Settings** in the navigation pane, then **Import** on the Administration page.
3. Click the **Download Utility** button and save it to a selected location.
4. Sign in to the Oracle Integration source and target environments.
5. Run the ImportTool utility from the command line. Use arguments to import an entire instance, one or more of its spaces, or one or more process applications or decision models.

**Format**

```
$java -jar ImportTool.jar srcType=PCS srcHost=http://host:port srcUser=user srcPass=password oicHost=http://host:port oicUser=user oicPass=password scope=scope
```

**Example: Import the entire Oracle Integration instance**

```
$java -jar ImportTool.jar srcType=PCS srcHost=http://abc01xyz.example.com:7001 srcUser=user1 srcPass=password1 oicHost=http://def02uvw.example.com:7001 oicUser=user1 oicPass=password1 scope=SPACE srcSpace="123456789"
```

**Example: Import process applications (delimit items by pipe line)**

```
$java -jar ImportTool.jar srcType=PCS srcHost=http://abc01xyz.example.com:7001 srcUser=user1 srcPass=password1 oicHost=http://def02uvw.example.com:7001 oicUser=user1 oicPass=password1 scope=PROJECT srcSpace="132457689" srcProject="Loan Application|Travel Application" oicSpace="123456789"
```

6. Review the import log.
Note:

You must check the logs and make sure there are no errors before proceeding with post migration steps.

A zip file containing the import log is created in the SRC_TO_OIC folder. It lists any errors that might have occurred along with a summary of the number of spaces or items (process applications or decision models) imported.

7. In Oracle Integration, change space settings as needed.

Return to the design-time Administration screen, and share the space(s) you migrated to Oracle Integration and change their permissions.

Migrate Your Visual Builder Design-Time Metadata

You can migrate a visual application from an Oracle Integration Classic instance by importing the application as an archive or by importing it from a Git repository.

The credentials for services used by the application are not imported. You need to supply the credentials after creating the new application from the imported sources. Similarly, the entries for mobile build configurations, such as iOS provisioning profiles, keystores, and passwords are not imported. You need to re-enter these details in the new instance.

Export Visual Builder Design-Time Metadata

In your Oracle Integration Classic instance, use the Export action to create a ZIP archive of the visual application you want to migrate to the new instance on Oracle Cloud Infrastructure.

When you export the application you can choose if you want the archive to include the development data contained in the application’s custom business objects. Some information, such as credentials for external REST end points, is removed when you export an application. This information needs to be provided after the archive is imported.

To export a visual application and its custom business objects:

1. Open the Visual Builder Home page of your Oracle Integration Classic instance.

2. On the Visual Applications home page, open the Application Options menu for the application version you want to export and select Export.

   If there are multiple versions of an application you must use the Options menu of the version that you want to export.

   Alternatively, when a visual application is open, you can choose Export in the application’s options menu in the toolbar.

3. Click Export with Data in the Export Application dialog box.

   When you choose to export the application with data, the archive will include a json file (entity.json) and spreadsheet (entity-data.csv) for each custom business object. The json file describe the business object and the spreadsheet contains the business object data in the development database. If you choose to
export the application without data, the archive will only contain the json file
describing the business objects.

The archive will always include the data for any business objects that are identified
as containing Application Setup Data.

The visual application and its resources are exported as an archive file. The archive is
saved to your local system in the location specified for your browser’s downloads.

Import Visual Builder Design-Time Metadata

In your new Oracle Integration instance on Oracle Cloud Infrastructure, you create a
new visual application by importing the archive of the visual application you exported
from the Oracle Integration Classic instance.

To import a visual application archive:

1. Open the Visual Builder Home page on Oracle Cloud Infrastructure.
2. Click Import on the Visual Applications home page.
3. Click Import from file in the Import Application dialog box.
4. Drag your visual application archive file on your local system into the dialog box.
   Alternatively, click the upload area in the dialog box and use the file browser to
   locate the archive on your local system.
5. Enter a valid application name and ID in the dialog. Click Import.

After you import a visual application, you might need to provide additional details such
as credential to service in the new copy of the application. For example, if the
application you are importing contains Process definitions, you will be prompted to
specify an existing Process application that contains all the process definitions
required by the application you are importing.

Import Visual Builder Design-Time Metadata from a Git Repository

If your visual application's sources are stored in a Git repository hosted on Oracle
Developer Cloud Service, you can import the application by configuring the connection
to the Git repository and creating the application from the imported sources.

To import a visual application from a Git repository:

1. Open the Oracle Visual Builder instance on Oracle Cloud Infrastructure.
2. Click Import on the Visual Applications home page.
3. Click Application from GIT in the Import dialog box.
4. Click Add Credentials in the Import Application from GIT dialog box.
5. Enter the URL, username and password for your Oracle Developer Cloud Service
   account. Click Save Credentials.
   After your credentials are checked and saved, the new credentials are added to
   the list in the Configure DevCS Credentials dialog box.
6. Select the URL of your account in the DevCS URL with Credentials field.
7. Select the project, repository and branch of the application's sources.
8. Enter a valid application name and ID. Click Import.
After you import a visual application, you might need to provide additional details such as credential to service in the new copy of the application. For example, if the application you are importing contains Process definitions, you will be prompted to specify an existing Process application that contains all the process definitions required by the application you are importing.
Complete the Post-Migration Tasks

After successfully migrating your Oracle Integration Classic instances to Oracle Cloud Infrastructure, test your integrations thoroughly, and then perform cleanup and other optional configuration tasks.

Topics:

- Verify Your Environment and Activate Your Integrations
- Complete Post-Migration Tasks for Processes
- Complete Post-Migration Tasks for Visual Builder Applications

Verify Your Environment and Activate Your Integrations

After Integrations migration to Oracle Integration is complete, perform the following post-migration tasks.

1. Verify your integrations (for example, check the connections, lookups, and more).
2. Activate your integrations (if you did not do so automatically during the import process).

Complete Post-Migration Tasks for Processes

After verifying and activating integrations, complete Processes tasks.

Topics:

- Perform Post Import Steps for Processes
- Handle In-Flight Process Instances
- Follow Migration Best Practices for Processes

Perform Post Import Steps for Processes

After import, perform additional Processes steps as needed.

1. In Oracle Cloud Infrastructure, manually reconfigure any federated SSO configurations for Processes that were configured in Oracle Integration Classic.
2. If you configured OAuth to work in Oracle Integration Classic for Processes, manually reconfigure it in Oracle Cloud Infrastructure. See Security, Authentication, and Authorization in REST API for Oracle Integration.
3. In runtime administration, configure supporting services and settings, as needed.

On the Oracle Integration Home page, select My Tasks in the navigation pane, then Workspace. In the My Tasks navigation pane, select Administration.
4. **IMPORTANT:** Under **Archive and Purge**, configure purge and archive settings for process instances and analytics data.

Be sure to verify the retention policies set to purge unused information. Set the **Purge Retention (Days)** setting on the **Schedule Instances Archive** tab to the number of days to retain completed process instances before they get purged. (Purged instances cannot be recovered.) Because increasing retention days increases database consumption, you must balance your organization’s retention needs with database capacity. See Archive and Purge Data in *Using Processes in Oracle Integration*.

5. Under **Services (Platform)** tab, configure Oracle Content and Experience settings if integrating documents in process applications. See How do I integrate with Oracle Content and Experience Cloud? in *Using Processes in Oracle Integration*.

6. Under **Services (Infrastructure)** tab, configure Oracle Cloud Storage settings and click **Test**.

Use values from earlier compartment and object storage configurations. See Create an Object Storage Bucket.

   a. In the **URL** field, enter the URL constructed as follows:

      ```
      https://swiftobjectstorage.region.oraclecloud.com/v1/tenancy
      ```

      For example:

      ```
      https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/tenancy
      ```

   b. In the **Container Name** field, enter the storage bucket name.

c. In the **User** field, enter your administrator username (my.email@oracle.com)

d. In the **Password** field, enter the OAuth token you generated when creating the storage bucket.

7. Under **Services (Infrastructure)** tab, configure the Oracle Notification Service for email notifications. This service must be configured for users to receive email notifications. See Enable Email Notifications.

8. Open imported decision models used by process applications. Decision models are automatically activated during import.

   Navigate to **Decisions** and update the references.

9. Activate imported integration flows if used by process applications. Note that integrations must be activated.

   In Oracle Integration, previously separate processes and integrations now reside in the same environment. After migration, you may need to reconfigure the integrations and invocations from both Processes and Integrations, as both support direct interaction with each other from within Oracle Integration.

   a. On the Oracle Integration Home page, select **Process Applications** in the navigation pane, open an imported application, and select **Integrations** in the navigation pane.

   b. Select an integration to update, click **Edit** and update as needed.

   c. Update the security for the called service. Click the **Security** tab, then the **Keystore Credential** and update as needed. Set or import credentials/certificates for each REST or SOAP connector used in your process applications.
d. Complete changes and click **Update**, and save.

10. Activate imported process applications. See Activate Applications in *Using Processes in Oracle Integration*. You can also use the Processes REST APIs that allow you to activate process applications.

    Previous runtime instance data is not migrated. See **Handle In-Flight Process Instances**.

11. Remap Processes user roles (swim lanes) for all process applications.

    Process role mappings are not migrated. Reconfigure role mappings for each process application under **Manage Roles** in Workspace Administration. Note that users must be migrated to Oracle Integration before they can be mapped to roles. See **Assign and Manage Roles in Using Processes in Oracle Integration**.

### Handle In-Flight Process Instances

After import, follow guidelines to handle currently running process instances.

**Handling short lived instances and their human tasks**

- For a short overlapping period, run the instances in parallel in both environments, then schedule a transition period during which no new instances are created in the classic environment.
  - Let classic environment instances and tasks complete and drain, with no new instances created. At the same time, ensure all new instances and tasks get run on the Oracle Cloud Infrastructure instance. During this transition period, end users must use two inboxes.
  - After the transition period, all process runtime activity ends on the classic environment. End users no longer need to use two inboxes.

**Handling long lived instances and their human tasks**

- For a short overlapping period, run the instances in parallel in both environments, then terminate instances in the classic environment and manually recreate them in the Oracle Cloud Infrastructure environment.
  - Let classic environment instances and tasks complete and drain, with no new instances created. At the same time, ensure all new instances and tasks get run on the Oracle Cloud Infrastructure instance. During this transition period, end users must use two inboxes during the transition period.
  - After the transition period, use the **Alter Flow** option in the classic environment to terminate instances. Create new instances with the same payload as in the classic environment in the Oracle Cloud Infrastructure environment. Use **Alter Flow** in Processes on the Oracle Cloud Infrastructure to move newly created replicas of the classic environment instances to the correct activity in the Oracle Cloud Infrastructure process. You can automate these actions by programmatically invoking the **Alter Flow** option in the classic environment and the Oracle Cloud Infrastructure environment. See **Alter the Flow of a Process Instance in Using Processes in Oracle Integration**.
Follow Migration Best Practices for Processes

Follow best practices whenever possible after migration to Oracle Integration.

- Wrap all rules and policy logic into decision models rather than Oracle Business Rules.
- While you are able to create integrations using REST and SOAP connectors in process applications, it is recommended that you create all integrations in the OIC Integrations area instead. From there, you can easily select them for use in structured and dynamic processes. This practice centralizes all integration efforts in the OIC Integrations area.
- Among many updates, Oracle Integration provides a new Task List, accessed on the Home page by choosing My Tasks. It provides a lighter weight interface with better performance.
  - My Tasks works best for end users
  - Workspace works best for advanced users for back office and tracking purposes

Complete Post-Migration Tasks for Visual Builder Applications

Any applications on the Oracle Integration Classic instance that are live will need to be staged and published again after they are migrated to the new instance on Oracle Cloud Infrastructure.

Developers can lock a live visual application to prevent changes to live application data, and then import the live data to the new version of the application on Oracle Cloud Infrastructure. After the live data is imported and the application is published, users can be instructed to use the new version of the live app. For users who access a web application, you’ll need to communicate the new URL that replaces the URL they used previously to access their web applications. For mobile applications, you’ll need to rebuild and republish your mobile apps to the app store(s) where you initially published it or use whatever mechanism you used previously to distribute the mobile app. Do this so users can update the mobile app on their device to a version of the app that was built using Oracle Visual Builder on Oracle Cloud Infrastructure.

Lock the Live Visual Builder Application

If you are migrating an application that is a live application, you can lock the live application to prevent any users from using the application while you migrate it to your new instance.

To lock an application:

2. On the Visual Applications home page, open the Application Options menu for the live application you want to lock and select Lock.
3. Click Lock in the Confirm Lock Application dialog box.

On the Visual Applications home page, the status of the app is now Live Locked.
Users are not able to use the app to edit data while it is locked.

**Export Application Data From the Live Database**

You can export all the data contained in your database as CSV files. The export tool creates one CSV file for each of the custom business objects in your database and packages the files as a ZIP archive.

To export the database data as a CSV file:

2. On the Visual Applications home page, open the visual application and open the Business Objects pane in the Navigator.
3. Click the **Options** menu in the Business Objects pane and select **Data Manager**.
4. Select the Live database in the dropdown list.
5. Click **Export All Data** to download a ZIP archive that contains CSV files with the data.

You will import the zip archive into the Live database of the published app on the new instance on Oracle Cloud Infrastructure.

**Import the Live Application Data into the New Application**

You can populate the database of your new visual app with the live data exported from your application on the Oracle Integration Classic instance.

You can import the archive containing the Live data into the Development, Staging or Live database of your new application, and then include the data when you stage or publish the app. For example, if your app is staged but not yet published, you can import the Live data into the Staging database and then publish the app, making sure that you select Include data from Stage in the Publish Application dialog box.

To import Live data into your application's database:

1. Open the new Oracle Visual Builder instance on Oracle Cloud Infrastructure
2. On the Visual Applications home page, open your visual application and open the Business Objects pane in the Navigator.
3. Click the **Options** menu in the Business Objects pane and select **Data Manager**.
4. In the Database dropdown list, select the database that you want to import the Live data into.
   
   The options available in the Database dropdown list depend upon if you have staged or published your application. If your application has not been staged, you can only select Development in the dropdown list.
5. Click **Import from file**.
6. Drag the archive with the CSV files into the drop area in the Confirm Import Data dialog box. Click **Import**.
7. In the Import from File dialog box, confirm that the data was imported successfully. Click **OK**.

You can confirm that the data was imported successfully in the Business Objects editor.
Publish the New Application with the Live Application Data

You can publish a staged version of your application from the Home page or from the main menu. After you publish a version of an app it is read-only and can no longer be changed. To make changes to update the app you need to create a new version.

To publish the application:

1. Open the new Oracle Visual Builder instance on Oracle Cloud Infrastructure
2. On the Visual Applications home page, open the Application Options menu for the application version you want to publish and select Publish.
3. For web applications, communicate the URL that end users must now use to access the web application(s) that you have published.
4. For mobile applications, update the app store with an instance of the app that you have just published so end users can update their installation with the newer version of the app that has been built using Oracle Cloud Infrastructure.
Migrate Integrations and Processes Together

You can use `opcmigrate` to migrate any Integrations and Processes design-time metadata in your instance. `opcmigrate` is preinstalled with your Oracle Cloud Infrastructure Compute Classic instance, enabling you to use this tool right away. A user that is an administrator of the tenancy (in the Administrators group) can perform the migration.

**Note:**

Visual Builder design-time metadata cannot be migrated with `opcmigrate`.

Topics:

- Perform Oracle Cloud Infrastructure Prerequisites
- Create an Oracle Cloud Infrastructure Compute Classic Instance
- Configure the Oracle Cloud Infrastructure Compute Classic Instance
- Update the Default Profile for Your Oracle Cloud Infrastructure Classic Services
- Update the Oracle Cloud Infrastructure Configuration File
- Learn About Available Commands
- Discover Instances in Your Source Environment
- List Your Oracle Integration Instances
- Export Design-Time Metadata
- Create the Target Instance on Oracle Cloud Infrastructure
- Edit the Target Configuration File
- Import Design-Time Metadata

**Note:**

If you want to run `opcmigrate` on any other system, follow the procedure to install the tool on any system running Oracle Linux 7.x, Windows, or MacOS. Note that not all features of this tool are available when you download and install it on other systems. See Install Oracle Cloud Infrastructure Classic Discovery and Translation Tool of Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure.
Perform Oracle Cloud Infrastructure Prerequisites

Before you create an Oracle Integration instance in an Oracle Cloud Infrastructure region, you must prepare for migration and create the required infrastructure.

- Plan and prepare for the migration process. See Prepare to Migrate Oracle Integration Classic to Oracle Cloud Infrastructure.
- Create the following Oracle Cloud Infrastructure resources if they do not already exist:
  - A compartment
  - A storage bucket
  - A user and group to which you assign the user to perform the migration.
  - Policy statements that enable you to perform the migration.
    See Create a Cloud Storage Compartment, User, Group, and Policies and Create an Object Storage Bucket.

Create an Oracle Cloud Infrastructure Compute Classic Instance

Using your Oracle Cloud Infrastructure Compute Classic account, create the source controller (also known as Control-S) instance with the following configurations. This action creates an Oracle Cloud Infrastructure Compute Classic image that includes the \texttt{opcmigrate} tool that you use to migrate Oracle Integration to Oracle Cloud Infrastructure.

\textbf{Note:}

You are not charged for this Oracle Cloud Infrastructure Compute Classic instance as long as you create resources in the \texttt{/oraclemigration} container. Steps for performing this task are described in Step 7.

1. Log in to the My Services Console.
2. From the \textbar menu, select \texttt{Services > Compute Classic}, then select your region.
3. On the Compute Classic page, click \texttt{Create Instance}.
4. In the upper right corner of the page, click \texttt{Customize}.
5. Click \texttt{Oracle Images}.
6. Enter \texttt{Migration} in the search field, and click the \texttt{Search} icon.
   The Control-S instance must be created in the same identity domain and site as the resources that you want to migrate. You can use the web console or any other interface to create an instance with the following specifications:
   - \textbf{Image:} Select \texttt{OL_7.5_UEKR4_x86_64_MIGRATION}. This image includes the \texttt{opcmigrate} tool.
Shape: Select General Purpose oc5 (4 OCPUs, 30 GB RAM) or any other shape with a sufficient number of OCPUs.

SSH Key: Associate an SSH public key with the Control-S instance. You'll use the corresponding private key to connect to the Control-S instance. Note that this key isn't the same as the SSH key pair used to access Linux source instances from Control-S when you migrate instances and block storage.

Network: Ensure that you select the shared network with a persistent public IP address.
- Select the default security list that allows SSH inbound.
- Ensure that security rules are in place to allow HTTPS outbound traffic.

Storage: Use a boot volume of 200 GB or larger.

Note:

In Oracle Cloud Infrastructure Compute Classic instances, block storage is limited to 2 TB per block volume. To extend, use LVM with multiple 2TB block volumes. Ensure that you format and mount this additional storage under /images.

7. The Control-S instance and associated storage volumes created for migration are by default billed at the applicable rates for your account. However, you can rename these resources so that the multipart name includes /oraclemigration as a container. Resources created in this /oraclemigration container are not billed to your account.

If you create the Control-S instance using the API, CLI, or a Terraform configuration, you can specify multipart resource names as /Compute-example/user@example.com/oraclemigration/resource-name when you create the resources.

If you create the Control-S instance and storage volumes using the web console, after the instance is created, modify the orchestration to move the instance and storage volumes to the /oraclemigration container.

To move the Control-S instance and storage volumes into the /oraclemigration container:

a. Log in to the Oracle Cloud Infrastructure Compute Classic web console and go to the Instances page.
b. When the Control-S instance is created with status **Running**, click the **Orchestrations** tab.

c. To move the Control-S instance to the /oraclemigration container, you can suspend the orchestration. Go to the relevant orchestration and from the ⬆️ menu, and select **Suspend**.

d. After the orchestration status changes to **Suspended**, from the ⬆️ menu, select **Update**.

e. On the update page, in the Instance section, click the ⬆️ menu and select **Edit JSON**.

f. In the Edit Orchestration Object JSON window, look for the instance name. This is usually displayed within the **template** section, after networking.

```
"name": "/Compute-example/user@example.com/instance-name",
```

Modify the instance name to include the /oraclemigration container:

```
"name": "/Compute-example/user@example.com/oraclemigration/instance-name",
```

Click **Update**.

g. To move storage volumes to the /oraclemigration container, you must terminate the orchestration. This step destroys all persistent and nonpersistent objects created by the orchestration. Do this only if you have not made any changes to your instance or storage volumes that you want to preserve. On
the Orchestrations page, go to the relevant orchestration, and from the menu, select **Terminate**.

**h.** After the orchestration status changes to **Stopped**, from the menu, select **Update**.

**i.** On the update page, in the Storage Volume section, go to the relevant storage volume, click the menu and select **Edit JSON**.

**j.** In the Edit Orchestration Object JSON window, look for the storage volume name in the template section:

```
"name": "/Compute-example/user@example.com/storage-volume-name",
```

Modify the instance name to include the **/oraclemigration** container:

```
"name": "/Compute-example/user@example.com/oraclemigration/storage-volume-name",
```

Click **Update**.

**k.** Repeat these steps for any other storage volume in this orchestration that you want to move to the **/oraclemigration** container.

**l.** When you have updated all the relevant resources, start the orchestration. On the Orchestrations page, go to the relevant orchestration and from the menu, select **Start**.
When your instance and other resources are created in the /oraclemigration container, they are listed in the web console with this container name prefixed to the user-specified name. Therefore, if you had named your Control-S instance Control-S, it now appears with the name oraclemigration/Control-S.

When the instance status is displayed as Running, you can log in to the instance as the opc user from your local system using your SSH private key.

**Configure the Oracle Cloud Infrastructure Compute Classic Instance**

All of the tools required for migration are installed on the host, but additional configuration is required to provide details about the source and target environments.

Remember that a single Oracle Cloud Infrastructure Compute Classic (also known as Control-S) instance must be used to migrate resources from a single specified Oracle Cloud Infrastructure Compute Classic account and site to a single specified Oracle Cloud Infrastructure tenancy, region, and availability domain.

All of the configuration settings are in a file called secret.yml. You can use the sample file available at /home/opc/ansible/secret.yml.sample to create your secret.yml file.

1. Enter the details of your Oracle Cloud Infrastructure Compute Classic account and your Oracle Cloud Infrastructure OCIDs.

   Here is an example of a secret.yml file, with sample values and information about each field. For help on YAML syntax, see https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html.

```yaml
# OCI info
compartment_id: ocid1.compartment.oc1..aaaaaaaa...
user_id: ocid1.user.oc1..aaaaaaaa...
fingerprint: a0:a0:a0:a0:a0...
tenancy_id: ocid1.tenancy.oc1..aaaaaaaa...
region: us-ashburn-1
availability_domain: kWVD:US-ASHBURN-AD-3

# version and shape used to the Control-T instance
# 'Oracle Linux' is the only supported operating_system
oracle_linux_version: '7.6'
shape: 'VM.Standard2.1'

# subnet must be from the availability_domain you specified
subnet_id: ocid1.subnet.oc1.iad.aaaaaaaa...
# optional passphrase if used for OCI PEM file
pass_phrase:
# The ocic_oci_sig_par should point to the PAR (pre-authorized request)
# for the ocic-oci-sig
# bucket in the OCI object storage. The write-only PAR allows the
```
system to write a signal
# object to the ocic-oci-sig when it is launched in the OCI side
# indicating it is ready to
# handle storage attachments for the volumes post migration.
ocic_oci_sig_par: PAR URL HERE

# OCI-Classic info
opc_profile_endpoint: compute.uscom-central-1.oraclecloud.com # or
another one
opc_password:
container: /Compute-tenancy/user@email.com

# OCI-C Object Storage Classic REST Endpoint, user name, password.
# refer to https://docs.oracle.com/en/cloud/iaas/storage-cloud/ssapi/
SendRequests.html
# for instructions how to find the endpoint.
# If you comment them away (or remove them) from this secret.yml, then
we will take
# your tenancy/opc_password given above as the username/password of
Object Storage;
# and the endpoint of Object Storage is default to:
# https://tenancy.storage.oraclecloud.com/v1/Storage-tenancy

# Object Storage Classic REST Endpoint, username, password.
opc_object_storage_endpoint:
opc_object_storage_username:
opc_object_storage_password:

# Control-S Instance settings
targetControllerAvailableStorageInGB: 2048

Note:
If you are migrating an Oracle Integration instance, you do not need to
provide the credentials and REST endpoint for Oracle Cloud
Infrastructure Object Storage Classic. That information is required when
you want to migrate VMs and block storage using the Oracle Cloud
Infrastructure Classic VM and Block Storage Migration Tool.

The secret.yml file contains sensitive information about your account.

2. Modify permissions on this file to restrict access.

   chmod 600 /home/opc/ansible/secret.yml

3. Apply the configuration to the system. This creates the required files /home/
opc/.opc/profiles/default and /home/opc/.oci/config.

   opcmmigrate migrate instance service setup

4. Copy the PEM key required for the API connection to the file /home/opc/.oci/
oci_api_key.pem on Control-S.
5. Modify permissions on the key file to restrict access.

   chmod 600 /home/opc/.oci/oci_api_key.pem

6. Ensure that the SSH key pair required for accessing your instance is available on
   the Control-S instance. You need the SSH private key to access the source
   instance when you export the configuration and access the target instance when
   you import the configuration.

7. Modify permissions on the key file to restrict access.

Update the Default Profile for Your Oracle Cloud
Infrastructure Classic Services

The opc migrate tool connects to your Oracle Integration source environment using
connection information that you provide in a profile file.

The information you provide in the profile file includes the user name or identity for
each service in the source environment and the service endpoint and region. If you
want to run the tool in multiple regions or tenancies, you can create separate profile
files for each region and tenancy.

1. Look up service-specific details in your Oracle Cloud Dashboard. You need the
   user name and API endpoint for each service.

2. Create the directory for the profile file, if the directory does not already exist. If you
   configured your Control-S instance by running the command opc migrate instance service setup,
   the default file has already been created for you based
   on the input you provided in the secret.yml file. If you have not run that
   command, create the required directory and profile file now.

3. Use the following template to create or update your profile file.

4. Save this profile with the file name default in the path ~/.opc/profiles.

5. Replace the sample values with values specific to each service. Note that when
   you migrate Oracle Integration instances, you can retain only the "compute" and
   "paas" sections of this file and you can remove other sections, such as
   "database", "lbaas", and "object_storage".

```json
{
   "global": {
      "format": "text",
      "debug-request": false
   },
   "compute": {
      "user": "/Compute-example/user@example.com",
      "endpoint": "compute.uscom-central-1.oraclecloud.com"
   },
   "paas": {
      "user": "user@example.com",
      "identity_id": "idcs-00000000000000000000000000000000",
      "endpoint": "psm.us.oraclecloud.com",
      "region": "uscom-central-1"
   }
}
```
a. Go to the Dashboard page in the My Services Console and find the **Classic Compute** box.

b. From the hamburger menu in the lower right, select **View Details**.

c. On the **Overview** tab, find the values for the following to copy to the default profile.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
</tr>
</thead>
</table>
| compute | • user: Copy the value from the **Service Instance ID** field (for this example, 111111111) and the administration user for the account, and assemble the value as follows.  

/Compute-111111111/firstname.lastname@example.com  

• endpoint: Copy the value from the **REST Endpoint** field. |
| paas    | • user: Enter the administration user for the account.  

• identity_id: Copy the value from the **Identity Service Id** field.  

• endpoint: Enter the endpoint for your instance.  

• region: Enter the region in which your instance is installed. |

6. Add the "oic" section in this file. Provide the admin user credentials and storage container details. Each "ServiceName" subsection within the "oic" section is for a single instance. For example, if you have three instances on your host that you want to migrate, you must create three separate "ServiceName" subsections.

```json
{
  "oic": {
    "ServiceName": {
      "admin_user": "user@example.com",
      "admin_password": "password",
      "storage_info": {
        "storageUser": "myemail@example.com",
        "storagePassword": "generated_token",
        "storageUrl": "https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/oraclemigration"
      }
    }
  }
}
```
a. Go to the Dashboard page in the My Services Console and find the **Storage Classic** box.

b. From the hamburger menu in the lower right, select **View Details**

c. On the **Overview** tab, find the values for the following to copy to the default profile.

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceName</td>
<td>Replace <code>ServiceName</code> with the specific instance name to migrate. You identify the instance names available for migration by running the <code>list</code> command. See List Your Oracle Integration Instances.</td>
</tr>
<tr>
<td></td>
<td>admin_user: Enter the service administration user for that particular instance. This is <em>not</em> the account administrator.</td>
</tr>
<tr>
<td></td>
<td>admin_password: Enter the password.</td>
</tr>
</tbody>
</table>

| storage_info   | storageUser: Enter the storage container user name.                       |
|                | storagePassword: Enter the password.                                      |
|                | storageUrl: Enter the storage container URL.                              |

7. If you create multiple profiles in the `~/.opc/profiles` directory, use the `--profile` option to specify the profile you want to use when you run the tool. If no profile is specified, the default profile is used.

Passwords are not specified in the profile file for security reasons. You are prompted to provide the password for services when you run the tool except for those defined in the `ServiceName` section.

**Update the Oracle Cloud Infrastructure Configuration File**

When you set up your Oracle Cloud Infrastructure Compute Classic image (Control-S) instance, the Oracle Cloud Infrastructure configuration file is created with information you supplied in the `secret.yml` file.

If you have not run the command to set up the Control-S instance, you can do so now. After the command completes, view and edit the file to ensure it has the appropriate values. Alternatively, you can manually create the required configuration file, `/home/opc/.oci/config`.

Here's an example of a `config` file with sample values:

```
[DEFAULT]
user=ocid1.user.oc1..aaaaaaaa...
fingerprint=81:45:...
key_file=/home/opc/.oci/oci_api_key.pem
pass_phrase=
tenancy=ocid1.tenancy.oc1..aaaa...
region=us-ashburn-1
```
Learn About Available Commands

You can obtain help on using `opcmigrate` commands.

**Note:** Additional information about various commands and options to specify when creating reports and filtering the output are provided. See Learn About Commonly Used Commands and Options in Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure.

1. Enter the following command to return a list of all available commands and general options.

   
   ```bash
   opcmigrate --help
   ```

2. Enter the following command to return a list of cloud services that can be migrated with this tool (including Oracle Integration).

   ```bash
   opcmigrate migrate --help
   ```

   The following output is returned.

   ```
   {rsm, oic, soacs, jcs, instance, database}
   ```

3. Enter the following command to return a list of options specific to discovering instance resources:

   ```bash
   opcmigrate discover --help
   ```

   usage: opcmigrate discover [-h] [-c [CONTAINER [CONTAINER ...]]] [-i ID] [-z ZONE] [-w] [--insecure]

   optional arguments:
   -h, --help show this help message and exit
   -c [CONTAINER [CONTAINER ...]], --containers [CONTAINER [CONTAINER ...]]
     one of more container paths in the format '/Compute-<domainid>/<path>', if not set the root path '/Compute-<domainid>' is used.
   -i ID, --id ID service id profile override
   -z ZONE, --zone ZONE zone name profile override
   -w, --with-storage-objects
     discover object names
   --insecure skip validation of endpoint SSL certificates

4. Enter the following command to return a list of options specific to migrating Oracle Integration.

   ```bash
   opcmigrate migrate oic --help
   ```
Oracle Integration Classic (OIC) Migration Commands:
{list,export,import,create}

list      List Oracle Integration Classic resources available for export.
export    Export Oracle Integration Classic resources to an archive in OCI Object Storage.
import    Import OIC resources from an archive in OCI Object Storage to an OCI target environment.
create    Provision Oracle Integration Service Instance in OCI.

5. Enter the following commands to return a list of options specific to each command you specify during the migration process. For example:

opcmigrate migrate oic create -h

opcmigrate migrate oic import -h

Discover Instances in Your Source Environment

To discover all instances available for migration, log in to the Control-S instance and run the following command.

Discover all the instances, including Oracle Integration instance resources. When prompted, enter the account administrator password for the entire customer account (not the service administration password).

opcmigrate discover

The following output is returned.

2019-07-15T15:25:30 INFO Authenticating with OCI Classic Compute API
2019-07-15T15:25:30 INFO Discovering resources for "000000000".
...
2019-07-15T15:25:51 INFO Storing discovered resources to 'resource-default.json'
To specify a profile other than `default`, run:

```
opcmigrate --profile profile_name discover
```

The profile name is included in the file name of reports generated by the `opcmigrate discover` command.

To specify a profile created in a directory other than `~/.opc/profiles`, provide the full path to the profile location with the `--profile-directory` option.

```
opcmigrate --profile-directory path_to_profile_file discover
```

### List Your Oracle Integration Instances

To list the instances in the source environment, run the following command.

List the Oracle Integration Classic discovered instances and their states when the instances were discovered (this may differ from their current states).

```
opcmigrate migrate oic list
```

The following output is returned.

```
Oracle Integration Classic Service Instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>myinstance1</td>
<td>1.0</td>
<td>READY</td>
<td></td>
</tr>
<tr>
<td>myinstance2</td>
<td>1.0</td>
<td>STOPPED</td>
<td></td>
</tr>
<tr>
<td>myinstance3</td>
<td>1.0</td>
<td>READY</td>
<td></td>
</tr>
</tbody>
</table>
```

This command uses the output generated by the `opcmigrate discover` command to identify and list the available Oracle Integration instances.

### Export Design-Time Metadata

To create an archive file of a specified instance, run the `export` command to export the design-time metadata from the instance.

You can only specify a single instance at a time as the value for the `ServiceName` parameter. This action exports all Integrations and Processes design-time metadata available in that instance to an archive file. The archive file is uploaded to the object
storage bucket you specified with "storageUrl" in the "ServiceName" section of the default profile in use.

**Note:**

Ensure that you replace the ServiceName parameter in the default profile with the actual instance name to export. You specify the ServiceName value when executing the export command. Otherwise, the export command fails. For example, change:

```json
{
   "oic": {
      "ServiceName": {
         "admin_user": "user@example.com",
         "admin_password": "password",
         ...
      }
   }
}
```
to:

```json
{
   "oic": {
      "myinstance1": {
         "admin_user": "user@example.com",
         "admin_password": "password",
         ...
      }
   }
}
```

See Update the Default Profile for Your Oracle Cloud Infrastructure Classic Services.

There are several ways to capture the content of the export command, in particular, the archive file name. You must know the archive file name when you execute the import command later in the migration process.

- **Direct Export Command Output to the Screen**
- **Redirect Export Command Output to a File**
- **Capture the Archive File Name and Monitor Command Progress**

**Direct Export Command Output to the Screen**

You can export the design-time metadata from the instance and show all progress on the screen. If you use this option, pay attention to the name of the archive file. You need the name when executing the import command. The name is not saved to a file unless you explicitly redirect output of the export command to a file.

```
opcmigrate migrate oic export -s ServiceName
```
The following output is returned. For this example, only Integrations design-time metadata is available to export. If this export also included Processes data, status for that component would also be displayed.

2019-07-15T15:26:44 INFO Exporting OIC service name 'ServiceName'
2019-07-15T15:26:50 INFO Started exportServiceInstanceArchive ['jobId': '10163', 'location': 'https://object_storage_host.us-ashburn-1.oraclecloud.com/v1/oraclemigration', 'status': 'NOT_STARTED', 'overallStatus': 'NOT_STARTED'}
2019-07-15T15:29:52 INFO ['jobId': '10163', 'archiveName': '', 'overallStatus': 'STARTING', 'componentStatus': [{'component': 'Integration', 'status': 'NOT_STARTED'}]}
2019-07-15T15:29:54 INFO ['jobId': '10163', 'archiveName': '', 'overallStatus': 'RUNNING', 'componentStatus': [{'component': 'Integration', 'status': 'NOT_STARTED'}]}
2019-07-15T15:29:57 INFO ['jobId': '10163', 'archiveName': '', 'overallStatus': 'RUNNING', 'componentStatus': [{'component': 'Integration', 'status': 'RUNNING'}]}
2019-07-15T15:30:03 INFO ['jobId': '10163', 'archiveName': 'archive_Local_Suite_Instance-10163.zip', 'overallStatus': 'COMPLETED', 'componentStatus': [{'component': 'Integration', 'status': 'COMPLETED', 'percentage': 100}]

Redirect Export Command Output to a File

You can redirect the output of the `export` command to a file. To capture the archive file name, be aware of the following issues:

- `opcmigrate` logs to `stderr`. Therefore, redirection must be performed with `&>` as follows:

```bash
opcmigrate migrate oic export -s ServiceName &> oic_export.log
```

Without the `&`, the log file is empty and the command continues to log to the screen.

- When logging is sent to a file (as opposed to the screen), the format of the log changes. The timestamp is omitted and each message is prefixed with `oic-migrator:`.

```bash
oic-migrator: INFO: Exporting OIC service name 'ServiceName'
oic-migrator: INFO: Started exportServiceInstanceArchive ['jobId': '16283', 'location': 'https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/CloudAccountName/oraclemigration', 'status': 'NOT_STARTED', 'overallStatus': 'NOT_STARTED']
oic-migrator: INFO: ['jobId': '16283', 'archiveName': '',
```
Capture the Archive File Name and Monitor Command Progress

If you want to capture both the archive file name and monitor the progress of the export command, pipe the output into the tee command. Note that:

- As with redirection, it must be done with |& instead of | to include stderr.
- Messages logged by the tee command to the screen have the same format as with redirecting to the log file. opcmigrate is not aware that its stderr is redirected to tee and that the output goes to the screen.
- Logging does not show the spinner, which indicates that the command is working and making progress.

```
opcmigrate migrate oic export -s ServiceName |& tee oic_export.log
```

You can also monitor command progress on the screen while capturing the log to a file with the script command. The syntax is more complex. With this method, the spinner is displayed after each log message.

```
script oic_export.log -c "opcmigrate migrate oic export -s ServiceName ; exit"
```

Script started, file is oic_export.log
2019-08-01T22:36:39 INFO Exporting OIC service name 'ServiceName'
PaaS Services Password [user@example.com]:

2019-08-01T22:36:57 INFO Started exportServiceInstanceArchive {'jobId':
Create the Target Instance on Oracle Cloud Infrastructure

Create a new Oracle Integration instance in an Oracle Cloud Infrastructure region.

Note:

You can also create an instance through the My Services Console. See Create an Oracle Integration Instance in Administering Oracle Integration.

Create a new Oracle Integration instance in Oracle Cloud Infrastructure.

```bash
opcmigrate migrate oic create -c target_config -t target_instance_name
```

The following output is returned.

```
2019-07-31T20:43:06 INFO Creating OIC service instance
'target_instance_name'
```
PaaS Services Password [firstname.lastname@example.com]:
2019-07-31T20:43:43 INFO {'details': {'message': 'Submitted job to create service [target_instance_name] in domain [idcs-00000000000000000000000000000000].', 'jobId': '1111111111'}}

Once instance creation completes, you can import your design-time metadata into the new instance.

Creating a new instance does not automatically create a user name and password for that instance. You must go to the My Services Console and create a user name and password for the new instance. The user name specified should be assigned the ServiceAdministrator role for this new instance. This information is required when editing the target configuration file. See Edit the Target Configuration File.

Edit the Target Configuration File

You must create an Oracle Integration target configuration file in the same ~/.opc/profiles directory as the default profile file or in a different profile directory that you specify with the --profile-directory option when you invoke the opc migrate oic export command. This file defines the details to create a new Oracle Integration instance with the opc migrate tool.

The target configuration file contains only PaaS ("paas") and Oracle Integration ("oic") sections. You can have the same identity domain for both the source and destination environments. If the same identity is used for both environments, then copy the "paas" section from the profile file without the "region" parameter.

You can create an instance without specifying any parameters. In those cases, default values are provided. The following example describes how to populate the target configuration file with specific parameters and values. The Integration and Process feature set is always provisioned, regardless of whether the source instance is Standard (Integration only) or Enterprise (Integration and Process).

- **TargetDefaultServiceName**: Enter the name of the new instance created in Create the Target Instance on Oracle Cloud Infrastructure. Enter the instance name appropriate to your environment.

- **admin_user**: Enter the administration user to use for importing the design-time metadata into the new instance. The admin_user name specified should be assigned the ServiceAdministrator role for the service instance in the My Services Console after the instance is created.

- **admin_password**: Enter the administrator password.

The customization parameters are specified in the "create_parameters" subsection of each target service name:

- **isBYOL**: Enter true to bring an existing Oracle Fusion Middleware license to the cloud to use with Oracle Integration or false to create a new Oracle Integration license.

- **numMessagePacks**: If isBYOL is set to the following:
  - true: Enter 1, 2, or 3 for the number of message packs per hour. Specifying true provides you with packages of 20K messages per hour.
  - false: Enter 1 through 12 for the number of message packs per hour. Specifying false provides you with packages of 5K messages per hour.
• notificationEmail: Enter an email address at which to be notified of instance provisioning progress (for example, when provisioning completes).

• region: Enter the region in which to create the instance. This parameter is optional.

• serviceDescription: Enter a description to identify this new service. The description is used only in the instance list display.

```
{
    "paas": {
        "user": "user@example.com",
        "identity_id": "idcs-00000000000000000000000000000000",
        "endpoint": "psm.us.oraclecloud.com",
        "password-file": "/home/opc/myociservicespass"
    },
    "oic": {
        "TargetDefaultServiceName": {
            "admin_user": "user@example.com",
            "admin_password": "password"
        },
        "TargetCustomizedServiceName": {
            "admin_user": "user@example.com",
            "admin_password": "password",
            "create_parameters": {
                "isBYOL": true,
                "numMessagePacks": 2,
                "notificationEmail": "user@example.com",
                "region": "ca-toronto-1",
                "serviceDescription": "Integration instance migrated from OCI-C"
            }
        }
    }
}
```

**Import Design-Time Metadata**

After the service instance starts on Oracle Cloud Infrastructure, import the new instance.

1. Import the archive file into the new instance. You specify the source instance so that the profile file can be accessed for storage container information.

```
opc migrate oic import -c target_configuration_file -s source_instance -t target_instance -a archive_Local_Suite_Instance-10163.zip
```

The following output is returned.

```
2019-07-15T15:26:44 INFO Importing OIC service name 'source_instance' from archive_Local_Suite_Instance-10163.zip to target_instance
```

...
2. Log in to your Oracle Integration instance.

3. Browse the pages and note that the design-time metadata you exported from Oracle Cloud Infrastructure Classic is now visible. For example, for Integrations, look for integrations, connections, lookups, and more.