**Oracle Cloud Infrastructure (OCI)**

It is a set of complementary cloud services that enable you to build and run a range of applications and services in a highly available hosted environment. OCI provides high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

*OCI is* ***physically hosted*** *in* ***regions*** *and* ***availability domains***

**OCI -Physical Architecture**

Components:

* **Regions** - localized geographic area, are globally distributed data centers
* **Availability domain** - one or more data centers located within a region
* **Fault Domains** - is a grouping of hardware and infrastructure within an availability domain. (Distribute your **instances** so that the instances are not on the same physical hardware within a single availability domain)
* **Realms** - A realm is a **logical collection of regions**. Realms are **isolated** from each other and do **not share any data**. Your tenancy exists in a single realm and has access to the regions that belong to that realm. OCI currently offers realms *for commercial regions, government regions, and dedicated regions*.

**Account and Access Concepts:**

* **Tenancy** - When you **sign up or subscribe to Oracle Cloud services**, Oracle creates a tenancy for you. You can think of the tenancy as your account, but it is also a **secure and isolated partition within Oracle Cloud Infrastructure** where you can **create, organize, and administer** **your cloud resources**. When you sign up, your tenancy is created in your **home region**, but you can subscribe your tenancy to **as many regions as you need**. Large organizations can have multiple tenancies
* **Compartment** -allow you to **organize and control access** to your cloud resources. **collection of related resources** (such as **instances, virtual cloud networks, block volumes**). When you sign up for Oracle Cloud Infrastructure, Oracle creates your **tenancy**, which is the **root compartment** that holds all your cloud resources. You then create additional compartments within the tenancy (root compartment) and corresponding policies to control access to the resources in each compartment.
* **Identity Domains and Policies –** It is a container for managing **users and roles**, federating and provisioning of users, secure application integration through **Oracle Single Sign-On (SSO)** configuration, and **OAuth administration**.  
  A **policy** is a document that specifies **who can access which resources**, and how. You can write policies to control access to all of the services within Oracle Cloud Infrastructure. Access is granted at the group and compartment level, which means you can write a policy that gives a group a specific **type of access within a specific compartment, or to the tenancy itself.** If you give a group access to the tenancy, the group automatically **gets the same type of access to all the compartments inside the tenancy**
* **Oracle Cloud Identifier (OCID) -** Every Oracle Cloud Infrastructure resource has an Oracle-assigned unique ID called an Oracle Cloud Identifier (OCID). It's included as part of the resource's information in both the Console and API. **To use the API, you need the OCID** for your tenancy

ocid1.<RESOURCE TYPE>.<REALM>.[REGION][.FUTURE USE].<UNIQUE ID>

* **Security Zone -** A security zone is **associated** with **one or more compartments** and a security zone recipe. When you create and update resources in a security zone, Oracle Cloud Infrastructure **validates these operations against security zone policies** in the zone's recipe.

**Core Services Concepts:**

* **VCN - virtual version of a traditional network**—**including subnets, route tables, and gateways**—on which your **instances** **run**. A cloud network **resides within a single region** **but includes all the region's availability domains**. Each subnet you define in the cloud network can either be in a single availability domain or span all the availability domains in the region (recommended). You **need to set up at least one cloud network before you can launch instances**.
* **Instance** - is a **compute host running in the cloud**. An Oracle Cloud Infrastructure compute instance allows you **to utilize hosted physical hardware**, as opposed to the **traditional software-based virtual machines**, ensuring a high level of security and performance.
* **Block volume -** is a **virtual disk** that provides persistent block **storage space for Oracle Cloud Infrastructure instances.** Use a block volume just as you would a **physical hard drive** on your computer, for example, **to store data and applications. You can detach a volume from one instance and attach it to another instance without loss of data**.

**Oracle Cloud Infrastructure Cloud Adoption Framework**

**Applications Services**

**Infrastructure Services**

OIC Questions:

1. **What are the products within OIC?**

Integration Cloud, VBCS (Visual Builder Cloud Service), PCS (Process Cloud Service)

1. **Is OIC PaaS, SaaS or IaaS?**

The interviewer wants to test your understanding here. OIC’s old name was ICS and it was started as a PaaS Service. Later most of the Platform Services are now part of IaaS. Definitely not SaaS

refer https://www.oracle.com/index.html Click on products to check the latest hierarchy

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Oracle Integration**   | Product | Unit price | Metric | | --- | --- | --- | | **Oracle Integration Cloud Service - Standard** | **$0.6452** | **5K Messages/Hour** | | **Oracle Integration Cloud Service - Enterprise** | **$1.2903** | **5K Messages/Hour** |   **Oracle Integration - Bring Your Own License (BYOL)**   | Product | Unit price | Metric | | --- | --- | --- | | **Oracle Integration Cloud Service - Standard - BYOL** | **$0.3226** | **20K Messages/Hour** | | **Oracle Integration Cloud Service - Enterprise - BYOL** | **$0.3226** | **20K Messages/Hour** |   **Application Integration - OIC**  UTILIZATION  Edit  1 instance  x  744 hrs/month  ESTIMATED MONTHLY COST  **$480.03**  License Edition      License Type      5k messages per Message Pack  Real Time Messages    Assume 50kB per message  File Processing [MB]    Total size of files processed per hour [MB]  Message Packs    5k messages per Pack, 20k for the BYOL   1. Can Oracle Integration Cloud be used to integrate with non-Oracle applications?   Yes, Oracle Integration Cloud can be used to integrate with a wide range of non-Oracle applications and data sources, including popular cloud applications such as Salesforce and Workday, as well as on-premises systems and custom applications.   1. What are the types of Integration Patterns?   App Driven, Scheduled Orchestration, File Transfer, Basic Routing, Publish to OIC, Subscribe to OIC   1. **Accelerators** are run-ready business integrations or technical patterns you can configure and activate. 2. **Recipes** are starter templates that give you a head start.   **Avoid Common Integration Style Pitfalls**   * Chatty Integrations –      * Scheduled Job that Never Stops Trying to Process      * Import an Externally Updated IAR File      * Synchronous Integration Doing Too Much      * Too Many Connections in an Integration      * Read Files with Many Records      * Integrations Running Unchanged Despite Changing Business Needs     [**https://docs.oracle.com/en/cloud/paas/integration-cloud/integrations-user/common-integration-style-pitfalls-and-design-best-practices.html#GUID-73AE4442-9D9E-446C-9E33-EAD36DBC45F1**](https://docs.oracle.com/en/cloud/paas/integration-cloud/integrations-user/common-integration-style-pitfalls-and-design-best-practices.html#GUID-73AE4442-9D9E-446C-9E33-EAD36DBC45F1)  **A synchronous integration calling any asynchronous request** response service:  **Calling asynchronous fire and forget** (one-way) is acceptable.  Oracle Integration does not currently allow modeling an asynchronous request response service. However, all scheduled orchestration styles internally use an asynchronous request response. Therefore, a synchronous integration using a scheduled orchestration is an antipattern.  **Parallel Processing in Outbound Integrations**  Separate the integration into multiple integrations:   * Create a main **parent integration** that only receives/processes the data. * Create separate child integrations to perform the individual outbound REST invocations. |  |  |

The interface between the main and separate child integrations can follow these approaches:

* Consist of dummy REST calls, but it must be **asynchronous**. Essentially, the asynchronous calls are not blocked by the response and **the fire-and-forget** design enables available threads to work on child integration processing in parallel, within the available system resources. This type of design is recommended because **if all synchronous REST calls are done in the same integration, a time out error may occur** if the sum of time taken for each synchronous call exceeds five minutes.
* Follow a **publish/subscribe** design approach (for example, putting the data events in a queue, having each child flow subscribe from the queue, and so on).

1. What is the difference between Trigger and Invoke Connection?

Trigger – It can only trigger. Trigger and Invoke Can invoke from inside the process as well.

1. What is the Difference between Rest and SOAP API?

At High level Rest is light weight and reduces the load on network, more user friendly, supports different formats. SOAP is heavy, Full object has to be sent for communication.

|  |  |
| --- | --- |
| REST | SOAP |
| REST is an architectural style that primarily uses HTTP as the communication protocol. It leverages HTTP verbs (GET, POST, PUT, DELETE) and status codes to interact with resources over the web. | SOAP is a protocol that uses XML-based messages for communication. It can work over various protocols such as HTTP, SMTP, and more, but it is commonly used with HTTP. |
| REST typically uses lightweight and human-readable data formats such as JSON (JavaScript Object Notation) or XML for message payloads. | SOAP uses XML for message formatting, including request and response envelopes. The XML structure provides a standardized format for defining message headers, bodies, and metadata. |
| REST follows a resource-based architectural style, where resources are identified by URLs (Uniform Resource Locators) and accessed via HTTP methods. Each resource can have multiple representations, and clients interact with these resources by sending requests and receiving responses. | SOAP focuses on the operations or methods provided by a service. It defines a contract-based interface using Web Services Description Language **(WSDL)**, which describes the available operations, their inputs, and outputs. |
| REST is known for its scalability and performance. | SOAP can be less scalable due to its XML payload, which tends to be larger than REST's lightweight data formats. SOAP also relies on more extensive processing and parsing of XML messages, potentially impacting performance. |

1. What are the prerequisites for OIC to read a Business Event from Oracle SaaS?

* Business Event should be enabled from SaaS
* CSF Key of OIC needs to be registered in SaaS Soa Composer
* OIC process should have subscribed the specific Event in SaaS
* In some cases, Fusion SaaS Certificates should be imported in OIC

1. What is Service Catalog URL of Oracle Fusion SaaS?

This is a WSDL URL used in OIC connections to fetch the list of services available. The newer versions of OIC do not need this URL now. The domain name can be directly used.

1. What is the difference between a template parameter and query parameter?

The template parameter is the part of the Rest API path and denoted by {variable} whereas Query parameter is query string appended after the question mark “?” in the rest API url.

* Template parameters are part of the URL path and used to identify a specific resource or entity.
* Query parameters are appended to the URL after the "?" symbol and provide additional information or parameters to the request.
* Template parameters use placeholders within the URL path, while query parameters use key-value pairs at the end of the URL.
* Template parameters provide a clear and readable way to indicate the target of the request, while query parameters offer flexibility for filtering, sorting, and customizing request behavior.

1. Have you Used Adapters? Why and when do you use Adapter instead of direct Rest or SOAP Call?

Adapters make calling specific business services easy as they expose the required services with all the security measures. The connection can only call the services to which it is authorized to.

1. What is a feature Flag in OIC?

To enable disable certain features which are still not part of mainstream GA application.

1. What is Opaque Schema?

When you just want to write/read the file as binary. Opaque schema can be used.

1. What is the use of package in the OIC?

We can use package functionality to group different processes which are part of a business integration flow. example if you have 3 flows being executed for a integration these 3 can be packaged so thats its easily identifiable and can be exported as a single package zip with 3 processes. Deployment also becomes easy.

1. Is editing XSL mappings allowed? Have you ever modified XSL mappings in OIC?

XSL mappings can be modified externally and can be uploaded as new maps. The integration process gets locked if an XSL modified outside is uploaded.

The newer release provides the capability to edit xsl in OIC mappings design view directly.

1. You are given an old integration IAR to be imported and when you do you get a function error, How do you fix it?

Some old functions may be deprecated or there can be namespace issues. Generally, xsl file needs to be corrected with the correct function and namespaces which the OIC can understand.

1. What is NameSpace in XML file?

Namespace defines and identifies uniquely the elements in the Payload.

1. When should we choose a scheduled orchestration over app driven?

When source payload is not available immediately Scheduled orchestration is used. This will not be realtime or near real-time. AppDriven integrations are better for realtime flows

1. What are Synchronous and Asynchronous Processes?

Synchronous- Fire and Wait for response (waits for a response before proceeding further.)

Asynchronous- Fire and Forget.

**The following expression indicates that this integration runs each month on the 1st, 10th, and 15th days of the month at 5:15 AM, 10:15 AM, 3:15 PM, and 8:15 PM.**

* Ical expression:
* FREQ=MONTHLY;BYMONTHDAY=1,10,15;BYHOUR=5,10,15,20;BYMINUTE=15;

**Every day between the hours of 5:30 PM – 7:30 PM, and during these hours it executes every 10 minutes.**

FREQ=DAILY;BYHOUR=17;BYMINUTE=30,40,50;BYSECOND=0;

&FREQ=DAILY;BYHOUR=18;BYMINUTE=10,20,30,40,50; BYSECOND=0;

&FREQ=DAILY;BYHOUR=19;BYMINUTE=10,20,30; BYSECOND=0;

**schedule runs daily at 8 AM and also monthly at 12 PM on day 1 and day 2.**

FREQ=DAILY;BYHOUR=8;

&FREQ=MONTHLY;BYMONTHDAY=1,2;BYHOUR=12;