Managed and Comanaged Consolidation and Deployment Strategies for Oracle Cloud Infrastructure

March 2023, version 1.0

Public

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

Purpose

This paper provides a strategy and methodology for consolidating applications, databases, and virtual machines (VMs) in Oracle Cloud Infrastructure (OCI) with a focus on the following OCI services:

- Managed and comanaged Exadata Cloud Service
- Managed and comanaged Exadata Cloud@Customer
- VM and bare metal database (DB) systems
- Autonomous Data Warehouse
- Autonomous Transaction Processing
- VMs (application and database compute capacity)

Although this paper discusses non database assets, such as applications and VMs, its focus is on managed and comanaged database consolidation to OCI. Consolidation and isolation techniques using tenancies, compartments, virtual clouds networks (VCNs), and policies are covered at a high level.

Introduction

How many times have you started a migration project and were provided details such as these?

- "I have 11 applications connecting to 27 databases, and I need to migrate to Exadata Cloud@Customer or OCI in the next 30–60 days."
- "I have 7,000 databases, and I need to migrate them to OCI over the next 2 years."

How do you organize the project to create a deployment plan that meets an aggressive timeline, without exceeding capacity in the middle of the project, all while meeting your customer's objectives? This paper attempts to answer those questions for small and large migration projects with practical strategies and a methodology that meet such objectives. It provides benefit for all application, database, and VM services targeted for OCI when you have been given the following resources:

- A new application and database inventory (estate) that contains little information
- An existing architecture and capacity plan that lacks sufficient details about how the assets should be deployed, isolated, and transformed into the target architecture

The methodology outlined in this paper has the following steps:

1. Perform estate discovery (inventory)
2. Get resource usage for the estate
3. Classify and assign estate assets
4. Identify consolidation objectives, strategy, and rules
5. Identify the initial target architecture
6. Identify target resource limits
7. Define the resource management strategy
8. Run a modeling scenario

Following these steps identifies important details up front and brings the customer into the decision-making process, thus increasing migration success. This paper also provides a case study that outlines the use of the Migrate 360 application to show a real-world example and how these techniques can be used to obtain a successful project outcome.

Perform Estate Discovery (Inventory)

All migration and consolidation strategies begin with a discovery step. This step typically involves getting the business and technical details of your customer's estate and inventory.

- An *estate* is a list of the applications, databases, and VMs that are to be consolidated and migrated to the target architecture.
- *Business details* are typically the drivers for the purchase of the target architecture.
- *Technical details* represent the resources and physical assets identified as the customer estate.

An example estate is a list of 11 applications, 23 databases, and 23 VMs with related CPU, memory, and storage usage. Examples of business and technical details are as follows.

Business Details

Following are some typical, example drivers for the purchase of the target architecture:

- Six Exadata Cloud@Customer full racks were purchased with the intent of migrating 300 databases.
- Exadata Cloud@Customer racks must be evenly distributed in data centers in New York and San Francisco, with one rack for nonproduction and two racks for production (one primary and one standby in each data center).
- The Finance application that contains five databases per lifecycle environment must be migrated and cut over within 30 minutes.
- Six Exadata Cloud@Customer VM clusters must be created with isolation requirements that maintain Payment Card Industry (PCI) and non-PCI separation of production and nonproduction at the frame, security, and network layers. This separation consists of 200 databases assigned to the PCI zone and 100 databases assigned to the non-PCI zone. Each cluster must be provided with sufficient headroom based on free-space requirements.
- One non-PCI Manufacturing application and one PCI Payment application are being migrated. Each application contains two web servers, two application servers, and two databases. Each application also supports one development, test, integration, and production lifecycle environment to be deployed in OCI while maintaining physical, network, and security isolation.
- Because of recent outages, all Finance production databases must be converted from a single instance to a 4-node Oracle Real Application Clusters (RAC) with Data Guard during the migration process.
- The multitenancy strategy must allow for the separation of all current company lines of business (LOB) assets per lifecycle environment.
- All standby assets must be deployed in a separate geographic location based on the corporate business continuity plan (BCP).

Technical Details

Following are some example resources and physical assets identified as a customer estate:

- For the identified estate of 300 databases, the customer requires 20% free CPU, 30% free memory, and 40% free disk space for postmigration growth for all assets.
- Each Finance production database (fin1, fin2, fin3, fin4, and fin5) requires a 40-GB system global area (SGA), a 20-GB program global area (PGA), 20 CPUs, and 10 TB of storage.

- The total resource requirement for the PCI zone is 200 Oracle CPUs (OCPUs). For the non-PCI zone, the requirement is 350 OCPUs.
- The conversion of the Finance database from tier 3 to tier 1 requires three more instances per database, consuming three times the amount of memory resources originally identified.
- Container databases (CDBs) require an extra consolidation attribute related to the time zone.
- All tier-3, single-instance databases must be converted to tier 2, which consists of 2-node RAC databases.

Get Resource Usage for the Estate

After discovery, the next step is to get an accurate representation of the resource usage of the source estate. You get this information by *reverse engineering* in the following ways:

- Querying configuration management repositories such as Oracle Enterprise Manager
- Querying the source databases directly
- Using OS tools to obtain the necessary application and VM metrics

Tip: Oracle Enterprise Manager has a longer retention period and should be your first choice for obtaining asset resource usage. When Oracle Enterprise Manager isn't available, you can obtain this information from the source database.

This step is critical for developing an enterprise consolidation plan. During this step, you identify the following source metrics for all applications, databases, and VMs:

- OCPU and vCPU usage
- Memory size (SGA and PGA)
- Storage requirements and database size

CPU Threads (vCPUs)

To understand the concept of CPU threads, you must understand the differences among cores, threads, Oracle vCPUs, and OCPUs.

- One OCPU is equivalent to one CPU core.
- Each CPU core, or OCPU, is multithreaded and is seen by the OS as two CPUs, which are referred to as vCPUs.

When you reverse engineer the CPU usage of a source database, the goal is to get the *number of CPU threads* used by the source application, database, or VM, *not* the CPU usage as a percentage. After you have this information, you then calculate 80% of the maximum threads returned. This number removes outliers and short-term, explainable spikes that don't represent the normal workload. This information gives you the following advantages:

- You can prevent the overallocation of CPU resources and unnecessary cost.
- You can prevent the underallocation of CPU resources and CPU starvation.
- You can identify appropriate CPU values for Oracle Resource Manager and instance caging to mitigate the impact of "noisy neighbors."

To get the optimal CPU allocation for a target database, run the queries in the following table against the Oracle Enterprise Manager repository or the Automatic Workload repository created for each individual database. These queries return the maximum number of CPU threads used for a given database. Then, identify the 80th percentile and validate with the customer that the outliers can be discarded. You can get application and VM resources from standard OS tools.

Note: For details about the data-retention policies for each of these repositories, review the Oracle documentation. The longer the retention policy, the more accurate the capacity plan will be.

Table 1: CPU Metric Queries

REPOSITORY	SQL QUERY

Enterprise Manager repository
 	
(first choice)
	`select ceil(max(maximum)) max_db_cpu_count`
	`from (select * from (select m.maximum from sysman.mgmt$metric_daily m`
	`where m.target_guid = :target_guid`
	`and m.target_type ='oracle_database'`
	`and m.metric_guid = HEXTORAW('AB9CAF80590230CCA321554A36A83E56')`
	`order by maximum desc )`
	`offset 5 rows`
	`FETCH NEXT 10 ROWS only)`

Automatic Workload repository
 	
(second choice)
	`select snap_id "snap",num_interval "dur_m", end_time "end",inst "inst",`
	`max(decode(metric_name,'Host CPU Utilization (%)', average,null)) "os_cpu",`
	`max(decode(metric_name,'Host CPU Utilization (%)', maxval,null)) "os_cpu_max",`
	`max(decode(metric_name,'Host CPU Utilization (%)', STANDARD_DEVIATION,null)) "os_cpu_sd",`
	`max(decode(metric_name,'Physical Write Total IO Requests Per Sec', average,null)) "write_iops",`
	`max(decode(metric_name,'Physical Write Total IO Requests Per Sec', maxval,null))`
	`"write_iops_max"`
	`from(`
	`select snap_id,num_interval,to_char(end_time,'YY/MM/DD HH24:MI') end_time,instance_number`
	`inst,metric_name,round(average,1) average,`
	`round(maxval,1) maxval,round(standard_deviation,1) standard_deviation`
	`from dba_hist_sysmetric_summary`
	`where`
	`metric_name in ('Host CPU Utilization (%)','Physical Write Total IO Requests Per Sec')`
	`)`
	`group by snap_id,num_interval, end_time,inst`
	`order by snap_id, end_time,inst;`

After you get the required CPU for each database, application, and VM, the next step is to get an accurate representation of the memory resource requirements of the source estate.

Memory, SGA, and PGA

To accurately size the memory for the compute capacity of the target architecture, determine the amount of memory used per application, database, and VM.

- Identify the amount of SGA and PGA memory required for one or more databases.
- Identify the amount of huge page memory required for the SGA pools on the Linux platform and the amount of PGA user memory.
- Identify appropriate SGA and PGA values for Oracle Resource Manager and pool minimums to provide pluggable databases (PDBs) with sufficient pool minimums in a multitenancy environment.
- Identify whether sufficient memory is available for the applications and VMs to meet the necessary resource demand.

To get the SGA and PGA metrics used for a given database, run the queries in the following table against the Oracle Enterprise Manager repository or the Automatic Workload repository created for each individual database. You can get application and VM resources from standard OS tools.

Table 2: SGA and PGA Metric Queries

REPOSITORY	SQL QUERY

Enterprise Manager repository
 	
(first choice)
	<pre>select ceil(max(maximum)/1024) max_db_sga_size
	                                     from (select * from (select m.maximum
	        from sysman.mgmt$metric_daily m
	                                     where m.target_guid = :target_guid
	                                        and m.target_type ='oracle_database'
	        and m.metric_guid = HEXTORAW('0DD82DEF4D7B007FB0D5492E417A7C26') order by maximum desc)
	                                        offset 5 rows FETCH NEXT 10 ROWS only)
	select name, value/1024/1024/1024 from
	sysman.mgmt$db_init_params_all
	where target_guid = :target_guid
	and name in ('pga_aggregate_limit' ,'pga_aggregate_target','sga_max_size',’ sga_target');</pre>

Automatic Workload repository
 	
(second choice)
	<pre>SELECT snap_id,
	    instance_number,
	    MAX (DECODE (stat_name, 'SGA', stat_value, NULL)) "SGA",
	    MAX (DECODE (stat_name, 'PGA', stat_value, NULL)) "PGA",
	    MAX (DECODE (stat_name, 'SGA', stat_value, NULL)) + MAX (DECODE (stat_name, 'PGA', stat_value,
	    NULL)) "TOTAL"
	   FROM
	    (SELECT snap_id,
	        instance_number,
	        ROUND (SUM (bytes) / 1024 / 1024 / 1024, 1) stat_value,
	        MAX ('SGA') stat_name
	       FROM dba_hist_sgastat
	      WHERE dbid = &DBID
	        AND snap_id BETWEEN &SNAP_ID_MIN AND &SNAP_ID_MAX
	   GROUP BY snap_id,
	        instance_number
	  UNION ALL
	     SELECT snap_id,
	        instance_number,
	        ROUND (value / 1024 / 1024 / 1024, 1) stat_value,
	        'PGA' stat_name
	       FROM dba_hist_pgastat
	      WHERE dbid = &DBID
	        AND snap_id BETWEEN &SNAP_ID_MIN AND &SNAP_ID_MAX
	        AND NAME = 'total PGA allocated'
	    )
	GROUP BY snap_id,
	    instance_number
	ORDER BY snap_id,
	    instance_number;</pre>

After you validate the required memory metrics for each application, database, and VM, the next step is to get the storage requirements of the source estate.

Storage Requirements and Database Size

To validate the storage capacity of the target architecture, determine the size of the databases.

- Identify the amount of storage required for the database in question.
- Identify whether the target storage capacity can support all the databases being consolidated onto the target architecture.
- Identify whether sufficient free space can be reserved for growth, as determined by the customer during estate discovery.
- Identify whether sufficient storage is available for the applications and VMs to meet the necessary resource demand.

To obtain the storage requirements for a given database, run the queries in the following table against the Oracle Enterprise Manager repository or the Automatic Workload repository created for each individual database. You can obtain application and VM resources from standard OS tools.

Note: The following queries were used to complete the DB_SIZE_GB column in the Migrate 360 database load sheet. See "Estate Resource Usage (Reverse Engineering)."

Table 3: Database Size Metric Queries

REPOSITORY	SQL QUERY
Enterprise Manager repository (first choice)	`select ceil(sum(tablespace_size)/1024/1024/1024) from sysman.mgmt$db_tablespaces  where target_guid = :target_guid;`

Automatic Workload repository
 	
(second choice)
	`select s.snap_id, round(sum(tablespace_size*f.block_size)/1024/1024/1024,2) size_gb`
	`from dba_hist_tbspc_space_usage sp,`
	`ts_info f,`
	`snap_info s`
	`WHERE s.dbid = sp.dbid`
	`AND s.dbid = &DBID`
	`and s.snap_id between &SNAP_ID_MIN and &SNAP_ID_MAX`
	`and s.snap_id = sp.snap_id`
	`and sp.dbid = f.dbid`
	`AND sp.tablespace_id = f.ts#`
	`GROUP BY  s.snap_id,s.dd, s.dbid`
	`order by  s.snap_id;`
	`--`
	`--For RAC`
	`select name Disk_group_name, total_size_gb,free_size_gb from (`
	`select name, nvl(sum(total_mb),0)/1024 total_size_gb, nvl(sum(free_mb),0)/1024 free_size_gb`
	`from  v$asm_diskgroup where name like 'DATA%' group by name`
	`union all`
	`select name, nvl(sum(total_mb),0)/1024 total_size_gb, nvl(sum(free_mb),0)/1024 free_size_gb`
	`from  v$asm_diskgroup where name like 'RECO%' group by name);`

Classify and Assign Estate Assets

The next step is to classify the estate assets so that they can be mapped to the target architecture and initiate a high-level capacity analysis. Assets must be assigned to appropriate compute and database shapes before they can be transformed into optimal configurations in the target architecture. Following are examples of these items:

- **Compute shape**: Compute shape options defined within OCI, such as VM.Standard.2.1
- **Exadata VM cluster**: Number of compute nodes, allocated memory, internal disk (/u02), and ASM disk
- **Database shape**: Small, medium, and large
- **Service tier (tiering definition and operational alignment)**: Bronze, Silver, Gold, Platinum
- **Security zone and isolation alignment**: Non-PCI, PCI, HIPAA, SOX compliance
- **Operational isolation:** Systems with their own maintenance and patching in an upgrade window

These topics are covered in detail in the following sections.

Compute Shapes

Compute shapes are templates that you use provision the target VMs in OCI. To ensure the necessary performance of the target architecture, align the capacity of the source VMs with the target architecture. Example compute shapes are provided in the following table. Other shapes—such as Flexible, Bare Metal, and Dedicated VM— can be used as well. For the most up-to-date shapes, see Compute Shapes in the OCI documentation.

Note: The APP_HOST_CPU_COUNT, APP_HOST_RAM_GB, APP_HOST_DISK_GB, and APP_OS values from the Migrate 360 application and VM load sheet are used to identify the appropriate shape. See "Estate Resource Usage (Reverse Engineering)."

Table 4: Standard VM Compute Shapes

SHAPE	OCPU	MEMORY (GB)	LOCAL DISK (TB)	MAX NETWORK BANDWIDTH	MAX VNICS TOTAL: LINUX	MAX VNICS TOTAL: WINDOWS
VM.Standard2.1	1	15	Block storage only	1 Gbps	2	2
VM.Standard2.2	2	30	Block storage only	2 Gbps	2	2
VM.Standard2.4	4	60	Block storage only	4.1 Gbps	4	4
VM.Standard2.8	8	120	Block storage only	8.2 Gbps	8	8
VM.Standard2.16	16	240	Block storage only	16.4 Gbps	16	16
VM.Standard2.24	24	320	Block storage only	24.6 Gbps	24	24

Database Shapes (Standardization)

A database shape is a template that determines the required target database resources—such as extra small, small, medium, large, and extra-large—for containers (CDBs) and pluggable databases (PDBs). The following table shows an example CDB and PDB standardization strategy. To use this strategy, each database must be assigned to a required shape and summarized in the modeling exercise. Standard database shapes provide certain advantages, including the following ones:

- Simplified capacity planning
- Proven configuration for new systems in which sizing can't yet be determined
- Standard deployments that are easy to size, manage, and tune
- Starting point for CPU, SGA, and PGA that can be fined-tuned based on performance requirements

Table 5: Database Sizes (Example Ranges Only)

PARAMETER	EXTRA SMALL (PDB)	SMALL (PDB)	MEDIUM (PDB)	LARGE (CDB/PDB)	EXTRA LARGE (CDB/PDB)	CUSTOM (>=)
`db_block_size`	8192	8192	8192	8192	8192	8192
`cpu_count`	2	4	8	16	32	32
`java_pool_size`	64 MB	80 MB	128 MB	128 MB	128 MB	128 MB
`large_pool_size`	64 MB	128 MB	256 MB	256 MB	512 MB	512 MB
`log_buffer`	128 MB	128 MB	128 MB	128 MB	128 MB	128 MB

PARAMETER	EXTRA SMALL (PDB)	SMALL (PDB)	MEDIUM (PDB)	LARGE (CDB/PDB)	EXTRA LARGE (CDB/PDB)	CUSTOM (>=)
pga_aggregate_target (PAT)	512 MB	1,024 MB	2,048 MB	4,096 MB	512 MB	512 MB
pga_aggregate_limit	2x PAT	2x PAT	2x PAT	2x PAT	2x PAT	2x PAT
processes	512	1024	1024	1024	3000	3000
redo size	512 MB	512 MB	1,024 MB	2,048 MB	4,096 MB	4,096 MB
resource_manager_plan	<name>_plan	<name>_plan	<name>_plan	<name>_plan	<name>_plan	<name>_plan
sga_target	2 GB	4 GB	8 GB	16 GB	24 GB	24 GB
shared_pool_size	64 MB	128 MB	256 MB	256 MB	512 MB	1,024 MB
user_large_pool	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Service Tier (Tiering Definition and Operational Alignment)

A *service tier* or *tiering definition* is a reference architecture or blueprint that aligns itself with specific criteria and services levels to meet the needs of the business. As you work with customers, you encounter many tiering definitions. However, Oracle's Maximum Availability Architecture (MAA) is the most comprehensive definition in the industry and uses the tiers Bronze, Silver, Gold, and Platinum.

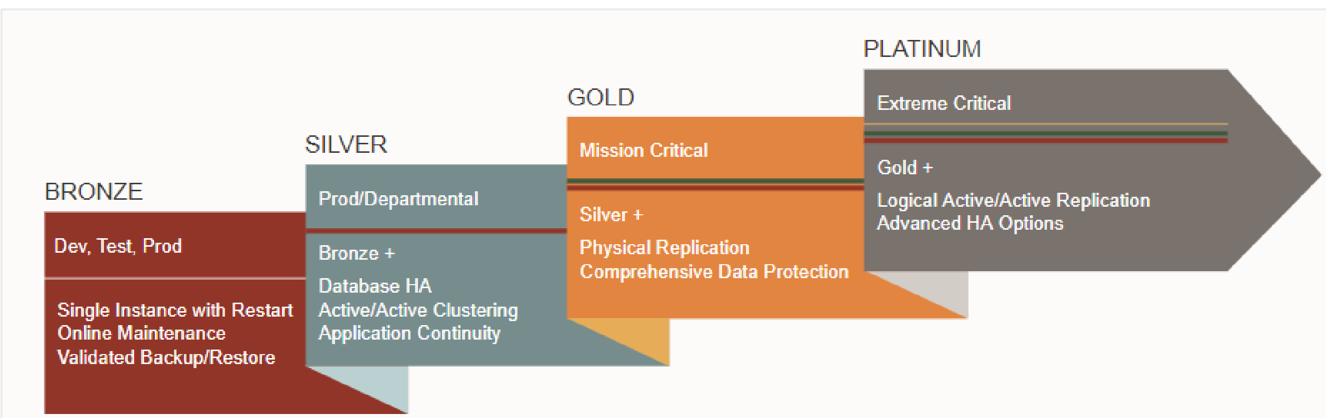


Figure 1: Oracle MAA Reference Architecture, from docs.oracle.com/en/database/oracle/oracle-database/21/haiad/index.html

Although this paper is not meant to cover MAA in detail, you must understand the concept of customer service tiers and the impact that it has on the consolidation plan as you transform the estate into the target architecture. As assets are converted to higher tiers, more hardware might be required, such as more Oracle RAC nodes, Data Guard standbys, and application tiers. This conversion should be part of your capacity modeling scenarios.

For more information about MMA, see Best Practices for Database Consolidation.

Security Zone and Isolation Alignment

Many customers have non-PCI, PCI, HIPAA, and other security classifications that preclude them from sharing infrastructure. Identify these requirements early so that assets can be classified and mapped to an appropriate security zone during the modeling and consolidation exercise. The following figure shows an example of how this can be achieved through the Exadata Cloud@Customer platform and outlines multiple VM clusters with VM subsetting.

Note: Figure 2 shows a cluster isolation example using Oracle Cloud@Customer. Exadata Cloud Service VM subsetting provides the ability to segregate workloads based on isolation requirements. You can achieve application and VM isolation by using tenancies, compartments, virtual cloud networks (VCNs), subnets, policies, and bare metal options in OCI as needed. For details, see “Cloud Environment Isolation (Zones).”

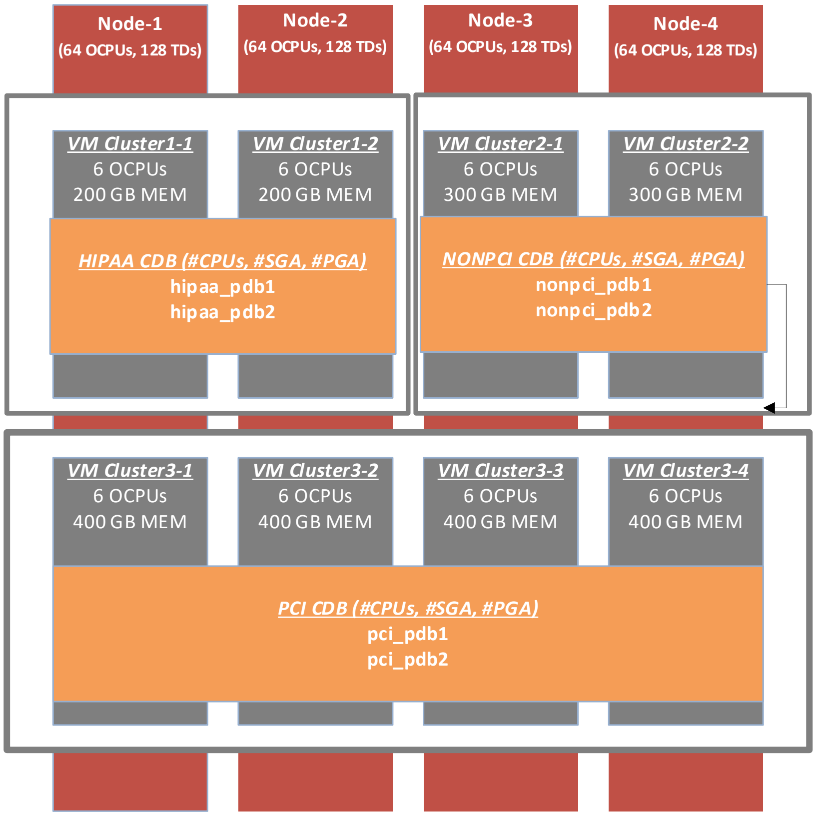


Figure 2: Exadata Cloud@Customer X9M Example Isolation Strategy Using VM Clustering

The figure shows VM cluster separation for PCI, non-PCI, and HIPAA, with the following configuration:

- Total memory: 5,560 GB
- Number of VM clusters: 3
- Number of compute servers (nodes): 4
- Total OCPU per compute node: 64
- Total memory per node: 1,390 GB

Operational Isolation

Many customers have requirements that prevent them from sharing software versions with other applications and strict requirements for patching and upgrade schedules. Such requirements impede the ability to share resources, and the resources must be isolated with dedicated capacity, which results in increased file system use for Oracle binaries. Identify these requirements early so that resources can be allocated with sufficient capacity during the consolidation exercise.

Cloud Environment Isolation (Zones)

Sometimes you're required to design isolation into a cloud architecture and consolidation plan. OCI is well-positioned to support this requirement. Resources such as regions, availability domains, tenancies, compartments, VCNs, security zones, tag-based access control, and network security groups enable you to comply with the most restrictive corporate and government IT security and isolation requirements.

When such requirements are identified, you must architect an appropriate security and isolation model to meet the needs of your customer. This paper doesn't cover this concept in detail. However, after you have defined this model, understand how the source assets map to the target architecture, and the consolidation rules used to map the assets. Understanding the OCI architecture, hierarchy, and security capabilities is critical to completing such a mapping and consolidation plan. The following list outlines how you can use these resources to maintain isolation in your consolidation plan:

- **Bare metal host**: OCI gives you control of the physical host (bare metal) machine. Bare metal compute instances run directly on bare metal servers without a hypervisor. When you provision a bare metal compute instance, you maintain sole control of the physical CPU, memory, and network interface card (NIC). Bare metal servers provide the highest level of server resource isolation and control in the consolidation plan.
- **Region**: A collection of availability domains in a single geographic location. Regions let you isolate assets to geographic locations in a consolidation plan.
- **Availability domain**: One or more isolated, fault-tolerant data centers that host cloud resources. A region contains one or more availability domains. Availability domains provide data center isolation in the consolidation plan.
- **Tenancy**: A root compartment that holds all of a customer's cloud resources. You can achieve isolation by deploying assets into multiple tenancies in the consolidation plan.
- **Compartment**: A collection of related resources (such as instances, VCNs, and block volumes) that can be accessed only by certain groups that have been given permission by an administrator. Think of a compartment as a logical group and not as a physical container. However, compartments alone can't provide a boundary that controls access to resources. Providing isolation requires a VCN that encompasses a compartment resource with the necessary security list, tags, and policies to restrict certain users or groups to those compartment resources.
- **Virtual cloud network (VCN)**: A virtual version of a traditional network that includes subnets, route tables, and gateways where instances run. A VCN resides within a single region but includes all the region's availability domains and is a critical item for security and isolation in a consolidation plan.
- **Tag-based access control**: Lets you write a policy with a set of conditions and tag variables to control access to a resource. Access can be controlled based on the requesting resource (group, dynamic group, or compartment) or on the target of the request (resource or compartment). Tag-based access control provides policy flexibility by letting you define access policies with tags that span compartments, groups (users), and resources (instances). It can also help you customize the isolation and consolidation plan.

The following diagrams show examples of how these resources can be deployed to help with isolation requirements, in a single-tenancy deployment and a multitenancy deployment.

Figure 3 shows a single-tenancy isolation example. HR and Sales are isolated by life cycle through compartments and tag-based access control, and administrator responsibilities are also separated. Such isolation allows separate source-to-target asset mapping. Tag-based access control can be applied to the requestor or target within the tenancy.

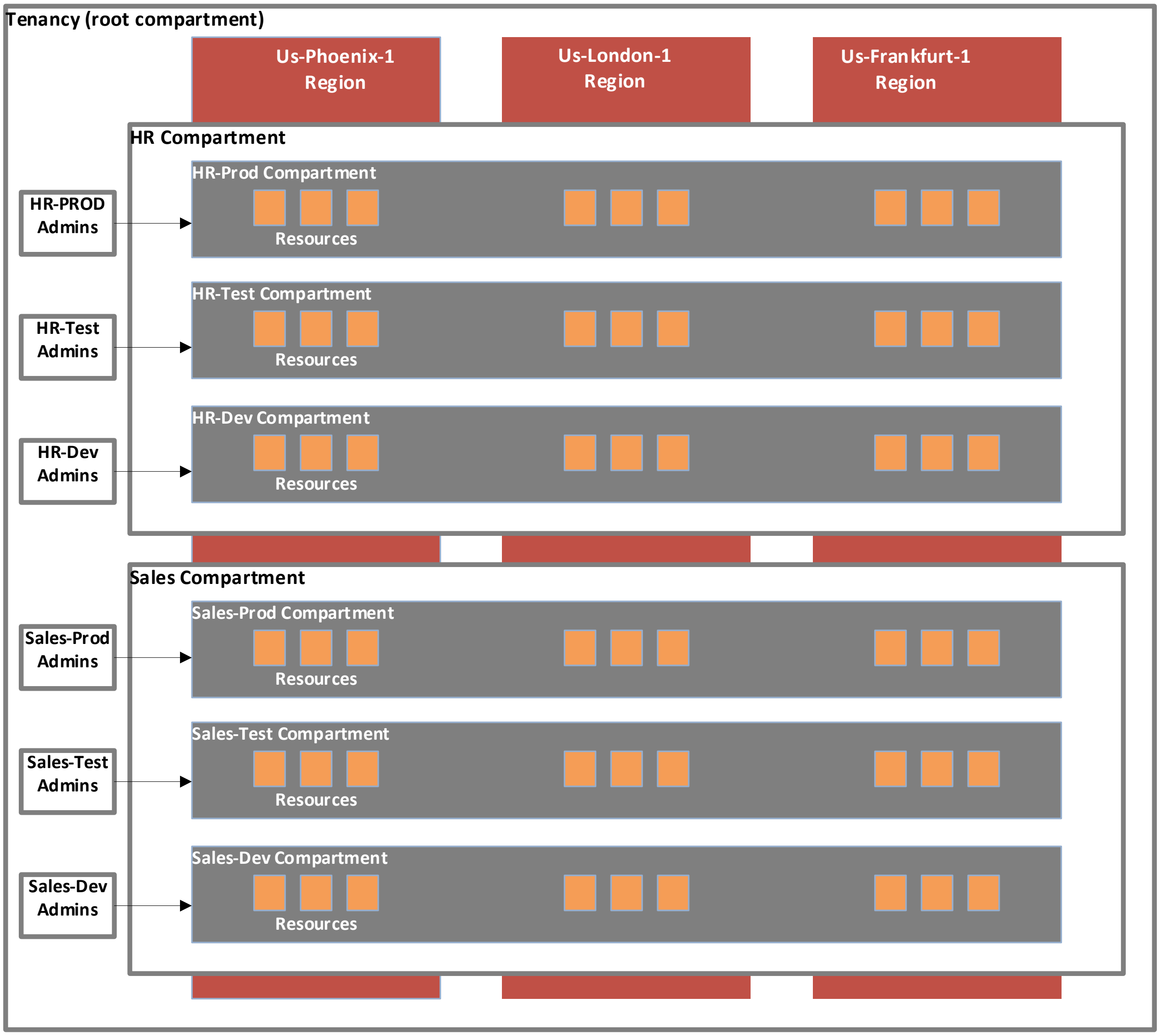


Figure 3: Single Tenancy Isolation Example

Figure 4 shows a multitenancy isolation example. HR and Sales are isolated by tenancy. Life cycle is isolated through compartments and tag-based access control in the tenancies, and administrator responsibilities are also separated in the tenancies. Such isolation allows separate source-to-target asset mapping. Tag-based access control can be applied to the requestor or target.

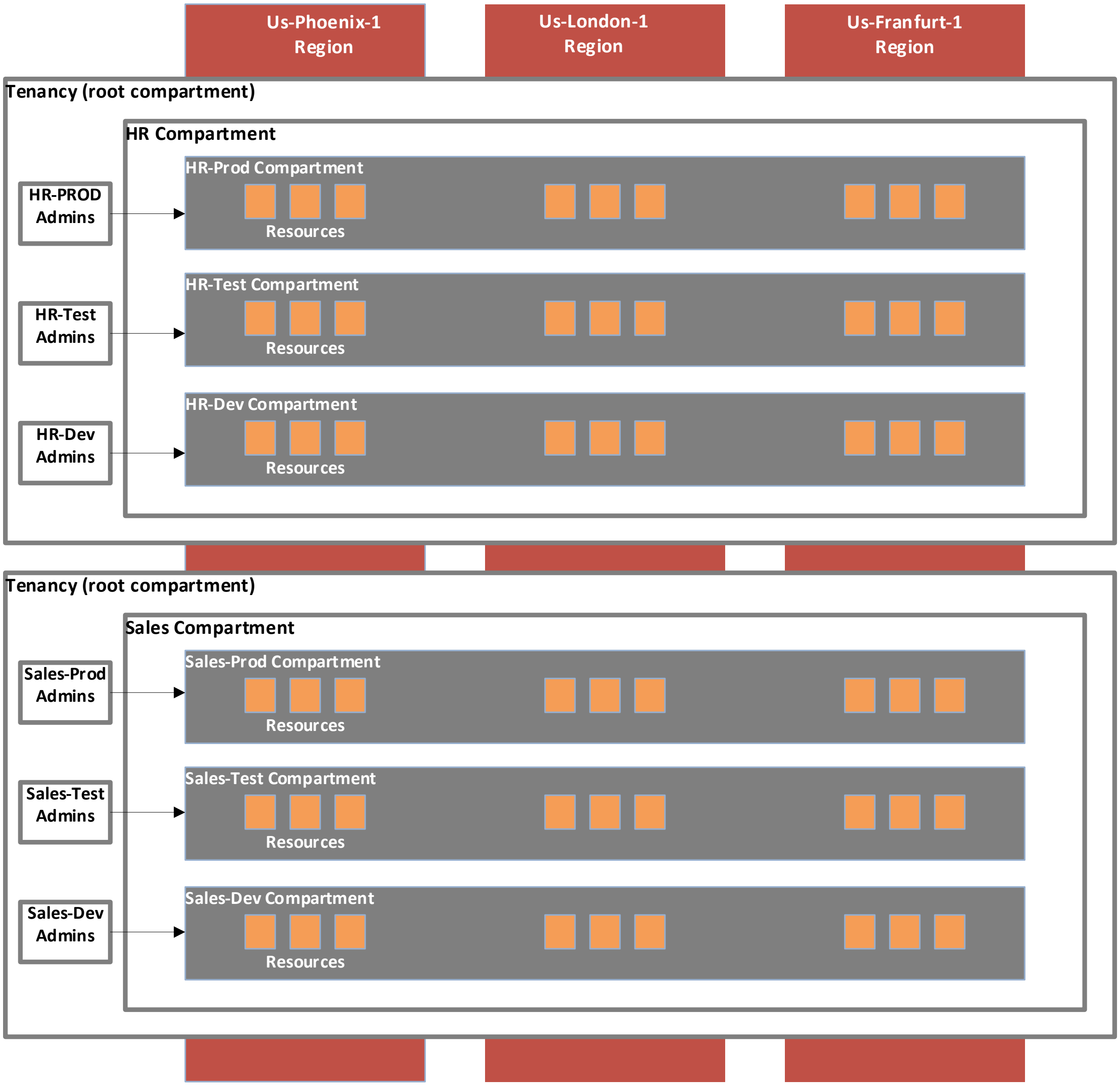


Figure 4: Multitenancy Isolation Example

Identify Consolidation Objectives, Strategy, and Rules

After all the assets are classified, you can begin the process of building an enterprise consolidation and deployment strategy plan. Discuss critical items, such as the following ones, with the customer to validate their expectations and educate them about Oracle technologies and how those technologies relate to the target architecture:

- Consolidation objectives
- Migration strategy
- Multitenancy conversion rules
- Oracle RAC conversion rules

The following sections cover these items in detail, with a focus on customers migrating from legacy environments to a new engineered system and cloud paradigm. However, these topics can be used for any migration.

Consolidation Objectives

Before you can create an enterprise consolidation and implementation plan, you must understand the customer's objectives for the target architecture. One customer might be interested in cost savings and obtaining the maximum consolidation density possible, while another might be interested in maximum performance. These objectives, combined with migration strategies and deployment architectures, can drastically impact the final consolidation plan.

Although these objectives can be created and altered in real time, you achieve maximum benefit by defining them *before* the migration project. Following are some example consolidation objectives:

- Maximum consolidation density, maximum performance, or a combination
- Security separation based on VMs, multitenant containers, or both
- Streamlined operational activities with reduced downtime

Migration Strategy

Choose the migration strategy that meets the customer's needs: rehosting, replatforming, rearchitecting.

Rehosting

During *rehosting*, also known as *lift-and-shift*, the source system is migrated to the target system with no changes introduced during the migration process. Many customers who want a fast migration with the least amount of risk to the business choose this strategy.

During this type of migration, the source and target service tier definitions and architectures remain the same. All transformations are moved to a later date, except for Transparent Data Encryption (TDE), which is mandatory for all OCI database targets. Moving to the target architecture as fast as possible is the main business driver for this strategy.

Replatforming

During *replatforming*, also known as *lift-improve-and-shift*, the source system is migrated to the target system with minor changes introduced during the migration process. Many customers who want to take advantage of the new features and benefits of the target architecture while performing an upgrade during the migration process with minimal risk choose this strategy.

Examples of this strategy include migration and upgrade to the OCI Database service to take advantage of the operational benefits of database as a service (DBaaS) or to Exadata Cloud@Customer to increase manageability and performance, and possibly convert to a single-tenancy database model. However, this strategy is reserved for customers who can perform minimal testing of their target systems before the final cutover and for whom migration speed is secondary to obtaining basic operational and upgrade benefits.

This strategy supports the following actions during migration:

- Upgrading the application and database to a later version
- Converting the database from a single instance to an active/passive Oracle RAC
- Converting the database from nonmultitenancy (standard Oracle database) to single tenancy (CDB with 1 PDB or tenant)

This strategy does *not* support the following actions during migration:

- Converting the database from a single instance to an active/active Oracle RAC
- Converting the database from nonmultitenancy to multitenancy (multiple databases sharing the same container)
- Converting the service tier (for example, Gold to Platinum)
- Integrating with the OCI Vault service
- Compression

Rearchitecting

During *rearchitecting*, also known as *transformation*, the source system is migrated to the target system with many changes, driven by extensive business requirements, that would be difficult to achieve in the current environment. Such changes include adding new features and increasing scalability, performance, and availability.

Examples of this strategy include migration and upgrade to the OCI Database service to take advantage of the operational benefits of DBaaS or to Exadata Cloud@Customer to increase manageability and performance, and convert to a multitenancy database model, scaling horizontally with Oracle RAC. This strategy is reserved for customers who can perform extensive testing of their target systems before the final cutover and for whom migration speed is secondary to fulfilling the requirements set for the new target system by the business.

This strategy supports the following actions during the migration, and additional options are possible:

- Upgrading the application and database to a later version
- Converting the database from single instance to active/active Oracle RAC
- Converting the database from nonmultitenancy to multitenancy (multiple databases sharing the same container)
- Converting the service tier (for example, Bronze to Silver, or Gold to Platinum)
- Integrating with the OCI Vault service
- Compression
- Database standardization
- Complex multitenancy consolidation based on business rules

Multitenancy Conversion Rules

IT organizations have traditionally used virtualization and clustering technologies to consolidate their databases, and many customers have attempted to consolidate their applications into database schemas. This effort has typically resulted in limited consolidation density, increased management costs, and high development costs.

Oracle Multitenant simplifies the consolidation process by plugging multiple databases (PDBs) into a multitenant container database (CDB) without changing applications. In this new architecture, memory and background processes are required only at the multitenant CDB level and, to a certain extent, are shared by all tenants of the CDB. This architecture enables IT organizations to achieve a greater level of scalability and consolidation density without compromising the performance, scalability, and security of their systems.

To achieve the maximum benefits of the multitenant architecture, develop a set of consolidation rules *before* the migration start date to provide an optimal target configuration for the consolidation databases. Following are some typical examples of multitenancy rules:

- All nonproduction systems are deployed as multitenant (many PDBs per CDB), and all production systems are deployed as single tenant (one PDB per CDB).
- All PDBs are deployed in a multitenant CDB based on environment and line of business or business unit.
- All nonproduction systems are deployed as multitenant with a maximum of 50 PDBs per CDB.

Note: In general, Oracle Multitenant doesn't have limits on how many PDBs can be deployed in a CDB. However, licensing and target architecture selection can impact limits and are covered in later sections of this paper. Limits should align with customer objectives and operational goals.

- All PDBs are deployed in a multitenant CDB based on service tiers (for example, Bronze, Silver, Gold, or Platinum).
- All PDBs are deployed in a multitenant CDB based on container configurations to include character sets (not required) and time zones to allow for common time-zone date values.
- All PDBs are deployed in a multitenant CDB based on security zones (for example, non-PCI, PCI, HIPAA, and SOX).
- All PDBs in a particular CDB can be patched in the same maintenance window.
- All PDBs in a particular CDB are of the same workload type: Data Warehouse (DW) or Online Transaction Processing (OLTP).
- All PDBs in a particular CDB have similar resource needs. Very large PDBs are not mixed with smaller ones, although small and medium PDBs could be mixed.

Because many rules can be used for a multitenant consolidation project, it's critical to identify them before migrations begin to achieve the customer's consolidation objectives. These rules are used to identify groups of databases into a concept called a *cohort* that eventually becomes a database container.

Oracle RAC Conversion Rules and Workload Management

Oracle Real Application Clusters (RAC) lets a single Oracle Database run across multiple servers to maximize availability and enable horizontal scalability while accessing shared storage. When you combine Oracle RAC with Oracle services, the workload can be isolated to any node of a cluster and is the first layer of a customer's workload management strategy. User sessions that connect to Oracle RAC instances can therefore fail over and safely replay changes during outages, without any changes to end-user applications, thus hiding the impact of the outages from end users.

When migrating to OCI, you can convert a single instance to Oracle RAC in the same operation. When doing so, you need to understand the consolidation ramifications and impacts on capacity and sizing. The main sizing configurations when making such a conversion are as follows:

- Active/active duplicated resource profile
- Active/active reduced resource profile
- Active/passive

The following sections explain these configurations in detail.

Active/Active Duplicated Resource Profile

The active/active duplicated resource profile duplicates the resources (CPU and memory) for every Oracle RAC instance. For example, a database that requires 8 OPCUs and a 16-GB SGA is deployed as two Oracle RAC nodes with 8 OPCUs and 16 GB of memory for each instance. If an instance fails, one Oracle RAC instance can instantly support the capacity requirements of the business.

To achieve this capability, you must duplicate the capacity reservations for the database, thus duplicating the resources in your capacity model.

Active/Active Reduced Resource Profile

The active/active reduced resource profile reduces the resources (CPU and memory) for every Oracle RAC instance by a certain percentage (for example, 50%). For example, a database that requires 8 OPCUs and a 16-GB SGA is deployed as two Oracle RAC nodes with 4 OPCUs and 8 GB of memory for each instance. If an instance fails or an operational procedure requires a failover to another node, one Oracle RAC instance can't support the peak capacity requirements of the business and performance degradation might occur. However, you aren't required to duplicate the capacity reservations for the database, thus reducing the resources in your capacity model.

One strategy to resolve such issues is to add a percentage increase in your consolidation plan to account for the temporary loss of an instance, using the following example:

- target SGA = current SGA / number of target Oracle RAC nodes + 25%
- target CPU = current CPU threads / number of target Oracle RAC nodes + 25%

Active/Passive

The active/passive configuration maintains the resource profile identified from the source database. For example, a database that requires 8 OPCUs and a 16-GB SGA is deployed as two Oracle RAC nodes with 8 OPCUs and 16 GB of memory for each instance. However, one of the instances remains in a stopped state to be started manually as needed. If an instance fails or an operational procedure requires a failover to another node, the other Oracle RAC instance is started and can support the peak capacity requirements of the business. However, you are not required to duplicate the capacity reservations for the database, thus reducing the resources in your capacity model.

Identify the Initial Target Architecture

Many architectures are available in OCI, and it's important to identify them early in the planning process so you can discuss and review their limits with the customer. After you identify the initial target architecture, understand the limits of that architecture and whether the infrastructure is comanaged or managed, and dedicated or shared:

- **Comanaged infrastructures** are managed by the customer and by Oracle—for example, nonautonomous Exadata Cloud@Customer.
- **Managed infrastructures** are managed completely by Oracle—for example, Autonomous Transaction Processing, Autonomous Data Warehouse, and Autonomous Database on Exadata Cloud@Customer.
- **Dedicated Exadata infrastructure** is dedicated to subscribing customers and is isolated from other tenants, with no shared processor, storage, or memory resources.
- **Shared Exadata infrastructure** is shared by multiple subscribing customers and is not isolated from other tenants, with shared processor, storage, and memory resources. The focus is on simplicity with a standardized configuration and lifecycle.

Autonomous Transaction Processing and Autonomous Data Warehouse

Autonomous Transaction Processing and Autonomous Data Warehouse are offered as dedicated and shared. You can have public or private endpoints, meaning public IP addresses for access from the public internet or private IP addresses limited to private network access. When you deploy an Autonomous Transaction Processing or Autonomous Data Warehouse database, you specify one of the following network access options:

- **Allow secure access from everywhere**: This option assigns a public endpoint, public IP address, and hostname to the database.
- **Virtual cloud network (VCN)**: This option assigns a private endpoint, private IP address, and hostname to the database. This option allows traffic only from the VCN that you specify; access to the database from all public IP addresses or VCNs is blocked.

With both options, some network planning is necessary. When you create a VCN and an IP address space, ensure that sufficient IP addresses are available to deploy all of your autonomous databases.

The Autonomous Transaction Processing database is optimized for transactional processing—tables are not compressed by default. The Autonomous Data Warehouse database is optimized for data warehousing—tables have Hybrid Columnar Compression (HCC) by default. Consider both of these points when identifying storage allocation in the consolidation plan.

Virtual Machine DB Systems

The OCI Database service runs on virtual machine (VM) DB systems and enables you to build, scale, and secure Oracle databases in the cloud. There are two types of DB systems on VMs:

- A 1-node VM DB system consists of one VM.
- A 2-node VM DB system consists of two VMs.

Both deployment models are offered with different shapes of up to 48 OCPUs. Each maintains a limit of one container database (CDB) per VM DB system. However, you could consolidate source databases as pluggable databases (PDBs) within the single CDB.

To create a DB system, you must first set up and configure the following network components:

- A VCN in the region where you want the DB system
- At least one subnet in the VCN (either public or private subnet):
    - Option 1: Public subnet with an internet gateway
    - Option 2: Private subnet

When planning for the DB system, you must address the requirements for IP address spaces within the consolidation plan. If you're setting up DB systems (and thus VCNs) in more than one region, ensure that the IP address spaces of the VCNs don't overlap. The subnet that you create for a bare metal or VM DB system can't overlap with 192.168.16.16/28, which is used by the Oracle Clusterware private interconnect on the database instance.

The following table lists the *minimum* required subnet size needed to deploy a DB system.

Note: The network services reserve three IP addresses in each subnet. Always allocate a larger IP address space than is required to deploy all DB systems that are targeted for the subnet in the consolidation plan. For the latest details on this topic, see VCNs and Subnets.

Table 6: IP Address Space Requirements for VM DB Systems

DB SYSTEM TYPE	NUMBER OF REQUIRED IP ADDRESSES	MINIMUM SUBNET SIZE
1-node bare metal or VM	1 + 3 reserved in the subnet = **4 x** the number of DB systems targeted for this subnet in the consolation plan	/30 (4 IP addresses)
2-node Oracle RAC VM	(2 addresses x 2 nodes) + 3 for SCANs + 3 reserved in subnet = **10 x** the number of DB systems targeted for this subnet in the consolation plan	/28 (16 IP addresses)

Exadata Cloud Service

Exadata Cloud Service is an Exadata system deployed in OCI. The latest model allows flexible configurations: you start with 2 compute nodes and 3 storage cells, and add 1 compute node or storage cell at a time up to 16 compute nodes and 32 storage cells.

Note: Always download the latest technical specification to understand such limitations, which evolve over time.

To create an Exadata Cloud Service instance, you must first set up and configure the following network components.

Note: Exadata Cloud Service instances use the Exadata Cloud Service resource model, which contains the Exadata Cloud Infrastructure and Cloud VM cluster resources. As such, networking is configured on the cloud VM cluster resource, compared to the DB system resource model in which networking is configured on the DB system resource.

- A VCN in the region where you want the DB system
- At least two subnets in the VCN:
  - Client subnet
  - Backup subnet
- Subnet option 1: Public client subnet with an internet gateway
- Subnet option 2: Private subnet

When planning for Exadata Cloud Service, address the requirements for IP address spaces within the consolidation plan. If you're setting up Exadata Cloud Service instances (and thus VCNs) in more than one region, ensure that the IP address spaces of the VCNs don't overlap. This is important if you want to set up disaster recovery with Oracle Data Guard.

The following table lists the *minimum* required subnet size needed to deploy an Exadata Cloud Service instance.

Note: The network services reserve three IP addresses in each subnet. Always allocate a larger IP address space than is required to deploy all DB systems that are targeted for the subnet in the consolidation plan. For the latest details on this topic, see Network Setup for Exadata Cloud Infrastructure Instances.

Table 7: IP Address Space Requirements for Exadata Cloud Service Instances

RACK SIZE	CLIENT SUBNET: NUMBER OF REQUIRED IP ADDRESSES	CLIENT SUBNET: MINIMUM SIZE	BACKUP SUBNET: NUMBER OF REQUIRED IP ADDRESSES	BACKUP SUBNET: MINIMUM SIZE
Base system or quarter rack	(4 addresses x 2 nodes) + 3 for SCANs + 3 reserved in subnet = 14	/28 (16 IP addresses)	(3 address x 2 nodes) + 3 reserved in subnet = 9	/28 (16 IP addresses)

RACK SIZE	CLIENT SUBNET: NUMBER OF REQUIRED IP ADDRESSES	CLIENT SUBNET: MINIMUM SIZE	BACKUP SUBNET: NUMBER OF REQUIRED IP ADDRESSES	BACKUP SUBNET: MINIMUM SIZE
Half rack	(4 x 4 nodes) + 3 + 3 = 22	/27 (32 IP addresses)	(3 x 4 nodes) + 3 = 15	/28 (16 IP addresses)
Full rack	(4 x 8 nodes) + 3 + 3 = 38	/26 (64 IP addresses)	(3 x 8 nodes) + 3 = 27	/27 (32 IP addresses)
Flexible infrastructure systems (X8M and higher)	4 addresses per database node (minimum of 2 nodes) + 3 for SCANs + 3 reserved in subnet	Minimum size determined by total number of IP addresses needed for database nodes	3 address per database node (minimum of 2 nodes) + 3 reserved in subnet	Minimum size determined by total number of IP addresses needed for database nodes

Exadata Cloud@Customer

Exadata Cloud@Customer is an Exadata system deployed in a customer data center, but Oracle manages the hardware. In the comanaged model, customers manage the VM clusters, grid infrastructure, and databases. In the managed model, the customer gets a dedicated Autonomous Database system in their data center, and Oracle manages the hardware.

To connect to the customer's corporate network, Exadata Cloud@Customer requires several host names and IP addresses for network interfaces on the client network and the backup network. The precise number of IP addresses depends on the Exadata system shape. These network configuration details, including host names and IP addresses, are specified when a VM cluster network is created. All IP addresses must be statically assigned IP addresses, not dynamically assigned (DHCP) addresses. The client network and the backup network require separate subnets.

The following table outlines the IP address requirements for the client and backup networks. The table specifies the maximum and recommended CIDR block prefix length for each network. The maximum CIDR block prefix length defines the smallest block of IP addresses that are required for the network. To allow for possible future expansion within Exadata Cloud@Customer, we recommend a smaller CIDR block prefix length, which reserves more IP addresses for the network. These details must be outlined with the customer to guarantee that sufficient IP address spaces are deployed for the number of Exadata Cloud@Customer frames deployed in the consolidation plan. For the latest details, see IP Addresses and Subnets for Exadata Database Service on Cloud@Customer.

Table 8: IP Address Space Requirements for Exadata Cloud @Customer

NETWORK	IP ADDRESS REQUIREMENTS FOR BASE SYSTEM, QUARTER RACK, OR HALF RACK	IP ADDRESS REQUIREMENTS FOR FULL RACK
Client network	• Maximum: /28 • Recommended: /27	• Maximum: /27 • Recommended: /26
Backup network	• Maximum: /29 • Recommended: /28	• Maximum: /28 • Recommended: /27

Compute Capacity

OCI provides fast, flexible, and affordable compute capacity to fit any workload need, from performant bare metal servers and VMs to lightweight containers. You can use this capacity for applications and databases in various configurations after it has been aligned with the necessary size and shape.

To launch a compute instance, you must first set up and configure the following network components:

- A VCN in the region where you want the compute instance
- At least one subnet in the VCN (either public or private subnet)
  - Option 1: Public subnet with an internet gateway
  - Option 2: Private subnet

When planning for the compute deployment, address the requirements for IP address spaces within the consolidation plan. For more information, see Best Practices for Your Compute Instances.

Identify Target Resource Limits

After you identify the initial target architecture, identify the limits for the target resources.

VM and Bare Metal DB Systems Resource Limits

Review the following sections for VM and database resource limits.

Single-Instance and Oracle RAC Limits

All single-node Oracle RAC DB systems support the following Oracle database editions:

- Standard Edition
- Enterprise Edition
- Enterprise Edition - High Performance
- Enterprise Edition - Extreme Performance

Two-node Oracle RAC DB systems require Oracle Enterprise Edition - Extreme Performance.

Shape and Storage Limits

OCI offers single-node DB systems on either bare metal or VMs, and 2-node Oracle RAC DB systems on VMs.

When you create a VM DB system, you choose a shape, which determines the resources allocated to the DB system. After you provision the system, you can change the shape to adapt to new processing capacity requirements. Because each DB system has only one CDB, CPU oversubscription isn't necessary.

The following shapes are available for bare metal and VM DB systems. For the latest information about these sizes and shapes, as well as network and storage details, see Compute Shapes.

Table 9: Example VM and Bare Metal Shapes

SHAPE	OCPU	MEMORY (GB)	MAXIMUM NUMBER OF RAC NODES	STORAGE OPTIONS
VM.Standard2.1	1	15	2	712–49,352 (GB)
VM.Standard2.2	2	30	2	712–49,352 (GB)
VM.Standard2.4	4	60	2	712–49,352 (GB)

SHAPE	OCPU	MEMORY (GB)	MAXIMUM NUMBER OF RAC NODES	STORAGE OPTIONS
VM.Standard2.8	8	120	2	712–49,352 (GB)
VM.Standard2.16	16	240	2	712–49,352 (GB)
VM.Standard2.24	24	320	2	712–49,352 (GB)
BM.DenseIO2.52	52	768	1	51.2 (TB) NVMe
BM.DenseIO1.36	36	512	1	28.8 (TB) NVMe

PDB Limits

With bare metal and VM DB systems, identify the number of pluggable databases (PDBs) allowed per container database (CDB) before performing a capacity modeling exercise. The following table outlines these limits in detail. For the latest information, see the documentation.

Table 10: Nonautonomous PDB Limits

DATABASE VERSION	STANDARD EDITION	ENTERPRISE EDITION	ENTERPRISE EDITION (EXTRA COST)
Oracle 18c (number of PDBs)	1	252	4096
Oracle 19c (number of PDBs)	3	252	4096

Comanaged Resource Limits

Review the horizontal and vertical scaling limits for your comanaged resources and align them with the consolidation rules and best practices outlined in this paper. This section outlines the limits of Oracle's comanaged solutions: Exadata Cloud@Customer and Exadata Cloud Service. To help you understand them, see the technical data sheets before initiating the migration project. Examples limits are as follows:

- The Exadata Cloud Service X8M architecture can scale vertically to 100 OCPUs with a quarter rack and vertically to 32 database servers and 64 storage servers. See the Oracle Exadata Cloud Service X8M data sheet.
- The Exadata Cloud@Customer X9M architecture has 5,560 GB of memory on a half rack with 4 database servers and can scale to 8 databases servers and 12 storage servers. See the Oracle Exadata Cloud@Customer X9M data sheet.

Virtual Machine CPU Limits and Oversubscription for Exadata Cloud@Customer and Exadata Cloud Service

As with Exadata Cloud@Customer VM cluster oversubscription, the ability to oversubscribe CPU within the VM at the database level while using Oracle Resource Manager (instance caging) lets you increase consolidation density and the overall CPU usage of the compute node and database resources. This allows for better use of compute node CPU resources when some databases are busy and others are not. However, as with VMs, CPU oversubscription at the database level forces physical CPU resources to be time-shared during busy periods. The introduction of Oracle Resource Manager at this point is critical in protecting against the noisy neighbor scenario.

For example, consider an Exadata Cloud@Customer instance with 24 CPU cores enabled in a quarter rack with one VM cluster:

- This configuration results in a VM cluster with two VMs, each with 12 OCPUs and 24 VCPUs.
- Deploying one database to this VM cluster that requires 24 OCPU cores and 48 VCPU threads results in an oversubscription factor of 1x (no oversubscription).
- Deploying two databases to this VM cluster that require 48 OCPU cores and 96 VCPU threads, but turning on only 24 OCPUs, results in an oversubscription factor of 2x (oversubscription).
- In the two-database scenario, deploy Oracle Resource Manager and instance caging to allow for CPU resource sharing and provide for a maximum return on investment.

At this point, determine reasonable oversubscription factors and align them with the Consolidation Objectives identified during this phase. Note the following important points:

- The higher the oversubscription factor, the higher the consolidation density. The higher the consolidation density, the greater the return on investment. However, the higher the consolidation density, the higher the risk of CPU throttling by Oracle Resource Manager and perceived performance problems.
- The lower the oversubscription factor, the lower the consolidation density. The lower the consolidation density, the lower the return on investment because of idle hardware.

The following table provides common values as starting point for the database oversubscription factors. During the phases of the migration project, constantly evaluate these values and review performance.

Table 11: Example CPU Oversubscription Factors

ENVIRONMENT	COMMON CPU OVERSUBSCRIPTION FACTORS	DESCRIPTION
Nonproduction	3x	Allows for a higher consolidation density for nonproduction assets, which traditionally don't have the resource requirements of their production counterparts. This factor should never exceed 5x.
Production	2x	Allows for average consolidation density for production assets while maintaining sufficient headroom for unplanned production workload activity. This factor should never exceed 3x.

Memory Optimization for Exadata Cloud@Customer and Exadata Cloud Service

As discussed in the CPU oversubscription sections in this paper, CPU can be oversubscribed and the level of oversubscription can be manipulated. However, memory is a finite resource and must be managed to maintain optimal performance on the target architecture. That being said, you can increase or decrease the VM cluster memory on the Exadata Cloud@Customer platform with a rolling downtime. The huge page percentage of allocated memory remains constant when you change the memory allocation. However, you should understand the physical limits of the memory resources and provide enough free space to allow for growth within it.

Details about the memory configuration of an Exadata Cloud@Customer frame are provided in the technical data sheet for the deployed model. Example X9M details are provided in "Appendix A: Exadata Cloud@Customer X9M Technical Data Sheet."

After obtaining these details, understand the following memory pool concepts and limits before deploying VM clusters:

- total frame memory = total memory available for guest VMs (GB) per quarter, half, and full rack
- number of VM clusters = necessary number of VM clusters to meet security separation, performance, and availability targets
- database server memory = total memory per VM within the VM cluster
- huge page allocation percentage = percentage of huge page memory configured within the database server memory reserved for all database SGA memory pools. Note: the default is 50% of memory.
- huge page allocation percentage free = percentage of free space within the huge page allocation that is set aside for SGA growth

The following table outlines common values as a starting point for database memory pool limits and reservations factors. During the phases of the migration project, constantly evaluate these values and review performance.

Table 12: Example Memory Limits and Percentages

ENVIRONMENT	HUGE PAGE ALLOCATION PERCENTAGE (% OF DATABASE SERVER MEMORY RESERVED FOR HUGE PAGES)	HUGE PAGE ALLOCATION PERCENTAGE FREE (% OF HUGE PAGE MEMORY WITH POOL RESERVED FOR GROWTH)
Nonproduction	60–70%	20%
Production	60%	20%

For example, consider a 2-node production Exadata Cloud@Customer VM cluster on a quarter rack assigned 200 GB of memory:

- This configuration results in a VM cluster with two VMs, each with 100 GB of database server memory.
- Given a 60% allocation, each VM contains 60 GB of memory for its huge page allocation and 40 GB of memory for PGA.
- Of the 60 GB of memory allocated for huge pages, 20 GB is reserved for growth before starting migration.

You can alter these values and percentages to meet the level of risk, growth, and consolidation density needed, as follows:

- Consolidation density, rather than performance, is typically the driving factor for nonproduction systems. As such, the huge page allocation can extend beyond 60% depending on PGA and user process requirements.
- Performance, rather than consolidation density, is typically the driving factor for production systems. As such, the huge page allocation shouldn't extend beyond 60% in most cases.

Review these items in detail before deploying databases onto the VM clusters.

ASM Storage Optimization for Exadata Cloud@Customer and Exadata Cloud Service

Just as memory is considered a finite resource, storage should also be managed to maintain balanced capacity on the target architecture. As with memory, VM storage under Automatic Storage Management (ASM) can be dynamically expanded based on need. However, you should understand the physical limits of the storage resources and provide enough free space to allow for growth within the allocation of the VM cluster to prevent unnecessary resizing operations.

Obtain the following details to identify the optimal storage allocation:

- Identify the ratio of storage between the DATA disk group and the RECO (recovery) disk group
  - If local backups are not required, the ratio is 80:20. This is the typical choice when Exadata Cloud@Customer backups are made to external storage or tape, or Exadata Cloud Service backups are made to Object Storage.
  - If local backups to Exadata storage are required, the ratio is 40:60.
- Identify rack size and total storage footprint (quarter, half, or full) by looking up the "total usable disk capacity" details on the storage configuration of the Exadata Cloud@Customer frame, which are provided in the data sheet for the deployed model. An example of an X9M frame is shown in "Appendix A: Exadata Cloud@Customer X9M Technical Data Sheet."
- Establish a level of free space within the DATA and RECO disk groups that depends on growth rates and level of risk. Acceptable values typically range from 60% to 80% used and must be identified before running capacity models and the start of migration.

Note: This information assumes that there is no requirement for a SPARSE disk group to support Exadata sparse clones.

File System Storage Optimization (/u02) for Exadata Cloud@Customer

As you progress through consolidation planning, you must identify how much local storage space is provisioned for each VM. This space is mounted at location /u02 and is used primarily for Oracle Database homes. Define the number of Oracle Database homes for the project. The amount of available local storage varies with the number of VMs running on each physical node because each VM requires a fixed amount of storage for the root file systems, Grid Infrastructure homes, and diagnostic log space. For storage limit details, see Manage VM Clusters.

To continue with the X9M-2 Exadata Cloud@Customer example, the total space available to all VMs on an Exadata Cloud@Customer X9M database node is 2,243 GB. Although each database node has 2,243 GB, for a single VM, you can allocate a maximum of 900 GB of local storage. Similarly, for a second VM, 1,800 GB of local storage is available given the maximum limit of 900 GB per VM. For a third VM, the amount of available space is 2,243 - (184-GB fixed overhead x 3) = 1691 GB. And so on for four and more VMs.

X9M-2 systems:

- Total available for VM images (base system): 1,077 GB
- Total available for VM images (quarter, half, or full racks): 2,243 GB
- Fixed overhead per VM: 184 GB

Table 13: Example x9M /u02 Storage Limits

# VMS	FIXED STORAGE ALL VMS (GB)	X9M-2 BASE SYSTEM ALL /U02 (GB)	X9M-2 QUARTER/HALF/FULL RACKS ALL /U02 (GB)
1	184	892	900
2	368	708	1800
3	552	524	1691
4	736	340	1507
5	920	Not applicable	1323

# VMS	FIXED STORAGE ALL VMS (GB)	X9M-2 BASE SYSTEM ALL /U02 (GB)	X9M-2 QUARTER/HALF/FULL RACKS ALL /U02 (GB)
6	1104	Not applicable	1139
7	1288	Not applicable	955
8	1472	Not applicable	771

As you plan for the Exadata Cloud@Customer implementation, the number of Oracle Database homes that you plan to deploy to each VM cluster should not exceed the space available on /u02. Consider estimating 40 GB per Oracle Database home.

Managed Resource Limits

Review the limits of the managed resources in the migration project. Managed resource limits are maintained and managed by Oracle and attributed to the vast automation of the Oracle Autonomous Platform. However, you must follow a few limits for the migration project and align them with the consolidation best practices outlined in this paper. The following sections outline the managed limits of the target architecture.

Limits for Autonomous Database on Exadata Cloud@Customer (Database CPU and Scaling)

Unlike Exadata Cloud@Customer comanaged databases (nonautonomous), the CPU limits and scaling of Exadata Cloud@Customer managed databases (autonomous) are managed automatically by Oracle through the autoscaling feature. For consolidation, this feature is a key differentiator because it enables an autonomous database to use up to three times more CPU and IO resources than its allocated OCPU count. For Exadata Cloud@Customer, fractional OCPUs are supported, and if three times the OCPU count results in a fractional value, it's rounded to the next whole number. For example:

- 0.1 OCPU can automatically scale up to one OCPU
- 0.5 OCPU can automatically scale up to two OCPUs
- 0.7 OCPU can automatically scale up to three OCPUs

To ensure that no single autonomous database can automatically scale up and consume all OCPUs available in the deployment pool, Autonomous Database uses the autonomous container database (CDB) as a limiting factor. During a scaling operation, the following items are considered:

- Autonomous Database compares the count of OCPUs available in its parent CDB to the number of OCPUs in use across all the autonomous databases in the parent CDB. If unused OCPUs are available, the database is scaled up. If the OCPUs are in use and unavailable, it is not.
- If the OCPU used to automatically scale an autonomous database came from another lightly loaded running autonomous database not using all of its allocated OCPUs, Autonomous Database automatically scales the autoscaled database down to manage overall capacity if the load increases on the other database and it needs its allocated OCPU back.

Therefore, to properly develop and manage your consolidation project, you must understand the OCPU allocation and limits of the CDBs, the number of PDBs in them, and the resources defined in the technical data sheet for the target architecture. The main areas of importance are outlined in the following table, referenced from [Introduction to ADB-D on Exadata Cloud@Customer](#).

Table 14: Example X9M Autonomous Container and Database Limits

RESOURCE	QUARTER RACK	HALF RACK	FULL RACK
Autonomous databases per rack	1000 (920 on Exadata X7 system)	2000 (1840 on Exadata X7 system)	4000 (3680 on Exadata X7 system)
Autonomous CDBs per rack	12	12	12
Autonomous databases per CDB	200	200	200
Autonomous databases per CDB with Autonomous Data Guard	25	25	25

Limits for Autonomous Database on Exadata Cloud Service (Database CPU Limits and Scaling)

Autonomous Database on Exadata Cloud Service is a "serverless" platform, with no requirement to deploy infrastructure or servers, or install database software. The Autonomous Database service is a clustered service with resilience built into every deployment. There is no requirement to presize databases on Autonomous Database; databases can grow or shrink on demand.

Autonomous Database offers the following complementary methods of scaling an autonomous database instance.

Fully Elastic Manual Online Scaling

This primary scaling method sets the base size of the autonomous database. The database is created with a user-specified number of CPUs and amount of storage, which sets the starter size of the database. All other parameters for the database are derived from those two input parameters.

After provisioning, you can scale the number of cores or the storage capacity of the database at any time without impacting availability or performance. This action is governed by the OCPU_COUNT parameter. You can change this parameter by using the OCI Console, the OCI CLI, a REST call, or one of the SDKs. This scaling method allows the base size of the database to be changed up or down at the full control of the customer and to be automated as required.

Note that the maximum number of cores that are available to any Autonomous Database shared database is 128.

Because you can scale the number of CPU cores or the storage capacity of the database at any time without impacting availability or performance, detailed architecture designs aren't required and are fully managed by Oracle.

CPU/IO Autoscaling

Autonomous Database autoscaling is designed to allow the database to scale automatically and with no service impact for short periods. It scales CPU and IO but doesn't alter the baseline size of the database, so other parameters aren't changed. For example, the scaling doesn't increase the number of sessions or the size of the storage.

- OCPU autoscaling lets the database use up to three times more CPU and IO resources, depending on workload requirements. OCPU autoscaling is enabled by default when a database is created.
- Storage autoscaling lets the database expand to use up to three times the reserved base storage, depending on storage requirements. Storage autoscaling is disabled by default when a database is created.
- You can manage scaling from the OCI Console by selecting or clearing the **OCPU auto scaling** and **Storage auto scaling** options.
- The maximum number of cores that are available to any autonomous database is 128, regardless of whether autoscaling is enabled. This means that a database with a CPU core count of 64 could automatically scale up to two times the assigned number of cores (2 x 64 = 128). A database with 42 cores (or fewer) could automatically scale up to three times the assigned number (3 x 42 = 126). For billing purposes, the database service determines the average number of CPUs used per hour.

Because the infrastructure is completely managed by Oracle, you don't need to closely watch limits and plan for them. However, when you're deploying Autonomous Database to Exadata Cloud Service, the following deployment options provide scalability limits and are related to Autonomous Database shared and dedicated.

Autonomous Database Shared

For the shared option, databases are deployed into the public cloud into a large pool of databases entirely managed by Oracle. You don't need to manage how databases are grouped or to specify which hardware resources are used to deploy the service. This service requires the least customer maintenance. Autonomous Database shared databases are deployed, managed, and charged at the database level and can scale to the maximum values per autonomous database.

Autonomous Database Dedicated

For the dedicated option, you're allocated specific infrastructure in Oracle's database cloud. You can then deploy from 1 to 12 autonomous container databases (CDBs) on this hardware, which act as "pools" of PDBs. When an autonomous database is subsequently provisioned, you can choose which CDB to provision it in.

- Autonomous Database dedicated databases are deployed and managed at the database level, similar to Autonomous Database shared databases, but customers are charged for the dedicated infrastructure (regardless of whether it's used).
- Dedicated hardware is deployed in units of quarter, half, or full rack. For up-to-date details about the sizes available, see Resource Limits and Characteristics of Infrastructure Shapes.

Define the Resource Management Strategy (Comanaged)

After you model the consolidation strategy, deploy a resource management plan as the last step in optimizing the consolidation strategy and protecting the customer's assets from noisy neighbors.

Container Scalability: Instance Caging

When deploying customer assets to a container, configure instance caging to guarantee that each asset has its required CPU allocation while in a single tenancy or multitenancy deployment. Use the following commands to configure instance caging and increase capacity for an existing container.

Note: Most OCI targets have two threads per OCPU. When you set the `CPU_COUNT` in a container, you specify the maximum number of vCPUs that the container can use. You can set this to fractional values in the instance caging strategy. For more information about OCPU and vCPU, see the "CPU Threads (vCPUs)" section earlier in this paper.

Configure Instance Caging

To deploy instance caging, run the following command at the CDB level. If you haven't deployed Oracle Resource Manager already, the simplest solution is to use `default plan` as shown here. If you intend to customize the default plan, use a nondefault plan so the default plan can be used for other activities and remains unchanged.

```
alter system set resource_manager_plan = default_plan;

alter system set cpu_count = X scope = both; X=# of VCPUs+
```

Add More vCPUs to an Existing Container

To add more vCPU threads by to an existing container, run the following command:

```
alter system set cpu_count = X scope = both;  X=# of VCPUs
```

Container Scalability: PDB Shares

When deploying customer assets to a multitenancy (multiple-customer) container, you can enable PDB shares to prevent one customer from using all the available resources and to provide a share of CPU capacity to every PDB. As with instance caging, PDB shares can be enabled at the vCPU level, and it operates using shares and limits. Each instance of a CDB is given an amount of vCPU to use by enabling database resource management (DBRM) and setting `CPU_COUNT`. The PDBs within that CDB are then given "shares" of the vCPU available to the CDB. Each PDB then receives the designated share of CPU resources, and the system is not oversubscribed.

Note: The rest of the section outlines an example of this and was taken from the [Dynamic CPU Scaling in Oracle Database 19c] blog post.

Each PDB can also be assigned a limit of CPU resources that it can use, which serves to prevent wide swings in database performance. Without a limit imposed on a PDB, each database can use the full amount of all available CPU on the system, which can appear to users as a large variation in performance. Shares are expressed as a share value, whereas usage limits are expressed as a percentage, as shown in the following table.

Table 15: Example PDB Shares and Usage

PLUGGABLE DATABASE	SHARES	SHARE %	USAGE LIMIT
PDB1	1	10%	20%
PDB2	2	20%	40%
PDB3	2	20%	40%
PDB4	5	50%	90%
Total	**10**	**100%**	

Using CPU minimum and maximum ranges, you can establish the lower and upper boundaries of vCPU available to each PDB to prevent wide swings in resources. Each PDB receives a guaranteed minimum amount of vCPU but can also automatically scale up to a maximum level. This feature is controlled by two parameters in each PDB: `CPU_MIN_COUNT` and `CPU_COUNT`. For details about dynamic CPU scaling with Database Resource Management (DBRM), see the [Dynamic CPU Scaling in Oracle Database 19c] blog post.

- `CPU_MIN_COUNT` is the minimum number of vCPUs that the PDB instance receives. The total of `CPU_MIN_COUNT` for all PDB instances should not exceed the `CPU_COUNT` of the CDB instance.
- When DBRM is enabled and `CPU_MIN_COUNT` has been set, the `CPU_COUNT` parameter defines the maximum number of vCPUs that a PDB instance can use.

We recommend creating a standardized ratio between `CPU_MIN_COUNT` and `CPU_COUNT`. For example, set `CPU_COUNT` equal to 3 times the value of `CPU_MIN_COUNT`. This means that Oracle Database vCPU usage would run within a 3x range from `MIN` to `MAX`.

As a workload is scaled up, `CPU_COUNT` and PDB shares can be expanded for each CDB and PDB by using the following example to enable DBRM.

Note: Resource Manager is enabled at the CDB level by setting the RESOURCE_MANAGER_PLAN at the root level to the name of a CDB resource plan. If the CDB resource plan has no configured CPU directives, that is, the SHARES and UTILIZATION_LIMIT directives are unset, then Resource Manager uses the CPU_COUNT and CPU_MIN_COUNT settings for the PDB to manage CPU utilization.

---

1. Connect to the container and set the minimum CPU.

```
conn / as sysdba
alter system set resource_manager_plan = <Client>_plan;
alter system set cpu_min_count="0.5"; - Container
```

2. Switch to PDB1.

```
alter session set container = pdb1;
```

3. Set the PDB-specific values.

```
alter system set cpu_min_count=1;
alter system set cpu_count=2;
```

4. Switch to PDB2.

```
alter session set container = pdb2;
```

5. Set the PDB-specific values.

```
alter system set cpu_min_count=2;
alter system set cpu_count=4;
```

The following diagram is a graphical example of how this strategy would look in a consolidated Exadata platform with containers running on nodes 7 and 8. Shares represent vCPU counts, and utilization limits can be added to set the maximum amount of CPU used by the PDB if required.

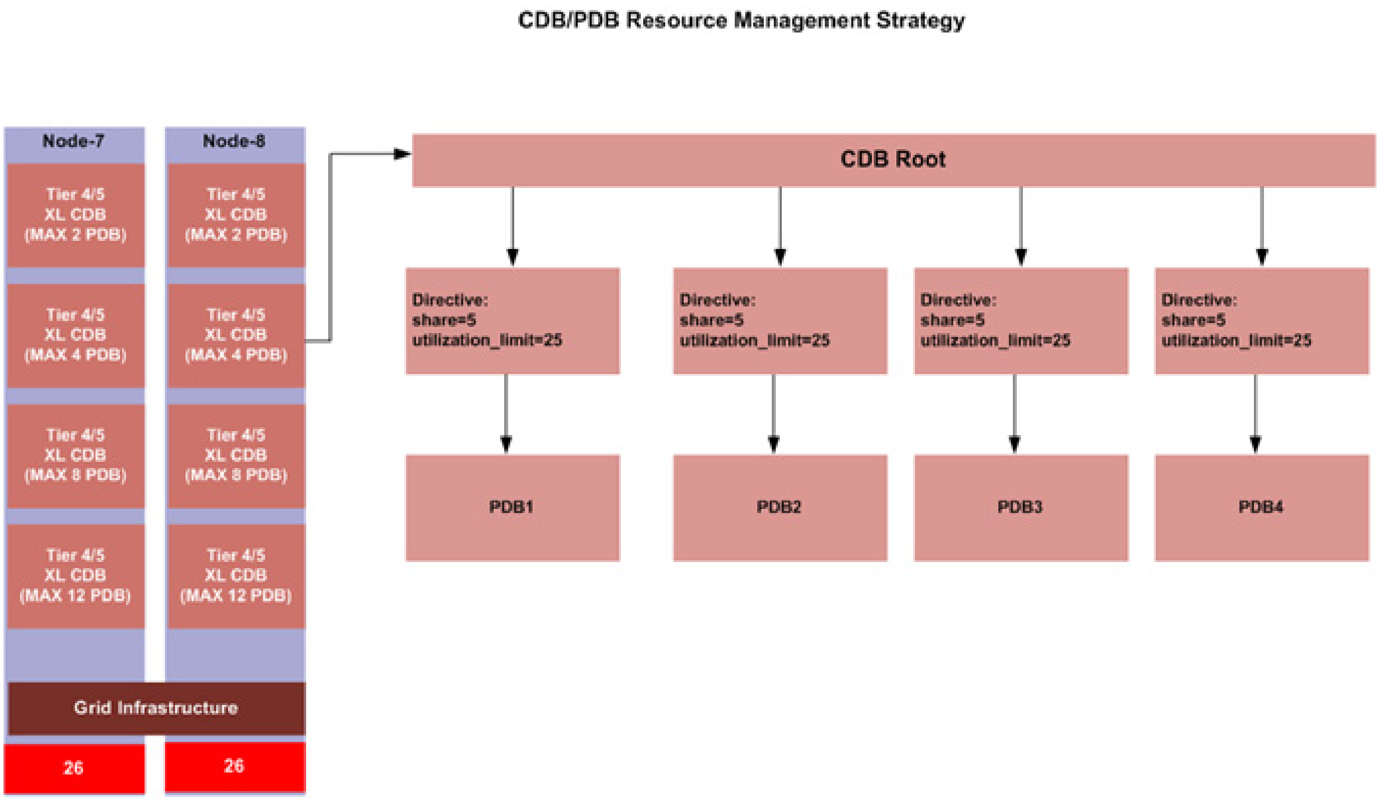


Figure 5: CDB Resource Management Strategy

Run a Modeling Scenario (Migrate 360 Case Study)

This section documents an example case study used to build a complete consolidation plan for a large number of workloads (996) using the Migrate 360 platform. Using the strategy outlined in this document, this model consolidation plan includes the following steps:

- Estate discovery (inventory)
- Estate resource usage (reverse engineering)
- Estate asset classification and assignments
- Identifying consolidation objectives, strategy, and rules
- Identifying the initial target architecture
- Identifying target resource limits
- Modeling the scenario (capacity analysis)

Estate Discovery (Inventory)

The customer identified the following business requirements for this model:

- They must reduce operational expenditures and run and maintain costs such as patching, upgrades, and tuning for the Finance line of business (LOB). They plan to migrate 101 databases to Oracle Autonomous Transaction Processing.
- They identified 340 databases with strict data isolation, data sovereignty, and security issues for the Sales LOB that must remain in the customer data center and behind customer firewalls. They plan to migrate these databases to Exadata Cloud@Customer.
- They identified 202 critical databases from the Customer Care LOB with multiple workload types and resource (CPU, memory) isolation and performance requirements. They plan to migrate these databases to Exadata Cloud Service.
- They require a cost-effective solution to easily build, scale, and secure database workloads and have identified 202 databases across the Online Banking LOB to be deployed to the OCI Database service and VM DB systems.
- They identified 151 departmental databases in the Marketing LOB to migrate to Autonomous Data Warehouse.
- They deployed 5 application servers that connect to *all* LOBs from a third-party cloud using Kubernetes Engine that they plan to migrate to OCI. The servers are expected to dynamically scale up to 300 compute servers as customer load increases and connect to databases to be deployed on Exadata Cloud Service, Autonomous Transaction Processing, Autonomous Data Warehouse, and VM DB systems.

The technical requirements for this model are as follows:

- For the identified estate of 996 assets (databases), all targets should maintain 20% free CPU, 30% free memory, and 40% free disk space for growth after migration.
- Databases targeted for Exadata Cloud@Customer will be deployed in separate non-PCI and PCI security network zones onto 12 X9M Exadata Cloud@Customer racks to secure customer data.
- All source databases are single instance and will be converted to Oracle RAC and distributed across two nodes in an active/active configuration.

- All source databases should be standardized into common sizes and shapes, which include small, medium, and large.
- All source databases are Oracle Database version 11.2.04 and should be upgraded to 19.2.
- The application servers will be deployed on VMs (compute) in OCI in a private subnet suitable to handle up to 300 compute servers.

Estate Resource Usage (Reverse Engineering)

The 996 databases are added to the Migrate 360 database inventory load sheet, a portion of which is shown in the following figure. After data is loaded into Migrate 360, all the assets are reverse-engineered from all source databases via Oracle Enterprise Manager by using the SQL queries in Table 1: CPU Metric Queries, Table 2: SGA and PGA Metric Queries, and Table 3: Database Size Metric Queries.

LINE_OF_BUSINESS	DATA_CENTER	APP_NAME	DB_SERVICE_TIER	HOSTNAME	CNDB_NAME	ENVIRONMENT	NLS_CHARACTERSET	SECURITY_ZONE	REQ_CUTOVER_TIME	SGA_GB	DB_SIZE_GB	RAC_NODES_NUM	LOADED_CPU_COUNT
Finance	DALLAS	OKE_1	BRONZE	host31	db_2	QA	WE8ISO8859P9	NONPCI	>24 HOURS AND <= 48 HOURS	16	10000	1	8
Finance	DALLAS	OKE_1	BRONZE	host32	db_3	UAT	WE8ISO8859P9	NONPCI	>24 HOURS AND <= 48 HOURS	16	10000	1	8
Finance	DALLAS	OKE_1	BRONZE	host133	db_4	PRODUCTION	WE8ISO8859P9	NONPCI	>24 HOURS AND <= 48 HOURS	16	10000	1	8
Finance	DALLAS	OKE_1	BRONZE	host134	db_4	BCP	WE8ISO8859P9	NONPCI	>24 HOURS AND <= 48 HOURS	16	10000	1	8
Finance	DALLAS	OKE_10	BRONZE	host4	db_46	DEVELOPMENT	AL32UTF8	NONPCI	>24 HOURS AND <= 48 HOURS	2	10	1	8
Finance	DALLAS	OKE_10	BRONZE	host5	db_47	QA	AL32UTF8	NONPCI	>24 HOURS AND <= 48 HOURS	2	10	1	8
Finance	DALLAS	OKE_10	BRONZE	host6	db_48	UAT	AL32UTF8	NONPCI	>24 HOURS AND <= 48 HOURS	2	10	1	8
Finance	DALLAS	OKE_10	BRONZE	host107	db_49	PRODUCTION	AL32UTF8	NONPCI	>24 HOURS AND <= 48 HOURS	2	10	1	8
Finance	DALLAS	OKE_10	BRONZE	host108	db_49	BCP	AL32UTF8	NONPCI	>24 HOURS AND <= 48 HOURS	2	10	1	8
Sales	CHARLOTTE	OKE_100	BRONZE	host44	db_496	DEVELOPMENT	AL32UTF8	PCI	>24 HOURS AND <= 48 HOURS	8	1000	1	8
Sales	CHARLOTTE	OKE_100	BRONZE	host45	db_497	QA	AL32UTF8	PCI	>24 HOURS AND <= 48 HOURS	8	1000	1	8
Sales	CHARLOTTE	OKE_100	BRONZE	host46	db_498	UAT	AL32UTF8	PCI	>24 HOURS AND <= 48 HOURS	8	1000	1	8
Sales	CHARLOTTE	OKE_100	BRONZE	host147	db_499	PRODUCTION	AL32UTF8	PCI	>24 HOURS AND <= 48 HOURS	8	1000	1	8
Sales	CHARLOTTE	OKE_100	BRONZE	host148	db_499	BCP	AL32UTF8	PCI	>24 HOURS AND <= 48 HOURS	8	1000	1	8
Sales	DALLAS	OKE_101	BRONZE	host25	db_501	DEVELOPMENT	WE8ISO8859P9	PCI	>24 HOURS AND <= 48 HOURS	16	10000	1	8

Figure 6: Migrate 360 Database Load Sheet

The five Kubernetes application servers are added to the Migrate 360 application inventory load sheet, as shown in the following figure. Memory, CPU, and storage are obtained from standard OS tools.

APP_SERVICE_TIER	APP_TYPE	SUB_APP_NAME	ENVIRONMENT	APP_HOSTNAME	APP_HOST_PURPOSE	DBHOSTNAME	DB_NAME	APP_HOST_CPU_COUNT	APP_HOST_RAM_GB	APP_HOST_DISK_GB	REQ_CUTOVER_TIME	DATA_CENTER	SECURITY_ZONE
GOLD	GENERIC	OKE_APP08	DEVELOPMENT	appsrc8	APACHE_TOMCAT	platinumsource	srcdb15	2	8	1024	>48 HOURS	CHICAGO	PCI
GOLD	GENERIC	OKE_APP09	DEVELOPMENT	appsrc9	APACHE_TOMCAT	platinumsource	srcdb16	2	8	1024	>48 HOURS	CHICAGO	PCI
GOLD	GENERIC	OKE_APP10	DEVELOPMENT	appsrc10	APACHE_TOMCAT	platinumsource	srcdb17	2	8	1024	>48 HOURS	CHICAGO	PCI
GOLD	GENERIC	OKE_APP11	DEVELOPMENT	appsrc11	APACHE_TOMCAT	platinumsource	srcdb18	2	8	1024	>48 HOURS	CHICAGO	PCI
GOLD	GENERIC	OKE_APP12	DEVELOPMENT	appsrc12	APACHE_TOMCAT	platinumsource	srcdb19	2	8	1024	>48 HOURS	CHICAGO	PCI

Figure 7: Migrate 360 Application and VM Load Sheet

Estate Asset Classification and Assignments

After the data is loaded into Migrate 360, the next step is to run the load and classification process, which classifies the estate and assigns each asset to an appropriate size and shape, as follows:

- All databases are assigned a standard “t-shirt” size, which can be used if standardization is a requirement.
- Migration methods are assigned based on the business and technical details.
- All VMs are mapped to an OCI compute shape, such as VM.Standard2.1, for both the application servers and the database servers used for the VM DB systems.

The following figure shows a snapshot of the database classification and assignment.

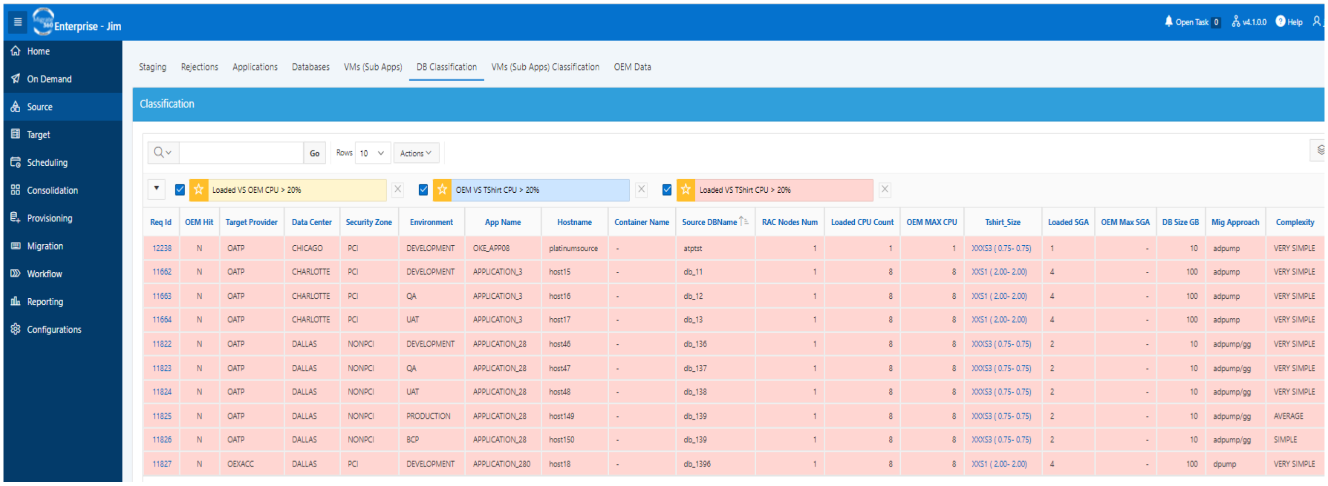


Figure 8: Migrate 360 Database Classification and Assignments

The following figure shows a snapshot of the application and VM classification and assignment.

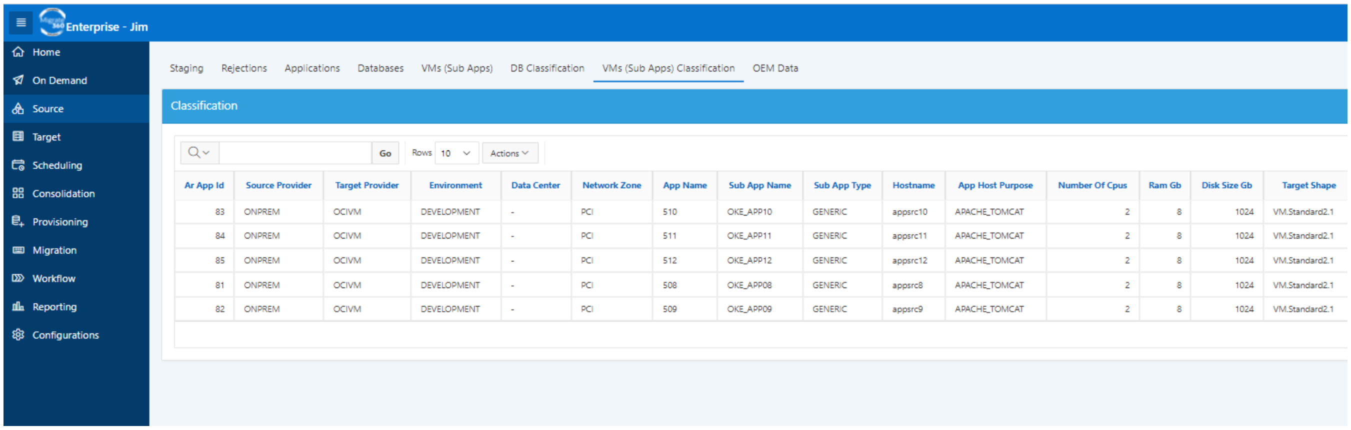


Figure 9: Migrate 360 Application and VM Classification and Assignments

Identifying Consolidation Objectives, Strategy, and Rules

The following consolidation objectives, strategy, and rules are used in this model:

- All assets are deployed into two private VCNs and compartments labeled `PCI` and `NONPCI`. All PCI assets are deployed in the `PCI` network, and all non-PCI assets are deployed the `NONPCI` network.
- The migration strategy is replatforming (lift-improve-and-shift), which calls for conversion to Oracle RAC and database upgrades from 11.2.04 to 19.2, along with deploying assets in the Oracle Autonomous Platform.
- All databases are deployed in the following service tier configurations:
    - Bronze: All development databases are deployed as single instance.
    - Silver: All QA and UAT systems are deployed in a 2-node Oracle RAC configuration.
    - Platinum: All production databases are deployed in a 2-node Oracle RAC configuration with Oracle Data Guard.

  No deployments have a Gold classification.

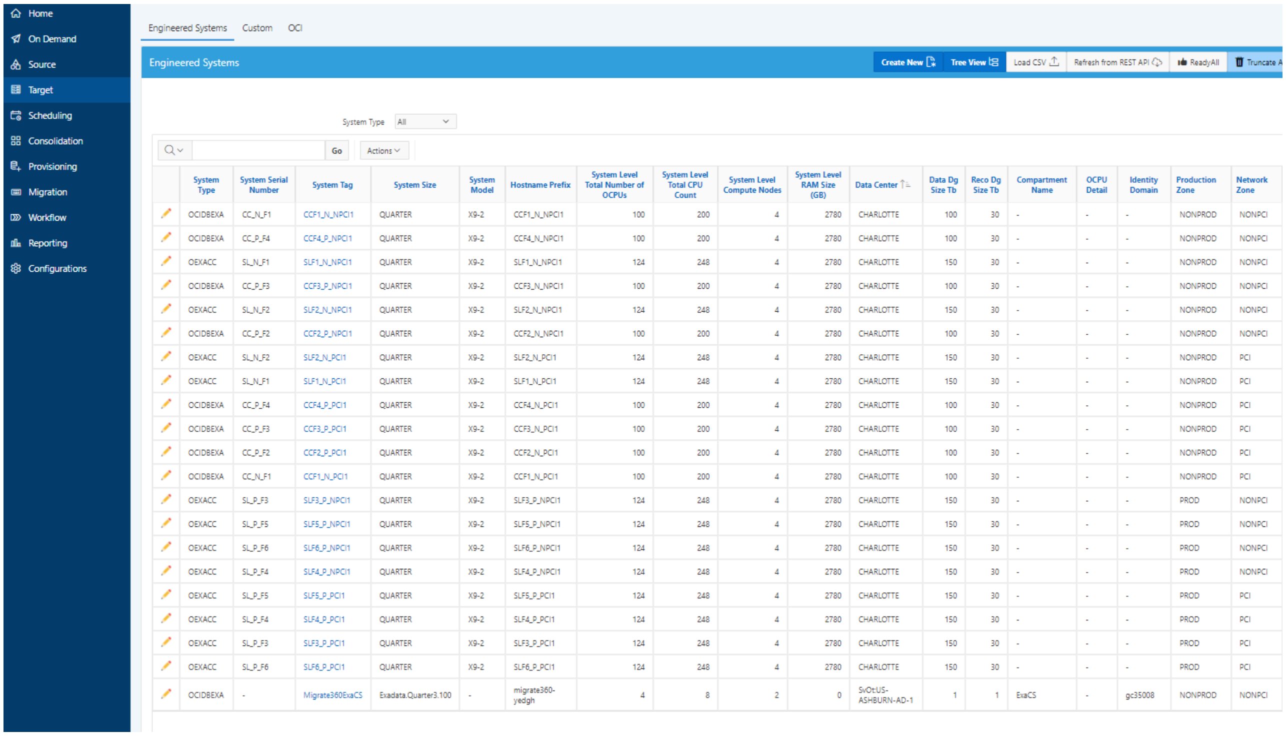
- Consolidation density for nonproduction systems is maximized, using aggressive multitenancy and consolidation rules, while maximizing production performance and availability:
    - All nonproduction databases are deployed in a multitenancy configuration and consolidated by LOB and lifecycle environments.
    - All nonproduction Exadata Cloud@Customer and Exadata Cloud Service systems are deployed with a 70% huge page allocation and a 5x CPU oversubscription to maximize consolidation density.
    - Production systems are deployed as single tenant (one PDB per CDB), and nonproduction systems are deployed as multitenant (multiple PDBs per CDB).
    - All production Exadata Cloud@Customer and Exadata Cloud Service systems are deployed with a 60% huge page allocation and a 3x CPU oversubscription to maximize performance and scalability.
- VM separation for non-PCI and PCI assets is provided according to the technical requirements.

Identifying the Initial Target Architecture

As identified in the estate discovery process, many technologies are being used to migrate the estate for this example customer and their assets. However, this distribution is not far from reality and represents an average OCI customer. Following is the initial target architecture for this example:

- Two VCNs are created, one to support a private PCI network and one to support a private non-PCI network for customer application and database access.
- Service quotas for 101 Autonomous Transaction Processing databases running on shared Exadata Infrastructure are established for the Finance LOB, with 47 deployed in the PCI network and 54 in the non-PCI network.
- Six Exadata Cloud@Customer X9M frames are targeted for 340 databases from the Sales LOB, with 185 databases deployed in a PCI VM cluster and network, and 155 databases deployed in a non-PCI VM cluster and network.
- Four Exadata Cloud Service X8M frames are targeted for 102 databases from the Customer Care LOB, with 127 databases deployed in a PCI VM cluster and network, and 75 databases deployed in a non-PCI VM cluster and network.
- Service quotas for 202 VM DB system databases are established for the Online Banking LOB, with 102 deployed in the PCI network and 100 in the non-PCI network.
- Service quotas for 151 Autonomous Data Warehouse databases running on shared Exadata Infrastructure are established for the Marketing LOB, with 186 deployed in the PCI network and 65 in the non-PCI network.
- Service quotas to support a maximum of 300 OCI Container Engine for Kubernetes compute servers are established to support the customer Container Engine for Kubernetes application in the PCI network. Additional service quotas are required to support 100 web and application servers in the non-PCI network.

The identified Exadata Cloud@Customer and Exadata Cloud Service frames are then loaded into Migrate 360 and displayed in the target section, as shown in the following figure.



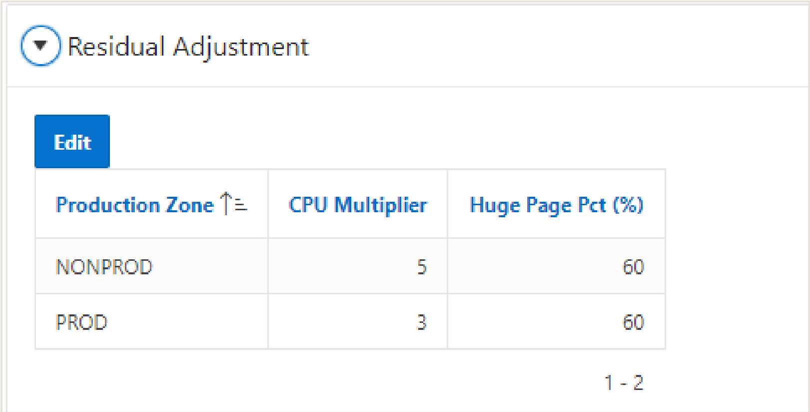
System Type	System Serial Number	System Tag	System Size	System Model	Hostname Prefix	System Level Total Number of OCPUs	System Level Total CPU Count	System Level Compute Nodes	System Level RAM Size (GB)	Data Center	Data Dg Size Tb	Reco Dg Size Tb	Compartment Name	OCPU Detail	Identity Domain	Production Zone	Network Zone
OCIDBEXA	CC_N_F1	CCF1_N_NPCI1	QUARTER	X9-2	CCF1_N_NPCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	NONPCI
OCIDBEXA	CC_P_F4	CCF4_P_NPCI1	QUARTER	X9-2	CCF4_N_NPCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	NONPCI
OEXACC	SL_N_F1	SLF1_N_NPCI1	QUARTER	X9-2	SLF1_N_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	NONPROD	NONPCI
OCIDBEXA	CC_P_F3	CCF3_P_NPCI1	QUARTER	X9-2	CCF3_N_NPCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	NONPCI
OEXACC	SL_N_F2	SLF2_N_NPCI1	QUARTER	X9-2	SLF2_N_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	NONPROD	NONPCI
OCIDBEXA	CC_P_F2	CCF2_P_NPCI1	QUARTER	X9-2	CCF2_N_NPCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	NONPCI
OEXACC	SL_N_F2	SLF2_N_PCI1	QUARTER	X9-2	SLF2_N_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	NONPROD	PCI
OEXACC	SL_N_F1	SLF1_N_PCI1	QUARTER	X9-2	SLF1_N_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	NONPROD	PCI
OCIDBEXA	CC_P_F4	CCF4_P_PCI1	QUARTER	X9-2	CCF4_N_PCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	PCI
OCIDBEXA	CC_P_F3	CCF3_P_PCI1	QUARTER	X9-2	CCF3_N_PCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	PCI
OCIDBEXA	CC_P_F2	CCF2_P_PCI1	QUARTER	X9-2	CCF2_N_PCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	PCI
OCIDBEXA	CC_N_F1	CCF1_N_PCI1	QUARTER	X9-2	CCF1_N_PCI1	100	200	4	2780	CHARLOTTE	100	30	-	-	-	NONPROD	PCI
OEXACC	SL_P_F3	SLF3_P_NPCI1	QUARTER	X9-2	SLF3_P_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	NONPCI
OEXACC	SL_P_F5	SLF5_P_NPCI1	QUARTER	X9-2	SLF5_P_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	NONPCI
OEXACC	SL_P_F6	SLF6_P_NPCI1	QUARTER	X9-2	SLF6_P_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	NONPCI
OEXACC	SL_P_F4	SLF4_P_NPCI1	QUARTER	X9-2	SLF4_P_NPCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	NONPCI
OEXACC	SL_P_F5	SLF5_P_PCI1	QUARTER	X9-2	SLF5_P_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	PCI
OEXACC	SL_P_F4	SLF4_P_PCI1	QUARTER	X9-2	SLF4_P_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	PCI
OEXACC	SL_P_F3	SLF3_P_PCI1	QUARTER	X9-2	SLF3_P_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	PCI
OEXACC	SL_P_F6	SLF6_P_PCI1	QUARTER	X9-2	SLF6_P_PCI1	124	248	4	2780	CHARLOTTE	150	30	-	-	-	PROD	PCI
OCIDBEXA	-	Migrate360ExaCS	Exadata.Quarter3.100	-	migrate360-yedgh	4	8	2	0	SvOt:US-ASHBURN-AD-1	1	1	ExaCS	-	gc35008	NONPROD	NONPCI

Figure 10: Migrate 360 Target Exadata Frames for Modeling Effort

Identifying Target Resource Limits

The next step is to define the consolidation rules related to residual adjustments for this example. This section outlines how the resource limits discussed in this paper are enabled and set.

First, the residual adjustments are set to provide for the necessary huge page allocation and CPU oversubscription setting, as shown in the following screenshot.



Production Zone	CPU Multiplier	Huge Page Pct (%)
NONPROD	5	60
PROD	3	60

Figure 11: Migrate 360 Target Oversubscription and Huge Page Limits for Modeling Scenario

Next, the consolidation rules related to storage are defined. The following screenshot shows how the storage limits are defined. The maximum data disk group limit (DATA_DG_LIMIT_PCT) is set to 60% of all data disk group storage.

Configuration Parameters

Edit

Parammeter Name	Parameter Value
CUSTDB_DATA_DG_TB_LOCAL_BKUP	60
CUSTDB_DATA_DG_TB_OUTSIDE_BKUP	80
CUSTDB_HOST_LOCAL_STORAGE_GB	100
CUSTDB_NEW_QUOTE_BACKUP_LOC	LOCAL
CUSTDB_USABLE_STORAGE_TB	100
DATA_DG_LIMIT_PCT	60
DEFAULT_CONSOLIDATION_STRATEGY	LIFTSHIFT
DEFAULT_PDB_TO_CDB_TSHIRT_SIZE	XXXL4
DEFAULT_TARGET_PLATFORM	ODBCS
TARGET_ARCHIVELOG_RETENTION	3
TARGET_DB_COMPRESS_RATIO	1.0

Figure 12: Migrate 360 Target Configuration Limits for Modeling Scenario Highlighting Data Disk Group Used Percentage

Next, the consolidation rules related to multitenancy consolidation, and whether to allow for single tenancy or multitenancy based on environment, are defined. This represents the ability to enable these configurations at a global level.

Using the following example, all PROD systems are forced to single tenancy (one PDB per CDB) and all non-PROD systems default to multitenancy.

CDB Consolidation Single Tenant Rules

Edit

Environment	Application ID	Is Single Tenant	Is Non-CDB
BCP	NA	Y	N
PROD	NA	Y	N

Figure 13: Migrate 360 Single Tenancy Versus Multitenancy Rules Based on Environment

For systems that are not single tenancy, the following rules are used to map them to an appropriate target cohort, or CDB. All parameters marked with a YES are used from the loading process for a multitenancy consolidation strategy.

Note: In the following example, the customer requested a multitenancy strategy based on Data Center, Environment, Line of Business, Business Group, Security Zone, Service Tier, and Time Zone.

CNDB Mapping Columns

Edit

CNDB Column Name	Used For Map
BUSINESS_GROUP	YES
CUSTOM_ATTR1	NO
CUSTOM_ATTR2	NO
CUSTOM_ATTR3	NO
CUSTOM_ATTR4	NO
CUSTOM_ATTR5	NO
CUSTOM_ATTR6	NO
CUSTOM_ATTR7	NO
CUSTOM_ATTR8	NO
CUSTOM_ATTR9	NO
DATA_CENTER	YES
ENVIRONMENT	YES
LINE_OF_BUSINESS	YES
NLS_CHARACTERSET	NO
SECURITY_ZONE	YES
SERVICE_TIER	YES
TIME_ZONE	YES

Figure 14: Migrate 360 Multitenancy Rules (PDBs per CDB)

Next, additional resources need to be allowed to be added to the target database container based on the number of PDBs. This provides more resources as more databases are added to the container, based on need.

CDB Consolidation SGA Adjustment

Edit

No Pdbs From	No Pdbs To	Sga Increment By	Cpu Increment By
0	4	0	0
5	10	2	0
11	20	4	0

Figure 15: Migrate 360 Multitenancy Resource Increases Based on Number of PDBs

Finally, the Oracle RAC conversion settings are applied. As noted in the technical details section of this case study, all source databases are single instance and are converted to Oracle RAC and distributed across two nodes in an active/active configuration. The following conversion rule converts all single instance databases to 2-node Oracle RAC for the modeling scenario.

General RAC Convert Rule

Go Actions Edit

Environment	Db Service Tier	Source RAC Node From	Source RAC Node To	Target RAC Node	T-shirt Change By	Increment Decrement By
DEVELOPMENT	SI	1	1	1	0	DECREASE
DEVELOPMENT	EXA-RAC	1	2	2	0	INCREASE

Figure 16: Migrate 360 Oracle RAC Conversion

Modeling the Scenario (Capacity Analysis)

After the necessary parameters and limits are set, the capacity modeling can begin. Migrate 360 provides the ability to model the existing estate, target architecture, and future needs, as outlined in the following reports:

- **Project Impact Analysis**: This report outlines the current estate and the target architecture that has been loaded into the Migrate 360 application.
- **Target Resource Projection**: If the Project Impact Analysis report identifies a capacity shortfall, the Target Resource Projection report identifies the amount of hardware required to map the entire estate or the estate based on a cutoff date.

Both reports can be run from the Capacity Planning dashboard.

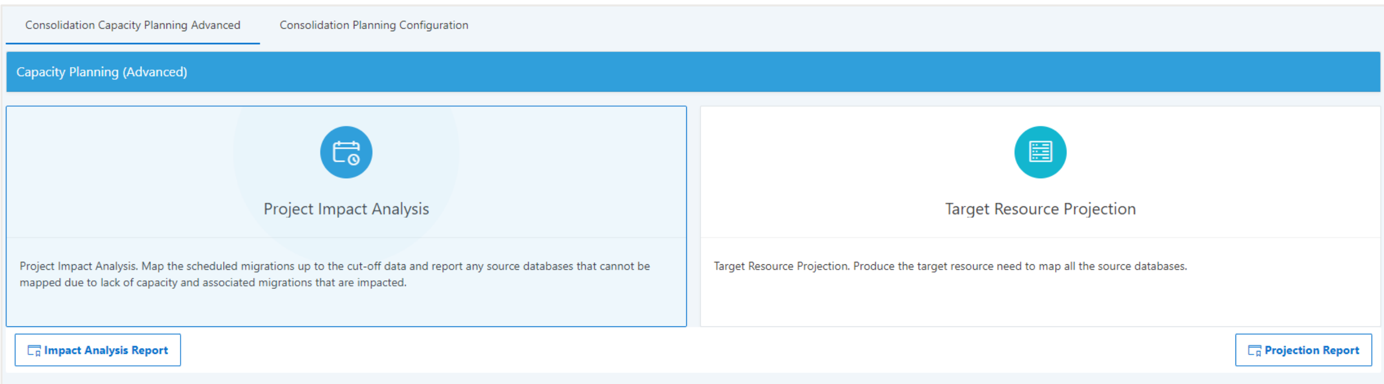


Figure 17: Migrate 360 Capacity Planning Dashboard

Project Impact Analysis Report

This report consolidates the source databases to the target architecture based on the default configuration, such as environment, network, and data centers, to validate all source and target capacity resources based on the migration cutoff dates. The report shows resources that are unmapped either because of a lack of capacity or because of improper configuration.

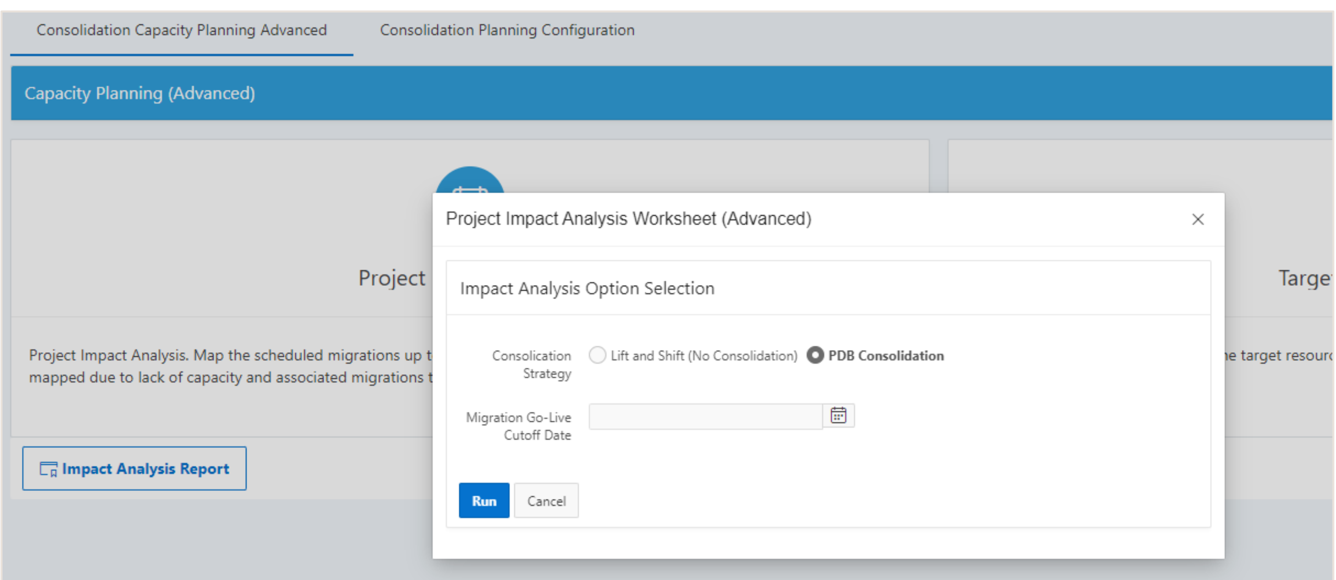


Figure 18: Migrate 360 Project Impact Analysis

Target Resource Projection Report

This report projects the capacity required on the target architecture based on the default configuration, such as environment, network, and data centers, to validate all source and target capacity resources based on the migration cutoff dates. The report shows resources that are unmapped either because of a lack of capacity or because of improper configurations, as well as recommend additional hardware.

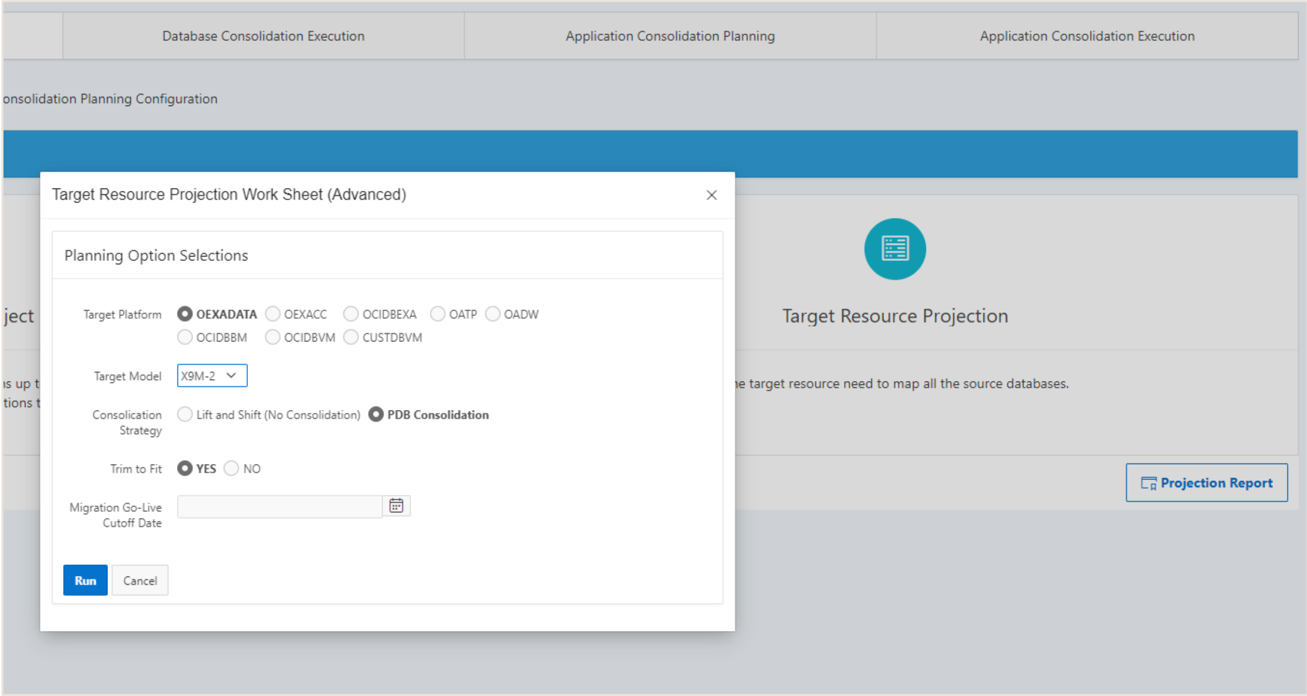
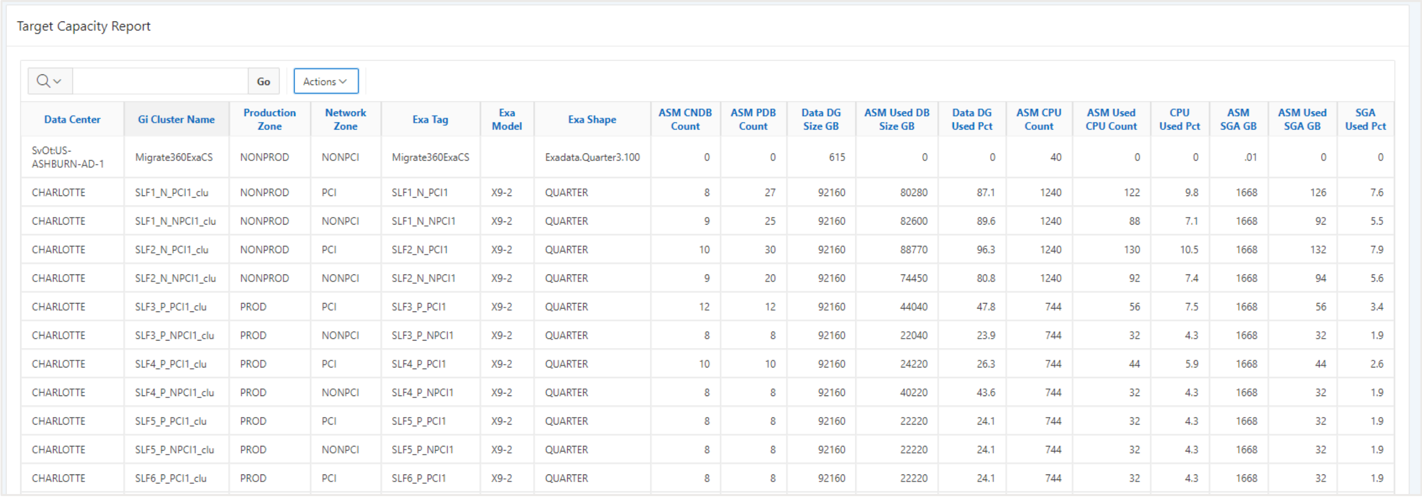


Figure 19: Migrate 360 Target Resource Projection

Target Capacity Report with Unmapped and Mapped Migration

Both the Project Impact Analysis and Target Resource Projection reports provide target capacity with mapped and unmapped migration, as shown in the following figures.



Target Capacity Report

Data Center	Gi Cluster Name	Production Zone	Network Zone	Exa Tag	Exa Model	Exa Shape	ASM CNDB Count	ASM PDB Count	Data DG Size GB	ASM Used DB Size GB	Data DG Used Pct	ASM CPU Count	ASM Used CPU Count	CPU Used Pct	ASM SGA GB	ASM Used SGA GB	SGA Used Pct
SvOt:US-ASHBURN-AD-1	Migrate360ExaCS	NONPROD	NONPCI	Migrate360ExaCS		Exadata.Quarter3.100	0	0	615	0	0	40	0	0	.01	0	0
CHARLOTTE	SLF1_N_PCI1_clu	NONPROD	PCI	SLF1_N_PCI1	X9-2	QUARTER	8	27	92160	80280	87.1	1240	122	9.8	1668	126	7.6
CHARLOTTE	SLF1_N_NPCI1_clu	NONPROD	NONPCI	SLF1_N_NPCI1	X9-2	QUARTER	9	25	92160	82600	89.6	1240	88	7.1	1668	92	5.5
CHARLOTTE	SLF2_N_PCI1_clu	NONPROD	PCI	SLF2_N_PCI1	X9-2	QUARTER	10	30	92160	88770	96.3	1240	130	10.5	1668	132	7.9
CHARLOTTE	SLF2_N_NPCI1_clu	NONPROD	NONPCI	SLF2_N_NPCI1	X9-2	QUARTER	9	20	92160	74450	80.8	1240	92	7.4	1668	94	5.6
CHARLOTTE	SLF3_P_PCI1_clu	PROD	PCI	SLF3_P_PCI1	X9-2	QUARTER	12	12	92160	44040	47.8	744	56	7.5	1668	56	3.4
CHARLOTTE	SLF3_P_NPCI1_clu	PROD	NONPCI	SLF3_P_NPCI1	X9-2	QUARTER	8	8	92160	22040	23.9	744	32	4.3	1668	32	1.9
CHARLOTTE	SLF4_P_PCI1_clu	PROD	PCI	SLF4_P_PCI1	X9-2	QUARTER	10	10	92160	24220	26.3	744	44	5.9	1668	44	2.6
CHARLOTTE	SLF4_P_NPCI1_clu	PROD	NONPCI	SLF4_P_NPCI1	X9-2	QUARTER	8	8	92160	40220	43.6	744	32	4.3	1668	32	1.9
CHARLOTTE	SLF5_P_PCI1_clu	PROD	PCI	SLF5_P_PCI1	X9-2	QUARTER	8	8	92160	22220	24.1	744	32	4.3	1668	32	1.9
CHARLOTTE	SLF5_P_NPCI1_clu	PROD	NONPCI	SLF5_P_NPCI1	X9-2	QUARTER	8	8	92160	22220	24.1	744	32	4.3	1668	32	1.9
CHARLOTTE	SLF6_P_PCI1_clu	PROD	PCI	SLF6_P_PCI1	X9-2	QUARTER	8	8	92160	22220	24.1	744	32	4.3	1668	32	1.9

Figure 20: Migrate 360 Target Capacity Report

Unmapped Migrations

Go Live Date	PDB Application Id	PDB Name	MIG Method	Complexity	Req Cutover Time	PDB Data Center	PDB Environment	PDB Security Zone	PDB Service Tier	PDB RAC Nodes	PDB Tshirt Size	PDB Archivelog Daily GB	PDB NLS Characterset	Request Id	Source Provider	Target Provider	Aplcndb Id	CNDB Tshirt Size	CPU Count	SGA GB	Apl Exception
4/30/2022	284	pdb00475	rman dup	SIMPLE	>0 MINS AND <= 30 MINS	DALLAS	BCP	NONPCI	PLATINUM	1	XXXS3	0	AL32UTF8	11851	ONPREM	OEXACC	3856	XXS1	2	2	--TARGET NOT FOUND -- -PARAMETERS-- | DATA_CENTER:DALLAS | NETWORK_ZONE:NONPC | PRODUCTION_ZONE:PRO | POOL_ATTR1:NA | POOL_ATTR2:NA
4/30/2022	286	pdb00484	dpump	AVERAGE	>30 MINS AND <= 60 MINS	DALLAS	PRODUCTION	NONPCI	SILVER	1	XXXS3	0	AL32UTF8	11860	ONPREM	OEXACC	3859	XXS1	2	2	--TARGET NOT FOUND -- -PARAMETERS-- | DATA_CENTER:DALLAS | NETWORK_ZONE:NONPC | PRODUCTION_ZONE:PRO | POOL_ATTR1:NA | POOL_ATTR2:NA
4/30/2022	286	pdb00485	rman dup	SIMPLE	>30 MINS AND <= 60 MINS	DALLAS	BCP	NONPCI	SILVER	1	XXXS3	0	AL32UTF8	11861	ONPREM	OEXACC	3860	XXS1	2	2	--TARGET NOT FOUND -- -PARAMETERS-- | DATA_CENTER:DALLAS | NETWORK_ZONE:NONPC | PRODUCTION_ZONE:PRO | POOL_ATTR1:NA | POOL_ATTR2:NA

Figure 21: Migrate 360 Target Capacity Report, Unmapped Migrations

Mapped Migrations

Data Center	Production Zone	Network Zone	APLPDB Id	Mig Method	PDB Name	PDB RAC Nodes	PDB Tshirt Size	PDB CPU Count	PDB SGA GB	PDB DB Size GB	PDB Archivelog Daily GB	PDB Data Center	PDB Environment	PDB Security Zone	PDB Service Tier	PDB NLS Characterset	PDB Application Id	APLCNDB Id	CDB Name	CDB RAC Nodes	CDB Tshirt Size
CHARLOTTE	PROD	PCI	4917	rman dup	pdb00546	1	XXS1	2	2	100	0	CHARLOTTE	BCP	PCI	BRONZE	AL32UTF8	184	3792	CDB00270	1	XXS1
CHARLOTTE	PROD	PCI	5357	rman dup	pdb00986	1	XXS3	6	6	1000	0	CHARLOTTE	BCP	PCI	BRONZE	AL32UTF8	120	3599	CDB00077	1	XXS3
CHARLOTTE	PROD	PCI	4890	dpump/gg	pdb00519	1	XXS3	6	6	1000	0	CHARLOTTE	PRODUCTION	PCI	PLATINUM	AL32UTF8	292	3890	CDB00368	1	XXS3
CHARLOTTE	NONPROD	PCI	5245	dpump	pdb00874	1	XXS3	6	6	1000	0	CHARLOTTE	QA	PCI	BRONZE	AL32UTF8	100	3540	CDB00018	1	S2
CHARLOTTE	NONPROD	PCI	5290	dpump/gg	pdb00919	1	XXS3	6	6	10000	0	CHARLOTTE	QA	PCI	BRONZE	WE8ISO8859P9	109	3540	CDB00018	1	S2
CHARLOTTE	NONPROD	PCI	5300	dpump	pdb00929	1	XXXS3	.75	.75	10	0	CHARLOTTE	QA	PCI	BRONZE	AL32UTF8	110	3540	CDB00018	1	S2
CHARLOTTE	NONPROD	PCI	5320	dpump	pdb00949	1	XXXS3	.75	.75	10	0	CHARLOTTE	QA	PCI	BRONZE	AL32UTF8	114	3540	CDB00018	1	S2
CHARLOTTE	NONPROD	PCI	5325	dpump	pdb00954	1	XXS3	6	6	10000	0	CHARLOTTE	QA	PCI	BRONZE	WE8ISO8859P9	115	3540	CDB00018	1	S2
CHARLOTTE	NONPROD	PCI	4884	dpump	pdb00513	1	XXS3	6	6	1000	0	CHARLOTTE	UAT	PCI	SILVER	WE8ISO8859P9	291	3830	CDB00308	1	XXS4
CHARLOTTE	NONPROD	PCI	4909	dpump	pdb00538	1	XXXS3	.75	.75	10	0	CHARLOTTE	UAT	PCI	SILVER	AL32UTF8	296	3830	CDB00308	1	XXS4
CHARLOTTE	NONPROD	PCI	4839	dpump	pdb00468	1	XXXS3	.75	.75	10	0	CHARLOTTE	UAT	PCI	SILVER	WE8ISO8859P9	283	3830	CDB00308	1	XXS4
CHARLOTTE	PROD	NONPCI	5278	rman dup	pdb00907	1	XXXS3	.75	.75	10	0	CHARLOTTE	BCP	NONPCI	BRONZE	AL32UTF8	7106	3591	CDB00069	1	XXS1
CHARLOTTE	PROD	NONPCI	5084	rman dup	pdb00713	1	XXS3	6	6	1000	0	CHARLOTTE	BCP	NONPCI	BRONZE	WE8ISO8859P9	125	3610	CDB00088	1	XXS3
CHARLOTTE	NONPROD	NONPCI	5276	dpump	pdb00905	1	XXXS3	.75	.75	10	0	CHARLOTTE	UAT	NONPCI	BRONZE	AL32UTF8	7106	3552	CDB00030	1	XS3
CHARLOTTE	NONPROD	NONPCI	5331	dpump	pdb00960	1	XXS3	6	6	10000	0	CHARLOTTE	UAT	NONPCI	BRONZE	AL32UTF8	116	3552	CDB00030	1	XS3
CHARLOTTE	NONPROD	NONPCI	5341	dpump/gg	pdb00970	1	XXS3	6	6	10000	0	CHARLOTTE	UAT	NONPCI	BRONZE	AL32UTF8	118	3552	CDB00030	1	XS3

Figure 22: Migrate 360 Target Capacity Report, Mapped Migrations

The Target Resource Projection report also outlines the amount of hardware required during a shortfall.

Consolidation \ Capacity Planning \

Target Capacity Projection Report (Advanced)

Version Id DB-Version-145

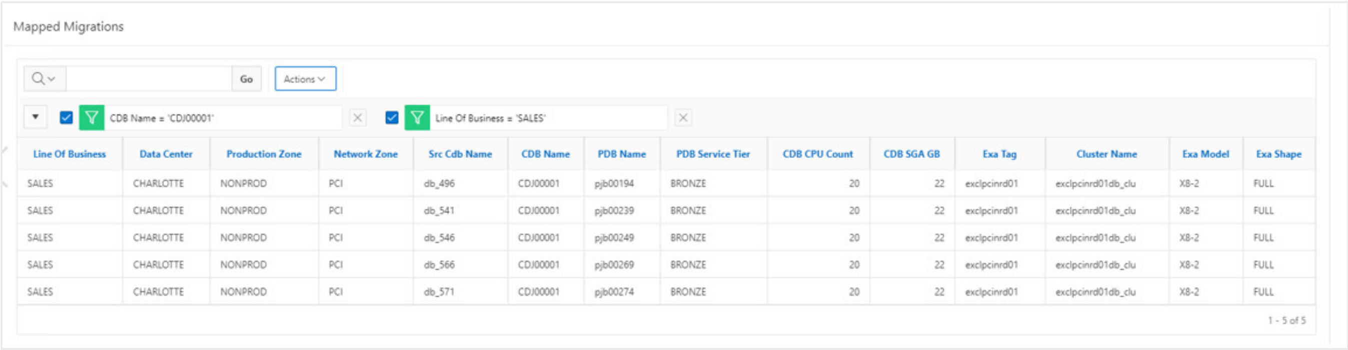
New Targets Needed

System Type	System Serial Number	System Model	System Tag	System Size	DB VMs	DB VM CPU Cores	DB VM RAM Size GB	Ops Ready	Ready Date	Hostname Prefix	Data Center	Production Zone	Network Zone	Pool Attr1	Pool Attr2	Pool Attr3	Pool Attr4	Pool Attr5	Cpu detail	Identity domain
OEXACC	NEWXCC_1811	X8-2	NEWXCC_1811	FULL	8	11	5760	N	-	NEWXCC_1811	CHARLOTTE	NONPROD	NONPCI	NA	NA	NA	NA	NA	-	-
OEXACC	NEWXCC_1813	X8-2	NEWXCC_1813	FULL	8	11	5760	N	-	NEWXCC_1813	DALLAS	NONPROD	PCI	NA	NA	NA	NA	NA	-	-

1 - 2 of 2

Figure 23: Migrate 360 Target Capacity Report, New Targets Needed

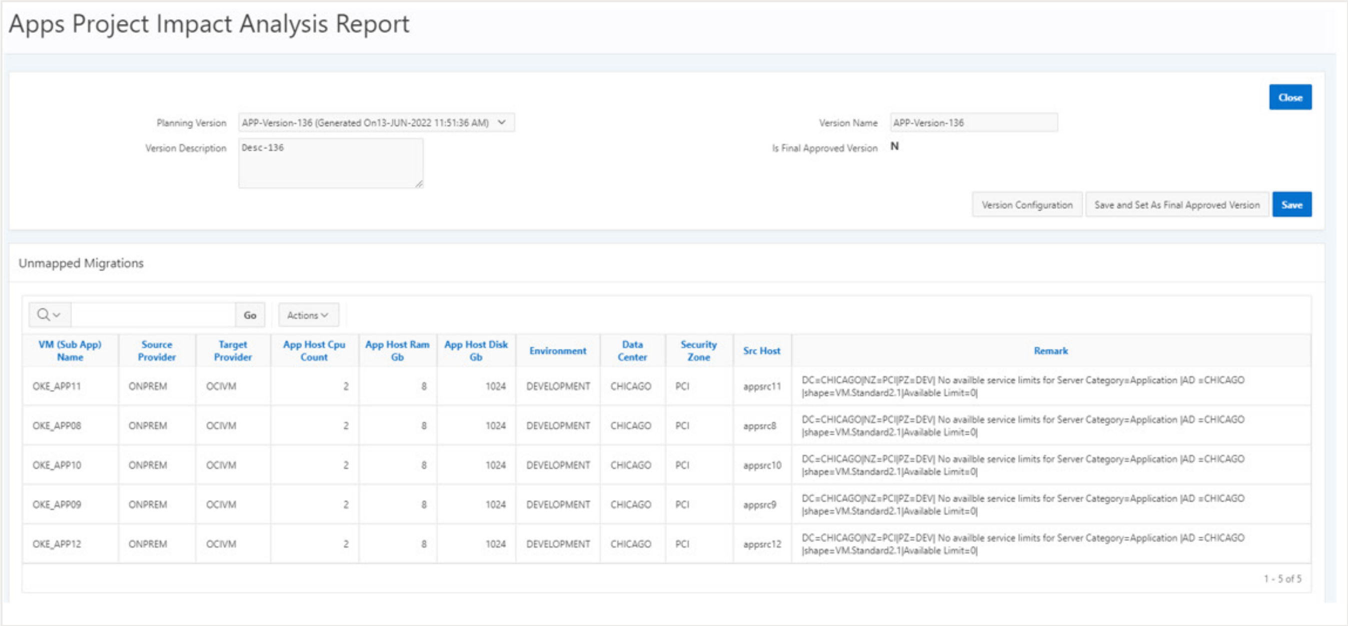
The Target Capacity report also validates the multitenancy rules as specified in the Consolidation Configuration section of Migrate360.



Line Of Business	Data Center	Production Zone	Network Zone	Src Cdb Name	CDB Name	PDB Name	PDB Service Tier	CDB CPU Count	CDB SGA GB	Exa Tag	Cluster Name	Exa Model	Exa Shape
SALES	CHARLOTTE	NONPROD	PCI	db_496	CDJ00001	pjb00194	BRONZE	20	22	exclpcinrd01	exclpcinrd01db_clu	X8-2	FULL
SALES	CHARLOTTE	NONPROD	PCI	db_541	CDJ00001	pjb00239	BRONZE	20	22	exclpcinrd01	exclpcinrd01db_clu	X8-2	FULL
SALES	CHARLOTTE	NONPROD	PCI	db_546	CDJ00001	pjb00249	BRONZE	20	22	exclpcinrd01	exclpcinrd01db_clu	X8-2	FULL
SALES	CHARLOTTE	NONPROD	PCI	db_566	CDJ00001	pjb00269	BRONZE	20	22	exclpcinrd01	exclpcinrd01db_clu	X8-2	FULL
SALES	CHARLOTTE	NONPROD	PCI	db_571	CDJ00001	pjb00274	BRONZE	20	22	exclpcinrd01	exclpcinrd01db_clu	X8-2	FULL

Figure 24: Migrate 360 Target Capacity Report, Multitenancy Strategy Validation

As with validating capacity with databases and servers, the Application Impact Analysis Report can validate the quotas and limits within the tenancy for the migration.



VM (Sub App) Name	Source Provider	Target Provider	App Host Cpu Count	App Host Ram Gb	App Host Disk Gb	Environment	Data Center	Security Zone	Src Host	Remark
OKE_APP11	ONPREM	OCIVM	2	8	1024	DEVELOPMENT	CHICAGO	PCI	appsrc11	DC=CHICAGO|NZ=PCI|PZ=DEV| No availble service limits for Server Category=Application |AD =CHICAGO |shape=VM.Standard2.1|Available Limit=0|
OKE_APP08	ONPREM	OCIVM	2	8	1024	DEVELOPMENT	CHICAGO	PCI	appsrc8	DC=CHICAGO|NZ=PCI|PZ=DEV| No availble service limits for Server Category=Application |AD =CHICAGO |shape=VM.Standard2.1|Available Limit=0|
OKE_APP10	ONPREM	OCIVM	2	8	1024	DEVELOPMENT	CHICAGO	PCI	appsrc10	DC=CHICAGO|NZ=PCI|PZ=DEV| No availble service limits for Server Category=Application |AD =CHICAGO |shape=VM.Standard2.1|Available Limit=0|
OKE_APP09	ONPREM	OCIVM	2	8	1024	DEVELOPMENT	CHICAGO	PCI	appsrc9	DC=CHICAGO|NZ=PCI|PZ=DEV| No availble service limits for Server Category=Application |AD =CHICAGO |shape=VM.Standard2.1|Available Limit=0|
OKE_APP12	ONPREM	OCIVM	2	8	1024	DEVELOPMENT	CHICAGO	PCI	appsrc12	DC=CHICAGO|NZ=PCI|PZ=DEV| No availble service limits for Server Category=Application |AD =CHICAGO |shape=VM.Standard2.1|Available Limit=0|

Figure 25: Migrate 360 Target Capacity Report, VM Consolidation and Quota Validation

Appendix A: Exadata Cloud@Customer X9M Technical Data Sheet

The following image shows hardware configuration values from Table 1 of the [Oracle Exadata Cloud@Customer X9M data sheet](#).

Exadata Cloud@Customer Typical Hardware Configurations

Service item	Base System	Quarter Rack	Half Rack	Full Rack
Number of Database Servers	2	2	4	8
Maximum Number of OCPUs	48	100	200	400
Total Memory Available for Guest VMs (GB)	656	2,780	5,560	11,120
Max # of VM Clusters per system	4	8	8	8
Min # of OCPUs per VM Cluster	4	4	8	16
Max Usable Local Storage Per DB Server (GB)	900 (with single VM)	2,020 (across 3 VMs)	2,020 (across 3 VMs)	2,020 (across 3 VMs)
Max Usable Local Storage Per VM (GB)	900	900	900	900
Number of Storage Servers	3	3	6	12
Total Cores in Storage Servers	144	144	288	576
Total Persistent Memory Capacity[6] (TB)	0	4.5	9.0	18.0
Total Flash Capacity (TB)	38.4	76.8	153.6	307.2
Total Usable Disk Capacity[1] (TB)	74	149	299	598
Max DB Size – No Local Backup[1] (TB)	59	119	239	479
Max DB Size – Local Backup[1] (TB)	29	59	119	239
Max SQL Flash Bandwidth[2] (GB/s)	25	75	150	300
Max SQL PMem/Flash Read IOPS[3,6]	562,500	3,000,000	6,000,000	12,000,000
Max SQL PMem/Flash Write IOPS[4,6]	518,000	1,410,000	2,820,000	5,640,000
Max SQL Disk Bandwidth[2] (GB/s)	2.7	5.4	10.8	21.5
Max SQL Disk IOPS[3]	3,900	7,800	15,600	31,000
Max Data Load Rate[5] (TB/hr)	3.8	7.5	15.0	30.0
Network Connectivity	Per Database Server: • 2x 10/25 Gb Ethernet (backup) • 2x 10/25 Gb Ethernet (client) Per Control Plane Server: • 2x 10/25 Gb Ethernet (minimum internet connectivity of 50Mbs down and 10Mbps up required)			

Appendix B: Exadata Cloud Service X8M Technical Data Sheet

The following image shows hardware configuration values from Table 1 of the Oracle Exadata Cloud Service X8M data sheet.

Exadata Cloud Service Typical Hardware Configurations

Service item	Base System[1]	X8M Quarter Rack[2]	X8M Half Rack[2]	X8M Full Rack[2]
Number of Database Servers	2	2	4	8
Maximum Number of OCPUs	48	100	200	400
Total Memory Available (GB)	720	2,780	5,560	11,120
Min # of Enabled OCPUs	4	4	8	16
Usable Local Storage Per DB Server (GB)	200	1,163	1,163	1,163
Number of Storage Servers	3	3	6	12
Total Cores in Storage Servers	144	144	288	576
Total Persistent Memory Capacity (TB)	0	4.5	9.0	18.0
Total Flash Capacity (TB)	38.4	76.8	153.6	307.2
Total Usable Disk Capacity[3] (TB)	74	149	299	598
Max DB Size – No Local Backup[3] (TB)	59	119	239	479
Max DB Size – Local Backup[3] (TB)	29	59	119	239
Max SQL Flash Bandwidth[4] (GB/s)	25	75	150	300
Max SQL PMEM/Flash Read IOPS[5]	562,500	3,000,000	6,000,000	12,000,000
Max SQL PMEM/Flash Write IOPS[6]	518,000	1,410,000	2,820,000	5,640,000
Max SQL Disk Bandwidth[4] (GB/s)	2.7	5.4	10.8	21.5
Max SQL Disk IOPS[5]	3,900	7,800	15,600	31,000
Max Data Load Rate[7] (TB/hr)	3.8	7.5	15.0	30.0
Network (Client/Backup)	10 GbE	25 GbE	25 GbE	25 GbE

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle