Deploying a Private Database Cloud using Enterprise Manager 12c Cloud Control Cookbook
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Overview</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Private Database Cloud Enterprise Manager 12c Capabilities Overview</strong></td>
<td>3</td>
</tr>
<tr>
<td>Dedicated Database</td>
<td>4</td>
</tr>
<tr>
<td>Database 12c Multitenant</td>
<td>4</td>
</tr>
<tr>
<td>Schemas</td>
<td>4</td>
</tr>
<tr>
<td>Clone Including Snap Clones</td>
<td>4</td>
</tr>
<tr>
<td><strong>Private Database Cloud Architecture Design</strong></td>
<td>6</td>
</tr>
<tr>
<td>DBaaS Cloud Definitions</td>
<td>6</td>
</tr>
<tr>
<td>Defining Database Service Tiers and Catalogs</td>
<td>7</td>
</tr>
<tr>
<td>Technical versus Self-Service Catalog</td>
<td>9</td>
</tr>
<tr>
<td>Bronze Tier Example</td>
<td>9</td>
</tr>
<tr>
<td>Bronze Reference Architecture</td>
<td>9</td>
</tr>
<tr>
<td>Silver Tier Example</td>
<td>10</td>
</tr>
<tr>
<td>Silver Reference Architecture</td>
<td>10</td>
</tr>
<tr>
<td>Gold Tier Example</td>
<td>10</td>
</tr>
<tr>
<td>Gold Reference Architecture</td>
<td>10</td>
</tr>
<tr>
<td>Platinum Tier Example</td>
<td>11</td>
</tr>
<tr>
<td>Platinum Reference Architecture</td>
<td>11</td>
</tr>
<tr>
<td>Service Tier Recovery Point and Time Objectives</td>
<td>11</td>
</tr>
<tr>
<td><strong>Private Database Cloud Consolidation Guidelines</strong></td>
<td>13</td>
</tr>
<tr>
<td>- Cloud Consolidation Model</td>
<td>13</td>
</tr>
<tr>
<td>- Cloud Consolidation Planning</td>
<td>13</td>
</tr>
<tr>
<td>- Cloud Consolidation Testing</td>
<td>14</td>
</tr>
<tr>
<td>Configuration Pollution and Sprawl</td>
<td>15</td>
</tr>
<tr>
<td><strong>Implementation of a DBaaS using Cloud Management Pack</strong></td>
<td>16</td>
</tr>
<tr>
<td>Step 1: Configure Privilege Delegation for Database Servers</td>
<td>16</td>
</tr>
<tr>
<td>Step 2: Defined Named Credentials</td>
<td>17</td>
</tr>
<tr>
<td>Step 3: Configure Software Library</td>
<td>18</td>
</tr>
<tr>
<td>Step 4: Creating Cloud Users and Roles</td>
<td>19</td>
</tr>
</tbody>
</table>
Step 5: Create Self Service Portal Users......................................................... 20
Step 6: Defining Databases Zones ................................................................. 21
Step 6: Defining Database Pools................................................................. 24
Step 8: Quotas .......................................................................................... 28
Step 9: Defining Service Templates for Self Service Provisioning ......... 28
Step 10: Using Self Service Portal ............................................................... 37
Managing Your Private Database Cloud.................................................. 40
  Defining Monitoring Architecture.......................................................... 40
  Administration Groups ........................................................................... 40
  Incident Management and Incident Rules ............................................ 41
  Compliance Framework ........................................................................ 42
  Showback/Chargeback: ........................................................................ 43
  Expose Service Catalog via RESTFUL APIs ........................................ 45
Conclusion ............................................................................................... 47
Product Overview

The ability to deliver database services to a company in a fast, secure, and cost effective manner is an extremely critical part of being successful in today's highly demanding IT environments. Traditionally, the Oracle database administrator provides the database services businesses need to move forward, but resource constains caused by manual approaches can hinder the ability to provide critical services in a timely manner, increasing business costs and slowing progress. Using Oracle Enterprise Manager 12c, the database cloud administrator can remove the resource and infrastructure barriers and deliver database services dynamically to match supply with demand. This paper provides guidance on how to architect, configure and manage a private database cloud using Oracle Enterprise Manager 12c. Additional guidance regarding high availability and consolidation best practices and the foundation of Database as a Service (DBaaS) can be found at the Maximum Availability Architecture (MAA) white paper below:

High Availability Best Practices for Database Consolidation: The Foundation for Database as a Service [Oracle MAA]

Private Database Cloud Enterprise Manager 12c Capabilities Overview

The Oracle Enterprise Manager 12c Database as a Service solution offers the most comprehensive capabilities to date, providing provisioning, dedicated databases, pluggable databases, schemas and clones all using a self-service model. The cloud capabilities are rapidly growing to satisfy additional business requirements. Below is a description of the current capabilities of each of the above categories.
Dedicated Database

In the area of dedicated databases, EM currently supports the following capabilities:

a) Provisioning of a Single Instance Database
b) Provisioning of a Real Application Cluster (RAC) Database
c) Provisioning of a RAC One Node Database
d) Provisioning of a RAC One Node Database DataGuard configuration
e) Provisioning of a Real Application Cluster Database DataGuard configuration
f) Provisioning of a Single Instance Database DataGuard configuration

Database 12c Multitenant

In the area of pluggable databases, EM currently supports the following capabilities:

a) Provisioning of an empty pluggable database
b) Provisioning of a pluggable database from a database gold image

Schemas

In the area of Schema provisioning, EM currently supports the following capabilities:

a) Provisioning of a user defined schema
b) Provisioning of imported schema from a database profile

Clone Including Snap Clones

In the area of Snap Clones, EM currently supports the following capabilities:

a) Provisioning of a snap clone database
b) Creating and restoring snapshots using the self service portal
Deploying a Private Database Cloud using Enterprise Manager 12c Cloud Control Cookbook

to utilize storage copy-on-write (or similar) technology to allow the self service user to create database clones.

The Snap Cloning technology offers the following benefits:

- **Space efficiency**
  When using the copy-on-write technology, storage blocks are not allocated until the blocks are updated, using this technique can achieve extreme storage savings in the magnitude of 90%.

- **Time efficiency**
  The copy-on-write technology uses snapshots to create backups that make the restoration of data as simple as updating the snapshot pointers to their original location. The create of snapshot or restoring the database to the previous version only takes a few minutes, even for very large databases.

- **Time travel**
  During application quality assurance testing, it is often required to go back to an earlier incarnation of a database. Using Snap Clone and the self-service portal, users can create multiple database copies for testing without consuming significant space.

In the example below, the self service portal user is able to create a snap clone. If necessary, the functionality also allows the self service user to restore the previous version of the database within minutes regardless of the size of the database. This technique provides provisioning agility, storage savings and reduction of administrative overhead.
Private Database Cloud Architecture Design

DBaaS Cloud Definitions

PaaS Infrastructure Zone:

A PaaS infrastructure zone is a group of hosts. Each host in a zone represents a location at which a Service Instance is to be potentially deployed. For instance, in an estate of two datacenters, you may want to define two Zones (Datacenter-A, Datacenter-B). The PaaS Infrastructure Zone allows you to define the placement policy constraints for a specified set of targets and the users to who can access targets within the Zone.

Service Template:

A Service Template is a standardized service definition that is offered to self service users to create dedicated databases, pluggable databases or schemas. The cloud administrators can then map the Service Templates to resource pools and Zones. This feature enables SSA administrators to build a self-service catalog tailored to the specific needs of a company.

Software Pools:

A software pool is a set of homogeneous resources. A Database Pool, which is created in DBaaS, is a collection of database homes, databases, or container databases depending on the type of cloud service model selected.

A software pool has the following constraints:

- A target can belong to only one software pool.
- The name of the software pool and the version cannot be modified after it has been created.
- All targets in a software pool must be homogeneous.
- A Service Template can use multiple zones but only one software pool within each zone.

Oracle Software Library:

The Software Library is one of the core features offered by Enterprise Manager 12c. The software Library is a repository that stores certified or custom software entities such as Software Patches, Deployment Procedures, Reference Gold Images, components, Application Software and their associated directive scripts. The Software Library enables you to select Oracle-supplied entities and customize them, or create a custom entity of
your own. Once defined, you can reference these reusable entities from a Deployment Procedure to automate the operations such as: patching, provisioning, and so on. The Reference File Location option of the Software Library allows an organization to leverage existing infrastructure for sourcing software and scripts. This is very useful for multi-site implementations to minimize the data transfer between targets and OMS.

The Reference File Location supports the following options:

- **HTTP**: An HTTP storage location represents a base URL.
- **NFS**: An NFS storage location represents an exported file system directory on a server.
- **Agent**: An Agent storage location can be any host monitored by an Enterprise Manager 12c Agent.

In global Enterprise Manager 12c implementations where targets can be distributed across the globe, moving large files over the WAN can be challenging. A Local Reference File Location can be utilized to store a database version 11.2.0.4 gold build image that can be used to provision database homes.

**Provisioning Profiles:**

A database provisioning profile is an entity that captures source database information for provisioning. It contains software configurations. A profile can represent a complete database or a set of related schemas that form an application. A single profile can be used to create multiple Service Templates.

A cloud administrator can create a database provisioning profile as a one-time activity, which can be used by operators for mass deployments.

**Defining Database Service Tiers and Catalogs**

Before engaging in an architecture design, we need to understand the term catalog and the types of catalogs within a Private Cloud architecture. Let's start with the Business Catalog. A business catalog describes the services that will be available to the cloud consumers in terms of availability.
Note: The business catalog named and described below is for illustration purposes only. Depending on the implementation, one customer's Gold tier may be another customer's Platinum etc.

Once the business catalog is defined, the business catalog must be converted to the Technical Catalog. The Technical Catalog describes the underlying technology utilized in order to implement the database services that will be available to the cloud consumers.
The last term to understand and define is the Self-Service Catalog. The Self-Service Catalog describes the database services that will be available to the cloud consumers in an automated self-service fashion.

Technical versus Self-Service Catalog

In an effort to improve the speed, agility and automation of the database provisioning, services from the technical catalog can be provided on-demand using the self-service model. Currently, the Oracle Enterprise Manager 12c cloud framework can satisfy all the above technical catalogs in a self-service model.

In very complex implementations, the self-service catalog is usually a subset of the technical catalog. A company can provide the Bronze and Silver tier in the self-service model but choose to deploy the Gold and Platinum tiers using native Oracle Enterprise Manager 12c capabilities.

Bronze Tier Example

The Bronze Tier provides a cost effective database solution where availability and data protection requirements are not as stringent. A database created under this tier can be recovered as of the last backup. In this tier, Oracle recommends the use of a single instance database installed on commodity hardware or Oracle Multitenant configuration. If the reference architecture is an engineered system, a single node RAC database is recommended.

Bronze Reference Architecture
Silver Tier Example

The Silver Tier provides clustering improving and reducing planned and unplanned downtime. Silver can utilize Oracle RAC One Node technology in order to achieve the HA requirements of the tier or a two node RAC. The Silver tier can be recovered as of the last backup. In this tier, Oracle recommends the use of RAC one node, two node RAC or a Multitenant RAC configuration.

Silver Reference Architecture

Gold Tier Example

The Gold Tier utilizes database replication technology to eliminate single point of failure and provides a higher level of data protection for all types of planned and unplanned outages including database corruption, database failures and site failures. For this tier, Oracle recommends the use of a RAC database with a RAC standby database.

Gold Reference Architecture
Platinum Tier Example

Mission critical tiers such as the Platinum Tier, should be implemented in its own reference architecture. In some cases, companies elect to provision these tiers using Oracle Enterprise Manager 12c native capabilities, as opposed to self-service. For this tier, Oracle recommends the use of a multi-node RAC database with multiple multi-node RAC standby databases.

Platinum Reference Architecture

Service Tier Recovery Point and Time Objectives

The service availability can be defined in terms of Recovery Time Objective and Recovery Point Objective. Oracle technologies, such as Real Application Clusters and DataGuard can significantly improve the RTO and RPO but introduce additional costs and complexities. In order to avoid unnecessary complexities and cost choose the database tiers wisely.

- The **Recovery Time Objective (RTO)** is the goal for how long it may take until a service is recovered from an outage/incident.
- The **Recovery Point Objective (RPO)** defines how much data may be lost as a consequence of an outage/incident, i.e. to which point in time relative to the incident the data needs to be recovered to.

The table below illustrates an RTO and RPO design in relation to the Technical Service Catalog. Please note that the availability of the service is improved or reduced when the database configuration changes.
The table below illustrates typical incidents and resolutions of this particular Technical Service Catalog.

<table>
<thead>
<tr>
<th>Service Tier</th>
<th>DB Instance loss/crash</th>
<th>compute node loss in an frame</th>
<th>storage loss within frame / DB loss</th>
<th>Full Frame Loss</th>
<th>Data Center loss</th>
<th>Data corruption / loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platinum</td>
<td>RAC automatically fails over to other instance within same cluster.</td>
<td>RAC automatically fails over to other instance within same cluster.</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td>Recover from flashback if detected within 30s, otherwise recover from backup</td>
</tr>
<tr>
<td>Gold</td>
<td>RAC automatically fails over to other instance within same cluster.</td>
<td>RAC automatically fails over to other instance within same cluster.</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td>Fail/-switch-over to 2nd Data Center, Re-establish DataGuard</td>
<td></td>
</tr>
<tr>
<td>Silver</td>
<td>Restart DB instance or Activate Passive Database Instance If problem persists recover from backup</td>
<td>Activate Passive Database Instance</td>
<td>Recover from flashback if not also affected otherwise recover from backup</td>
<td>Create new Database instance on frame with free capacity and recover from backup</td>
<td>Create new Database instance on frame with free capacity and recover from backup</td>
<td></td>
</tr>
<tr>
<td>Bronze</td>
<td>Restart DB instance when node is repaired or create new Database Instance for same DB on different node</td>
<td>Restart DB instance</td>
<td>Recover from flashback if not also affected otherwise recover from backup</td>
<td>Create new Database instance on frame with free capacity and recover from backup</td>
<td>Create new Database instance on frame with free capacity and recover from backup</td>
<td></td>
</tr>
</tbody>
</table>
Private Database Cloud Consolidation Guidelines

In the area of database consolidation planning, Oracle Enterprise Manager 12c provides a number of techniques to ensure that consolidated applications and database workload can exist in harmony, avoiding unnecessary impact on application performance and even outages. When consolidating a database the following steps should be considered:

- **Cloud Consolidation Model**

  During the design of the DBaaS, it is important to consider the placement of the database in terms of consolidation. Please review the table below.

<table>
<thead>
<tr>
<th></th>
<th>Consolidation Density</th>
<th>Management</th>
<th>Isolation</th>
<th>Implementation &amp; Onboarding</th>
<th>Application Suitability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dedicated DB</strong></td>
<td>High</td>
<td>Easy</td>
<td>Good</td>
<td>Easy</td>
<td>All</td>
</tr>
<tr>
<td><strong>Dedicated Schemas</strong></td>
<td>Highest</td>
<td>Easy to Involved (based on required resource isolation)</td>
<td>Least</td>
<td>Difficult</td>
<td>Home grown Requires application Validation</td>
</tr>
<tr>
<td><strong>12c Multitenant Database</strong></td>
<td>Highest</td>
<td>Easy</td>
<td>Good</td>
<td>Easy</td>
<td>All but has to be certified for database 12c</td>
</tr>
</tbody>
</table>

As illustrated above, Schema based consolidations offer ease of management and patching, but limited isolation. On the contrary, the 12c Multitenant database option provides the highest consolidation density. The 12c Multitenant database unique architecture provides efficient and flexible database consolidation by improving ease of database expansion and server resource utilization efficiencies, whilst reducing the operation management costs.

If consolidation density and isolation are the driving factors, Oracle recommends the Database 12c Multitenant database. Please also note that not all applications are suitable for all consolidation models. For additional details refer to the Oracle Multitenant Consolidation study published on the Oracle Technology Network.

- **Cloud Consolidation Planning**

  In the area of database consolidation planning, the Oracle Consolidation Planner, by leveraging metric and configuration data collected from managed target servers, help determine the optimum consolidation scenarios.
To utilize the Oracle Consolidation Planner to determine the optimum database placement, consider the following steps:

1. **Source Server**: Identify the source database server to be discovered by Enterprise Manager 12c. When a source server is discovered and data collection is executed against the server, Enterprise Manager 12c has very accurate data on the resources utilized by the server and the workload of all the databases running on the server.

2. **Destination Server**: Identify the destination database server that the source server will be consolidated to. The destination server could be a pre-existing server or event yet-to-be-purchased engineered system such as the Oracle Exadata Database Machine or an Exalogic Elastic Cloud system.

3. **Consolidation Projects and Scenarios**: With the source and destination server identified, the next step is to use the consolidation planner to create consolidation projects and consolidation scenarios. The Oracle Consolidation Planner provides out of the box, a set of pre-configured consolidation scenarios, representing conservative, aggressive, and medium consolidation schemes. Alternatively, custom scenarios can be created that best suit the business requirements. Once created, the various scenarios can be compared to determine which consolidation strategy best meets the business requirements.

- **Cloud Consolidation Testing**

Once a Cloud Consolidation Model has been selected, the Oracle Consolidation Planner can intelligently place databases into the correct destination server. The next step is to ensure that database production workloads are not impacted by the server configuration changes.

The best tool to meet this need is the Oracle Real Application Testing option, which enables the database administrator to perform real-world testing of Oracle Database workloads. Using Real Application Testing, the database administrator can capture production workloads and assess the impact of system changes on these workloads before production deployment. Using the Oracle Real Application Testing, the database administrator can minimize the risk of instabilities associated with system changes.

Oracle Real Application Testing consists of two key components that can be used for consolidation testing. The SQL Performance Analyzer and Database Replay. Depending on the level of the testing and the business requirements, the database administrator can use either or both components to perform the required testing. Please refer to the Oracle documentation on [Real Application Testing](https://www.oracle.com) for additional configuration details.
Configuration Pollution and Sprawl

This cloud dimension is concerned with the compute resources that should be selected for a given database instance as well as avoiding database configuration pollution and sprawl.

Note: Although, it is tempting to create custom database configurations and footprints it is not recommended.

<table>
<thead>
<tr>
<th>DBaaS Databases Size</th>
<th>Initial Database Size</th>
<th>Initial Core Allocation</th>
<th>Initial Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>50GB</td>
<td>1</td>
<td>6GB</td>
</tr>
<tr>
<td>Medium</td>
<td>250GB</td>
<td>2</td>
<td>12GB</td>
</tr>
<tr>
<td>Large</td>
<td>1250GB</td>
<td>4</td>
<td>24GB</td>
</tr>
<tr>
<td>Extra Large</td>
<td>5TB</td>
<td>8</td>
<td>48GB</td>
</tr>
<tr>
<td>Custom</td>
<td>Not Recommended</td>
<td>Not Recommended</td>
<td>Not Recommended</td>
</tr>
</tbody>
</table>

The CPU resources can initially be managed using Instance Caging.

Once the load profile of the estate is better understood, databases can be scaled up/down to the next database size as shown above in order to improve server utilization efficiency, database performance and avoid server pollution and sprawl.

In the Oracle Cloud Framework database sizing can be easily implemented by configuring Service Templates as shown below:

- Single-Instance-11204-SM-With DataGuard
- Single-Instance-11204-MD-With DataGuard
- Single-Instance-11204-LG-With DataGuard
- Single-Instance-11204-XL-With DataGuard
Implementation of a DBaaS using Cloud Management Pack

Oracle Cloud Management Pack utilizes the Oracle Cloud Framework to deliver automation, improve provisioning speed and avoids database pollution by standardizing the database environment. It allows Cloud administrators to define and implement service reference architectures and encapsulate database architecture complexities.

One of the key benefits of the Oracle Cloud Framework is faster deployment of typical platforms by moving away from administrator driven provisioning to end user driven provisioning. Enterprise Manager allows administrators to:

- Create resources pools
- Standardize and automate the deployment processes
- Publish pre-configured service catalog
- Setup access controls
- Manage roles, users and quotas to limit over-consumption
- Enable metering and chargeback on the consumed resources
- Optionally, expose the Service Catalog via RESTFUL APIs

Step 1: Configure Privilege Delegation for Database Servers

The Oracle Enterprise Manager 12c credentials framework allows administrators to configure two types of privilege delegation tools: Sudo and Powerbroker. From the Setup menu, select Security and then Privilege Delegation.

During the configuration of Privilege Delegation a SUDO or Powerbroker template should be created and applied to all the hosts.
Step 2: Defined Named Credentials

The Enterprise Manager credential subsystem enables cloud administrators to store Credentials, in a secure manner, as preferences or operation credentials. The credentials can then be used to perform different system management activities, such as real-time monitoring, patching, provisioning, and other target administrative operations. In a DBaaS implementation the configuration of the named and preferred credentials is critical. In larger implementations of DBaaS,
the preferred credentials are vital especially when using automation using emcli or when using the cloud framework in conjunction to the lifecycle management pack.

Step 3: Configure Software Library

The Oracle Enterprise Manager 12c Software Library is a required component of any DBaaS implementation. The storage location for the Software Library must be accessible to all OMS's as local directories. In a multi-OMS scenario, you must set up a clustered file system using OCFS2, ACFS or NFS. For single OMS systems, any local directory is sufficient.
Step 4: Creating Cloud Users and Roles

In the Oracle Enterprise Manager 12c, cloud framework users and roles are critical components in defining user quotas. Quotas are assigned at the role level and users should be assigned to the roles based on the desired limits. Below is an example of a quota definition of the role PAAS_ROLE.

To create a DBaaS role, navigate to Setup > Security > Roles.

In this example, we created a role named PAAS_ROLE. Please ensure that the EM_SSA_USER is assigned to the PAAS_ROLE. In large implementations, to implement segregation of system resources, it is recommended to create a number of PaaS Roles.
Step 5: Create Self Service Portal Users

To create a DBaaS user, navigate to Setup > Security > Administrators.
In this example a user named SSUSER is created. Please ensure that the SSUSER is assigned to the PAAS_ROLE, EM_USER and Public as shown below.

Step 6: Defining Databases Zones

1. Define and create zones to satisfy the Reference Architecture Design.

A zone is a collection of hosts that are used by the cloud framework placement algorithm to intelligently place databases into zones. The placement algorithm will derive database placements based on the user quota and the placement policy that was configured by the cloud administrator.

Depending on the business requirements the cloud administrator will create a number of Database Zones. In this simple example, the requirement is to have multiple datacenters that will satisfy the disaster recovery requirements of the organization. For this reason we have created two PaaS Infrastructure Zones.

1. DataCenter- A - Primary
2. DataCenter -B - Standby
Below are the basic requirements of the DBaaS PaaS Zone:

a) A zone can be of type Host
b) The placement policy is defined at the zone level
c) The credentials provided should be of global credential type and valid across all the members of the zone

A PaaS Infrastructure Zone can consist of different types of database configurations. For example:

a) Real Application Clusters
b) Host Configure Snap Clones
c) Single Instance 11.2.0.4 and 12.1.0.1 database server
d) Multitenant 12c database configurations

A PaaS Infrastructure Zone can be assigned to one or more Oracle Enterprise Manager 12c roles.

To create a PaaS Infrastructure Zone, navigate to **Setup > Cloud > PaaS Infrastructure Zones** and click on Create.
Follow the Enterprise Manager wizard and specify the following as shown below:

- PaaS Name: "DataCenter-A"
- Placement Policy
- Target Hosts
- Credentials
- Role
Step 6: Defining Database Pools

Navigate to the Setup > Cloud > Database and select "Database Pools"

**Database Cloud Self Service Portal Setup**

- Request Settings
- Quotas
- Profiles and Service Templates

Dedicated Database pool

Click on the Create button and select For Database. The create database wizard will guide you to create a dedicated database Service Template.

The example below is an illustration of a singleton database 11.2.0.4 Service Template with an optional DataGuard implementation.
The following information should be provided in order to configure the above Service Template:

a) Database pool name (select a distinct name that describes purpose of the pool in this example: DataCenter-A Linux64-11202-SI,
b) Provide **Global** credentials for an Oracle named credential. In this case, the grid and root credentials are optional,
c) Select the Hosts and database homes that will define the pool,
d) Select the PaaS Infrastructure Zone, Platform, database configuration and database version to be installed in the pool. If you are planning to support multiple versions, a second pool is needed as shown below:
e) In this example, a standby database PaaS and pool are also selected.

---

**Pluggable Database Pool**

**Database Cloud Self Service Portal Setup**

- ✔ Database Pools
- ✔ Request Settings
- ✔ Quotas
- ✔ Profiles and Service Templates

Click on the **Create** button ![Create] and select **Pluggable Database**. The create database wizard will guide you to create a pluggable database Service Template.
The administrator uses the placement policy to set a maximum ceiling for resources. This protects resource consumption for the members of the database pool. The maximum number of pluggable databases that can be created in a container is 252.
Step 7: Request Settings

Navigate to the **Setup > Cloud > Database** and select "Request Settings"

![Image of Oracle Enterprise Manager Cloud Control 12c](image)

Using the **request setting screen** the cloud administrator is able to specify the following:

- The amount of time in advance the self service user can schedule a request.
- The maximum amount of time for which a self service user can retain the instance.
- The amount of time after the completed self service create requests will be purged from the repository.

All the Request settings are at the self service portal level.
Step 8: Quotas

Navigate to the Setup > Cloud > Database and select "Quotas"

Using the Quota, the cloud administrator is able to assign to roles the following:

1. The amount of memory all the users assigned to the roles can consume.
2. The amount of storage all the users assigned to the roles can consume.
3. The number of databases all the users assigned to the roles can request.
4. The number of schema requests all the users assigned to the roles can request.
5. The number of Pluggable database requests all the users assigned to the roles can request.

All the Quota settings are at the role level.

Step 9: Defining Service Templates for Self Service Provisioning

Navigate to the Setup > Cloud > Database and select "Profiles and Service Templates"
Dedicated Database Service Template Creation

During the general section of the Service Template creation wizard, specify the following entries:

Description:
- Service Template name (Single-Instance 11.2.0.4 small database with DataGuard)
- Select the database creation template from the Software Library

Database Definition:
- For Real Application Cluster database Service Templates select the number of nodes
- If the database SID prefix is to be provided by the user during the interview process, select specify by user at request time
- Provide the database domain name
In case the request is for a DataGuard configuration, select `enable standby database` and provide the standby instance name and Protection Mode. The available Protection Modes are listed below:

- Maximum Performance
- Maximum Protection
- Maximum Availability

In the DataGuard use case, the active database option is also available. Please refer to the Oracle DataGuard documentation for additional details:

Pools and Zones:

- In the Pool and Zone section of the wizard, identify the primary and standby pool if it is desirable.

During the Database section of the Service Template creation wizard specify the following entries:

Storage Type:

- Depending on the reference architecture, select Automatic Storage Management or File System for the storage type.
Fast Recovery:
  • Enabling the Fast Recovery area and the Archiving will allow the Self Service user to schedule backups and restore operations.

RMAN Duplicate Backup Location:
  • This directory is required for DataGuard configurations. After the creation of the primary database an RMAN duplicate will be performed in order to create the database DataGuard configuration. The directory MUST be present on all the members of the primary and standby pools.

During the Initialization section of the Service Template wizard, provide any optional initialization parameters that are required.
During the Script section of the Service Template wizard, the cloud administrator can provide scripts that need to be executed before and after the database provisioning and de-provisioning. The pre and post scripts are mainly used for the integration of the new provision database with the organizations' eco-system.
During the Role section of the Service Template wizard, the cloud administrator provides which users are allowed to access the Service Template.

Review the configuration setting and save the Service Templates.

Pluggable Database Service Template
During the General Section of the Service Template creation wizard, specify the following entries:

Pluggable Database:
- Select Empty Pluggable or Pluggable Databases from Profile
- Identify the Zone and Pool where a database will be placed
- Identify the Reference Container Database which the Pluggable Database will be part of

During the Configuration section of the Service Template creation wizard, specify the following entries as shown above:
During the Initialization section of the Service Template wizard, provide any optional initialization parameters that required to be set.

During the Script section of the Service Template wizard, the cloud administrator can provide scripts that need to be executed before and after the database provisioning and de-provisioning. The pre and post scripts are mainly used for the integration of the new provision database with the organizations' eco-system.
During the Role section of the Service Template wizard, the cloud administrator provides which users are allowed to access the Service Template.

Review the configuration setting and save the Service Templates.
Step 10: Using Self Service Portal

Requesting Dedicated Database

Navigate to **Enterprise > Cloud > Self-Service Portal**

1. Select Databases from the Manage dropdown menu
2. Click the Request button and select Database
3. Select the Service Template

Provide the requested information and click Submit.
After the database request is created it can be managed again through the Self-Service-Portal. The screen below shows a single instance and DataGuard request.

Requesting Pluggable Database

Navigate to Enterprise > Cloud > Self-Service Portal

1. Select Databases from the Manage dropdown menu
2. Click the Request button and select Pluggable Database
3. Select the Pluggable Database Service Template
Provide the requested information and click Submit.

After the pluggable database request is completed, the pluggable database can be managed again through the Self-Service-Portal. The screen below shows a pluggable database 12.1.0.1.
Managing Your Private Database Cloud

Defining Monitoring Architecture

Setting up Oracle Enterprise Manager's monitoring architecture requires the configuration of the following features. The features below should be carefully designed and configured to ensure the Enterprise Manager targets are monitored, are in compliance, alerted and optionally chargeback properly.

Additional guidance on monitoring strategies can be found on the following white paper *Strategies for Scalable, Smarter Monitoring using Oracle*.

1. The Administrations Groups and Template Collections
2. Incident Management and Incident Rules
3. The Compliance Framework
4. Metering and Chargeback/Showback

Administration Groups

In larger implementations, not all targets have the same criticality. Targets that support critical applications should be treated differently than development targets. Setting up the Enterprise Manager Administration Groups will allow the cloud administrator to segregate the Enterprise Manager Estate based on different usage profiles. A large Estate is likely to have different sets of monitoring settings (i.e. metrics, thresholds, collection schedules, corrective actions) for each usage profile. The details of specific metrics to monitor, the threshold values to use, as well as designing the Administration Groups hierarchy are outside the scope of this paper. The example below illustrates an Estate where the targets have been segregated by lifecycle status, database tier and server platform.
Incident Management and Incident Rules

Incident Management is a functional area in Oracle Enterprise Manager 12c that builds upon the existing monitoring capabilities. The main goal of Incident Management is to enable the cloud administrator to monitor and resolve service disruptions quickly by business priority. Instead of managing numerous discrete events, this feature enables the creation and management of fewer incidents.

Oracle Enterprise Manager 12c also provides features to assign, track, diagnose, notify and resolve the incidents.

Incident Rules enable the cloud administrator to automate operations on events, incidents, and problems. For example, you might want to create a rule that will automatically create an incident for target down events, or you might have another rule that sends notifications for critical incidents. In order to eliminate unnecessary alerts, the Enterprise Manager cloud administrator should implement a number of incident rules to tunnel alerts properly.

The example above illustrates the creation of an Incident Rule that enables the cloud administrator to limit the amount of alerts that are forwarded to operations. The Rule "Production_Critical_Metric_Alerts" will only forward alerts to operations(using Netcool), if the target property *Lifecycle Status* is set to Production or Mission Critical.
Compliance Framework

In a private cloud implementation, it is required to measure the security and overall compliance of DBaaS, as well as provide the high level compliance state of the estate to the business. In addition, the details of every violation should be provided to the compliance administrator in order to further investigate and resolve the violations. This can be easily achieved using the Oracle Enterprise Manager 12c Compliance Framework.

Oracle Enterprise Manager 12c uses a hierarchical approach to compliance monitoring. Starting with Compliance Frameworks, Compliance Standards and Compliance Rules. Oracle Enterprise Manager 12c, out of the box, provides 300 Compliance Rules that can be utilized immediately.

In addition to the Compliance Rules, Oracle Enterprise Manager 12c also provides, out of the box, a number of preconfigured Compliance Frameworks that are shown below:

For instance, the Payment Card Industry Data Security Standard (PCI DSS).

Depending on the business requirements it may be necessary to create custom Compliance Frameworks and Standards that satisfy additional business requirements. Additional information on configuring and managing the DBaaS Compliance can be found in the Oracle documentation.
Showback/Chargeback:

In a private cloud implementation, it is often required to measure resource utilization of the cloud translated into currency and Chargeback the organization that utilizes the resources. Oracle Enterprise Manager’s 12c metering and chargeback feature which are part of the Oracle Consolidation Planning and Chargeback plug-in uses the collected monitoring and configuration data to derive metric and chargeback data.

Oracle Enterprise Manager's 12c metering and chargeback features provide the cloud administrator the following capabilities:

- Ability to meter Oracle Enterprise Manager 12c targets
- Ability to assign rates to a metered resource
- Ability to manage a Cost Center hierarchy for Chargeback
- Ability to assign resources to a Cost Center

With the integration of Oracle Enterprise Manager 12c and the BI Publisher, the chargeback administrator can create custom reports that fit the needs of the business in addition to the reports that are available out of the box.

Oracle Enterprise Manager 12c enables the chargeback administrator to define external recipients for chargeback reports and choose from a variety of formats such as HTML, PDF, RTF, Excel and PowerPoint. BI Publisher is able to publish these reports to a website/portal or email them directly to the recipients.

The basis for metering and chargeback are metrics collected by Enterprise manager 12c. There are three types of chargeback metrics, configuration-base, availability-base and usage-base.

1. Configuration-based metrics refer to metrics such as number of CPUs or database versions; these metrics are considered static and have a daily aggregation.
2. Availability-based metrics measure the time an entity is available, also known as an entity's uptime. It is the number of minutes in a given hour an entity is considered available. The charge is prorated for downtime.
3. Usage-based metrics refer to metrics such as CPU utilization or memory consumption and have an hourly aggregation.
The table below is an example of Oracle pluggable database chargeable entities and their associated metrics. With the latest version of Oracle Enterprise 12c, version 12.1.0.4, custom target types and metrics can be added to satisfy additional requirements.

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Item</th>
<th>Resource Category</th>
<th>Source Type</th>
<th>User-Defined</th>
<th>Item Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Pluggable Database</td>
<td>Base Charge</td>
<td>Instance</td>
<td>fixed</td>
<td></td>
<td>Base charge for an Oracle Pluggable Database</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>CPU Count</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Logic CPU core count</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>CPU Time</td>
<td>Instance</td>
<td>metric</td>
<td></td>
<td>CPU time used by PDB instance (second)</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>CPU Utilization</td>
<td>Instance</td>
<td>metric</td>
<td></td>
<td>Percentage used of the total CPU time</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>CPU Utilization (SPECint(R)_rate_base2006)</td>
<td>Instance</td>
<td>metric</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>DB Time</td>
<td>Instance</td>
<td>metric</td>
<td></td>
<td>Database time used by PDB instance (seconds)</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Disk Read (Physical Operations)</td>
<td>Storage</td>
<td>metric</td>
<td></td>
<td>Count of reads from all physical disks.</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Disk Write (Physical Operations)</td>
<td>Storage</td>
<td>metric</td>
<td></td>
<td>Count of writes to all physical disks.</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Edition</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Database edition</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Machine Architecture</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Host CPU architecture</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Option</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>PDB instance database options or features used</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>RAC Node Count</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Number of the nodes in the rac</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Release</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Database release</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>SQL Executes</td>
<td>Activity</td>
<td>metrics</td>
<td></td>
<td>Total SQL execution count</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Tablespace Size</td>
<td>Storage</td>
<td>config</td>
<td></td>
<td>Total PDB instance disk space allocation (GB)</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Uptime</td>
<td>Uptime</td>
<td>metric</td>
<td></td>
<td>Total instance up time (hours)</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>User Transactions</td>
<td>Activity</td>
<td>metric</td>
<td></td>
<td>Total user transactions</td>
</tr>
<tr>
<td>Oracle Pluggable Database</td>
<td>Version</td>
<td>Instance</td>
<td>config</td>
<td></td>
<td>Database version</td>
</tr>
</tbody>
</table>

Additional information on metering and chargeback can be found in the Chargeback whitepaper.
Expose Service Catalog via RESTFUL APIs:

The Oracle Enterprise Manager 12c Cloud Management Pack provides the ability for integration with external web services using RESTFUL API's that use JSON payloads and operate on the Enterprise Manager cloud resource model to drive automation from external systems. The RESTFUL API's help integrate existing IT infrastructure and resources with cloud workflow, as well as connect different cloud services together and can be utilized to create custom self service portals for cloud workflows.

The list of the currently supported Restful API's are listed below:

Cloud Resource Model

The figure below shows the DBaaS Resource Model:
The following table describes the operations that are supported by the Database as a Service family type.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud</td>
<td>A cloud represents the user’s starting view of all accessible resources</td>
<td>√</td>
</tr>
<tr>
<td>Service Family Type</td>
<td>A service family type is a category of services that are offered by the cloud. For example, IaaS, DBaaS, MWaaS, etc. These categories are predefined.</td>
<td>×</td>
</tr>
<tr>
<td>DBaaS Zone</td>
<td>A DBaaS zone represents a logical boundary where the resources may reside to support deployment of database services</td>
<td>×</td>
</tr>
<tr>
<td>DB Platform Template</td>
<td>A DB platform template represents the definition of the deployable database service</td>
<td>×</td>
</tr>
<tr>
<td>DB Platform Instance</td>
<td>A DB platform instance describes the metadata about databases deployed using the service templates.</td>
<td>×</td>
</tr>
</tbody>
</table>

The following table describes the DBaaS Resource Model specifications for all the supported API's.

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Format</th>
<th>Media Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud</td>
<td>/em/cloud</td>
<td>application/oracle.com.dbaas.com.cloud+json</td>
</tr>
<tr>
<td>Service Family Type</td>
<td>/em/cloud/service_family_type/dbaas</td>
<td>application/oracle.com.dbaas.com.ServiceFamilyType+json</td>
</tr>
<tr>
<td>DBaaS Zone</td>
<td>/em/cloud/dbaaS/zone/&lt;zoneid&gt;</td>
<td>application/oracle.com.dbaas.com.DbaasZone+json</td>
</tr>
</tbody>
</table>
The following example illustrates a RESTFUL API request that retrieves the available service templates. The table above illustrates additional requests which can be composed.

Conclusion

The latest release of Enterprise Manager 12c Cloud Framework provides a abundance of new features and together with the Enterprise Managers' core features makes the Enterprise Manager 12c a critical data center application. Using Enterprise Manager 12c infrastructure, organizations can remove infrastructure barriers and significantly improve delivery of database services, compliance, availability and performance of critical applications.