DBaaS Private Cloud using Enterprise Manager Cloud Control 12c and Exadata Database Machine

ORACLE

What is Database-as-a-Service (DBaaS)?

- Cloud based
- Self service oriented
- Administrators publish a catalog of templates
  - DBCA templates, RMAN backups, schema exports, thin clones
- End users request services based upon the templates

ORACLE

A Cloud Called Database-as-a-Service

Cloud Based

Database-as-a-Service (DBaaS) is a cloud based service that provides end users with the capability to request a database or schema from a catalog of published templates. Once requested, the service is provisioned within the DBaaS cloud and connection details are provided to the user. DBaaS is a variant of a Platform-as-a-Service (PaaS) cloud in that end users request a platform upon which applications can be developed. Other cloud types are Infrastructure-as-a-Service (IaaS) where requests are made for an operating system and associated resources, and Software-as-a-Service (SaaS) where requests are made for access to a running application.

Cloud Based?

ORACLE

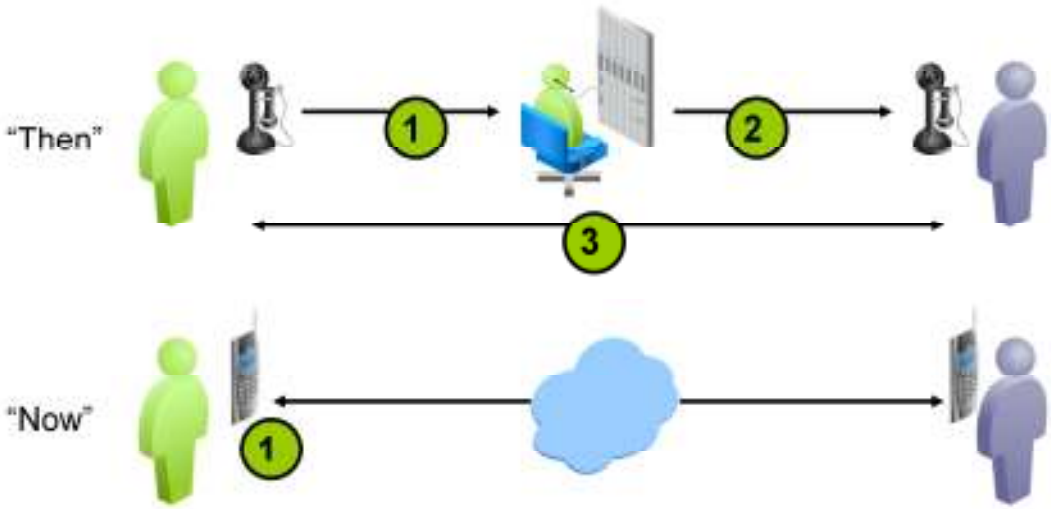
1 - 3

Lost on a Sea of Clouds?

Cloud computing has become the ubiquitous buzz phrase of enterprise IT in the 21st Century and is now filtering down into the consumer technology arena. Navigating the world of cloud computing can be daunting, however it could also be argued that it represents a return to the roots of computing when hardware resources were prohibitively expensive and hence by necessity were usually a shared resource about which end users knew very little.

The Cloud Is Not Such A Big Deal

- The cloud is a metaphor for obfuscated service provision.
- The cloud metaphor is neither new nor exclusive to IT.
  - Consider the phone call then and now.



ORACLE

1 - 4

We Consume Cloud Services Every Day

The cloud metaphor is invoked whenever the details of an implementation or provision of a service are obfuscated from the perspective of the end consumer. There are many examples of cloud-based services in everyday life, such as a phone call between two parties. Before the advent of the automated exchange, making a phone call involved multiple steps of which you were cognizant of:

1. Call the operator and ask for a connection to the other party.
2. Wait for the operator to connect you to the other party.
3. Converse with the other party (with the possibility of the operator sniffing all voice traffic).

Hence, when making a phone call, you were aware of how the call was routed to the other party, and could possibly even trace the phone lines along which the call had been transmitted.

These days, making a phone call only requires access to a phone service and knowledge of the other party's phone number. How your call is actually connected to the other party is obscured within the telecommunications cloud, and in this era of voice-over-IP, mobile communication, and global roaming, the means of communication is further obfuscated.

Other cloud services in your daily life could include:

- Your milk supply. Once, you would have known what dairy produced your milk, but today that information is obscured by large milk producers and their distribution network.
- The electricity supply grid. The power you consume may have been generated in another country.
- Mail delivery network. Whereas once your post could be guaranteed to be collected, sorted, and delivered by a single government agency, now there could be multiple independent contractors between the sender and recipient.

In all of these examples, the service or product you consume has not altered, only the way in which it is provided has changed.

To further illustrate how the cloud is already part of our everyday life, consider one more example:

- Browsing a website. You are typically unaware of the route taken by the HTTP packets between your browser and the site's web servers, or even where the web server is physically located.

Cloud Computing Explained

- From the consumers' perspective:
  - It is an IT capability or service.
  - Its implementation is both unknown and unimportant.
  - It is available anytime, from anywhere.
- From the providers' perspective:
  - It allows them to use computing resources efficiently, wherever, and however possible.
  - It allows for flexibility in resource allocation to meet varying consumer demand.
  - Meeting agreed service levels is more important than anything else.

ORACLE

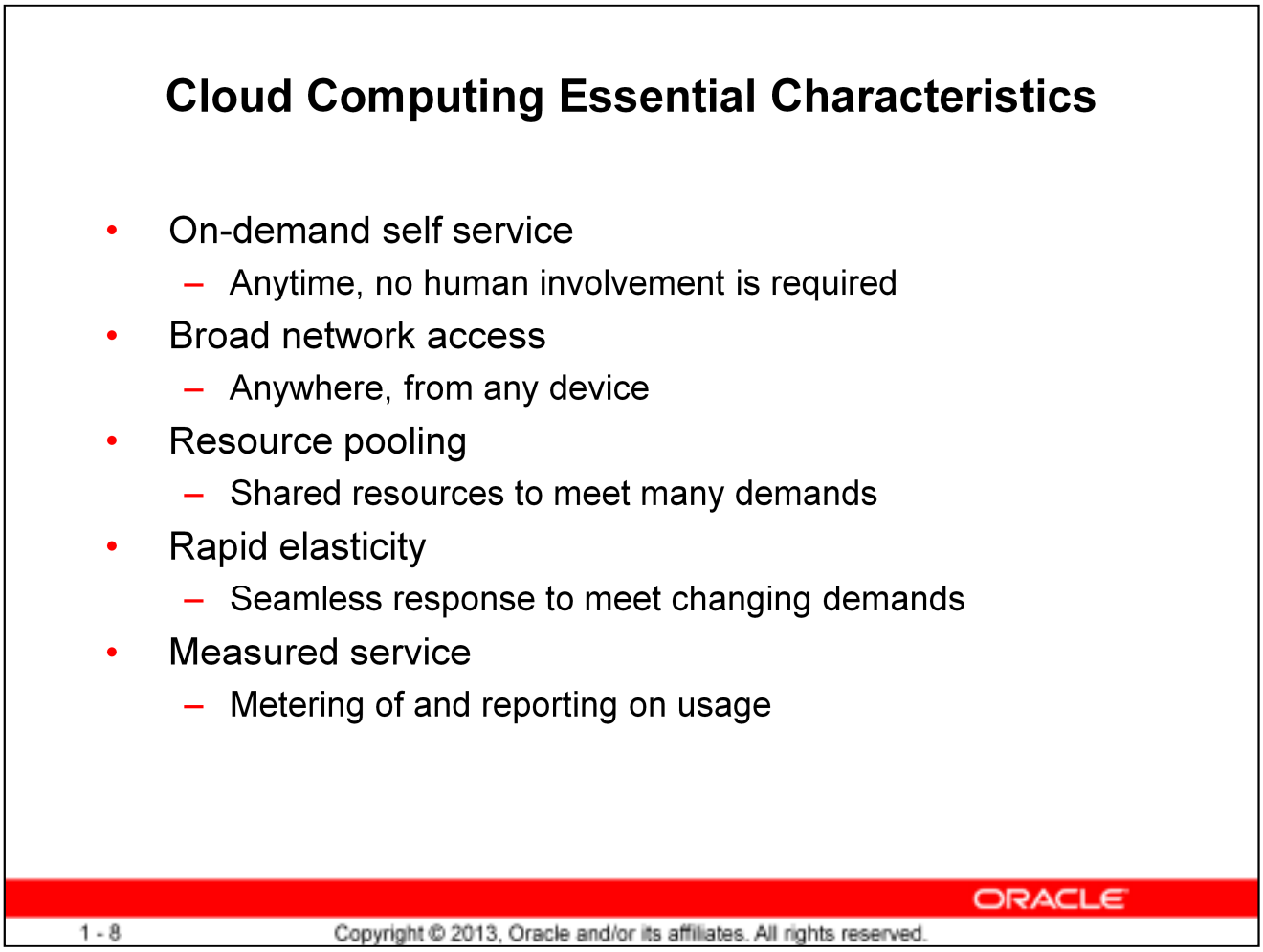
The US National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Every Cloud Has a Silver Lining

Cloud computing takes on a different aspect depending upon your perspective. However, all aspects present views of benefits that can be derived from cloud computing.

From the perspective of a consumer of cloud-based resources, the cloud is simply a capability or service that is used without having knowledge of how or where it is implemented. Indeed, knowledge of how the consumable product is provided is obscured by the very nature of it being accessed via "the cloud." Because implementation details are of no concern to the consumer, their primary interest is availability and usability.

From the perspective of a provider of cloud-based resources, the cloud allows them to service consumer demand by using whatever computing resources are available. This loosens the ties of physical resources to application topologies and gives the provider the flexibility and agility to deploy resources in the most efficient and timely manner possible. Like consumers of cloud-based resources, providers are also primarily interested in the availability and usability as the efficacy of their offering will be determined by the consumers' satisfaction with that offering, typically defined and measured through service-level agreements.



Five Essential Characteristics of Cloud Computing

The NIST definition of cloud computing includes five essential characteristics:

- **On-demand self service**

  Consumers can request and receive computing resources as required without human intervention by a provider.
- **Broad network access**

  The resources and self service portals provided by the cloud can be accessed through standard network-connected devices (for example, mobile phones, tablets, laptops, and workstations).
- **Resource pooling**

  Providers use their computing resources to serve the demands of many customers in a multi-tenancy model. All customers can then benefit from the dynamic allocation of resources from the pool to meet their demands. Examples of resources include storage, processing, memory, and network bandwidth.
- **Rapid elasticity**

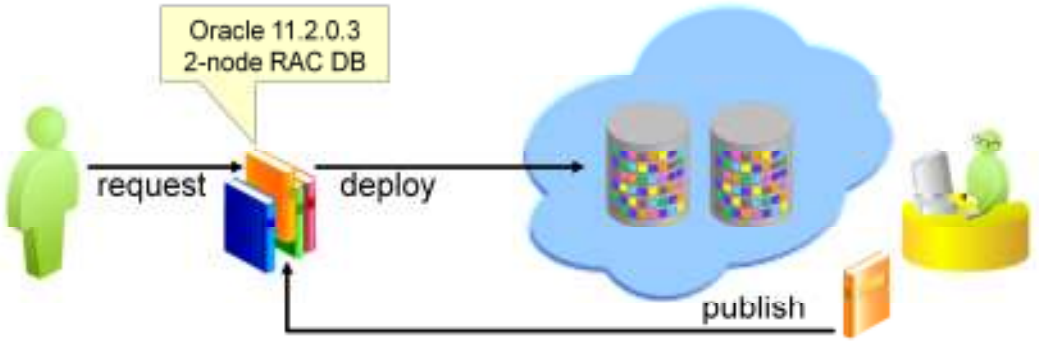
  Capacity of the cloud can be scaled up or down in response to consumer demand in a manner that appears to the consumer to provide unlimited capabilities.

- **Measured service**

  Consumption of cloud resources is measured in a manner appropriate to the service (for example, storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported upon by providers and consumers alike.

What is Database-as-a-Service? (continued)

- Cloud based
- Self service oriented
- Administrators publish a catalog of templates
  - DBCA templates, RMAN backups, schema exports, thin clones (on NetApp and ZFS only)
- Users request databases or schemas from the catalog



ORACLE

A Cloud Called Database-as-a-Service

Cloud Based

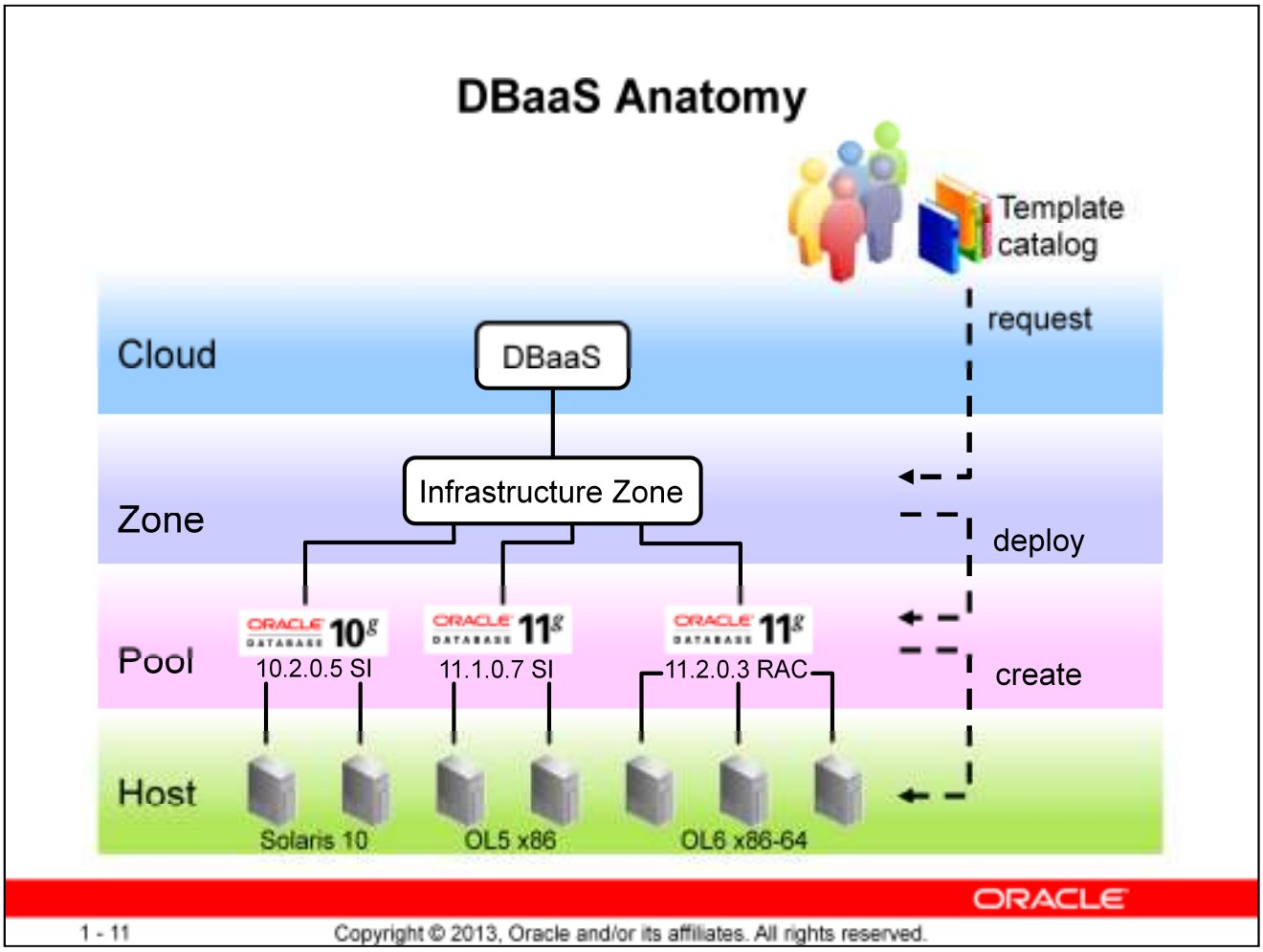
Database-as-a-Service (DBaaS) is a cloud based service that provides end users with the capability to request a database or schema from a catalog of published templates. Once requested, the service is provisioned within the DBaaS cloud and connection details are provided to the user. How or where the service is provisioned is neither exposed nor important to the end user.

Self Service Oriented

By definition, cloud services should provide a self service capability to end users, thereby eliminating the need for administrative involvement in provisioning requested services. DBaaS conforms to the definition by presenting end users with the Enterprise Manager Cloud Control 12c Self Service Portal pages where they can request and control services.

A Catalog of Templates

End users request services from a catalog of templates that are created and published by DBaaS administrators. Templates may be based upon DBCA templates, RMAN backups, schema exports, or thin clones of existing databases. DBaaS administrators nominate where services based upon the template can be provisioned, and who can provision them.



DBaaS Cloud Anatomy and Nomenclature

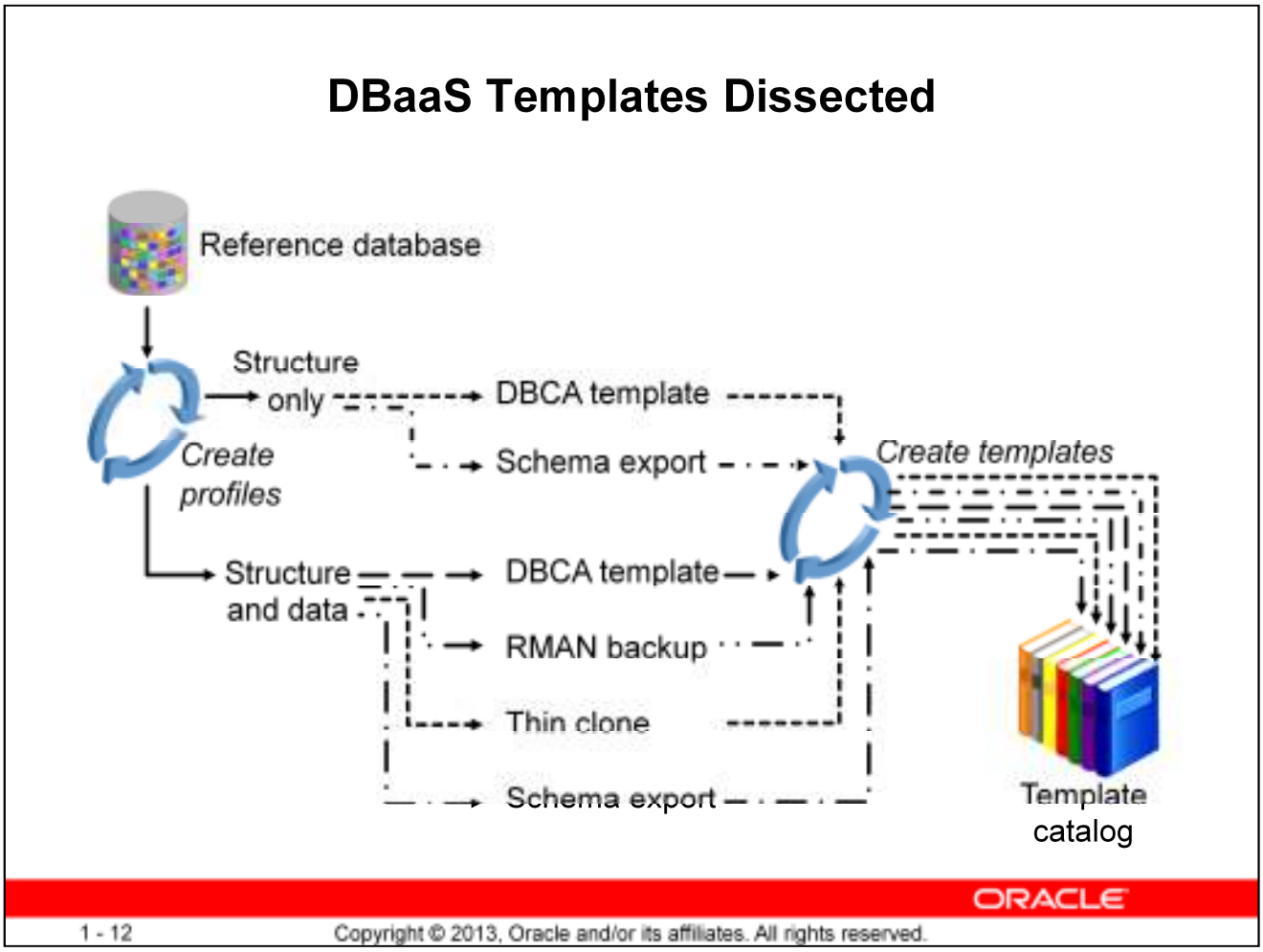
The anatomy and nomenclature of Enterprise Manager Cloud Control 12*c* clouds will be mentioned from this point on, so it is timely to introduce some of those elements for DBaaS clouds.

Clouds, zones, pools, and hosts

At an abstracted level, Enterprise Manager Cloud Control 12*c* clouds are composed of logical zones that may represent a physical geography such as East Coast and West Coast or functional topology such as Development and Testing. Zones are composed of logical pools and pools are composed of targets on hosts. The pools in DBaaS zones are collections of one or more Oracle Database homes (used for database requests) or databases (used for schema requests) of the same platform and version (for example, Oracle Database 11.2.0.3 RAC on Oracle Linux 6 x86-64).

Request from the zone, deploy to the pool, create on the hosts

Self service users request resources at the zone level from a catalog of templates. Enterprise Manager Cloud Control 12*c* will then determine which pool in the chosen zone can be used to satisfy the request, and appropriate jobs will be initiated on one or more hosts in that pool to create the entities required to fulfill the request.



From Reference Database to Template Catalog via Profiles

Creating the catalog of services that DBaaS users can use begins with the creation of a profile which is typically based upon an existing database managed by Enterprise Manager Cloud Control 12c, known as the “reference database”. The profile is then used to create a template that is published in the self service users’ catalog. One profile can be used to create many templates, each with different initialization parameters and other configuration details.

With Data or Not With Data, that is the First Question

When creating a profile from a reference database, you must decide if you want to include data from the reference database, or simply the structure of the reference database.

Profile format

Profiles can be created as one of a number of different formats, depending upon whether data is to be included or not:

- DBCA template – create a Database Creation Assistant (DBCA) template, with or without data
- RMAN backup – create a new Recovery Manager (RMAN) backup or use an existing backup of the reference database, with data

- Schema export – export one or more schemas in standard Oracle Export/Import format, with or without data
- Thin clone – create a thin clone of the reference data if using a supported storage technology (currently NetApp or Sun ZFS), with data

From Profile to Template

Once a profile has been created, you can use it to create one or more templates, adjusting initialization and other parameters for each template to create a varied catalog. Templates are based upon a single profile and are published to a combination of infrastructure zone(s) and self service role(s), thereby becoming available to users with those roles and with privileges on those zones.

Why Use Database-as-a-Service?

- Standardization – hardware, software, service templates
- Consolidation – hardware, administration
- Centralization – hardware, operations
- Optimization – computing and human resources
- Abstraction – product capability over installation details
- Flexibility – inherent in the cloud paradigm
- Self service – reduce the dependency of users upon DBAs

ORACLE

A Cloud Over Your Head Is Not Always Bad

The move to a computing cloud can be compelling for an organization on many fronts:

- Standardization
  - The cloud model naturally lends itself to the adoption of standards including hardware, application platforms, and integration technologies. The service template catalog for end users forces standardization of provisioned services.
- Consolidation
  - Combining physical infrastructure and IT budgets across multiple departments
  - More efficient purchasing, installation, maintenance, and operation processes
  - Reduced operational overhead with fewer physical pieces of infrastructure to manage
  - Potential gains per application deployment in terms of available compute resources
- Centralization
  - Consolidation may result in co-location of previously distributed infrastructure
  - Simplification of operations through the reduction of infrastructure

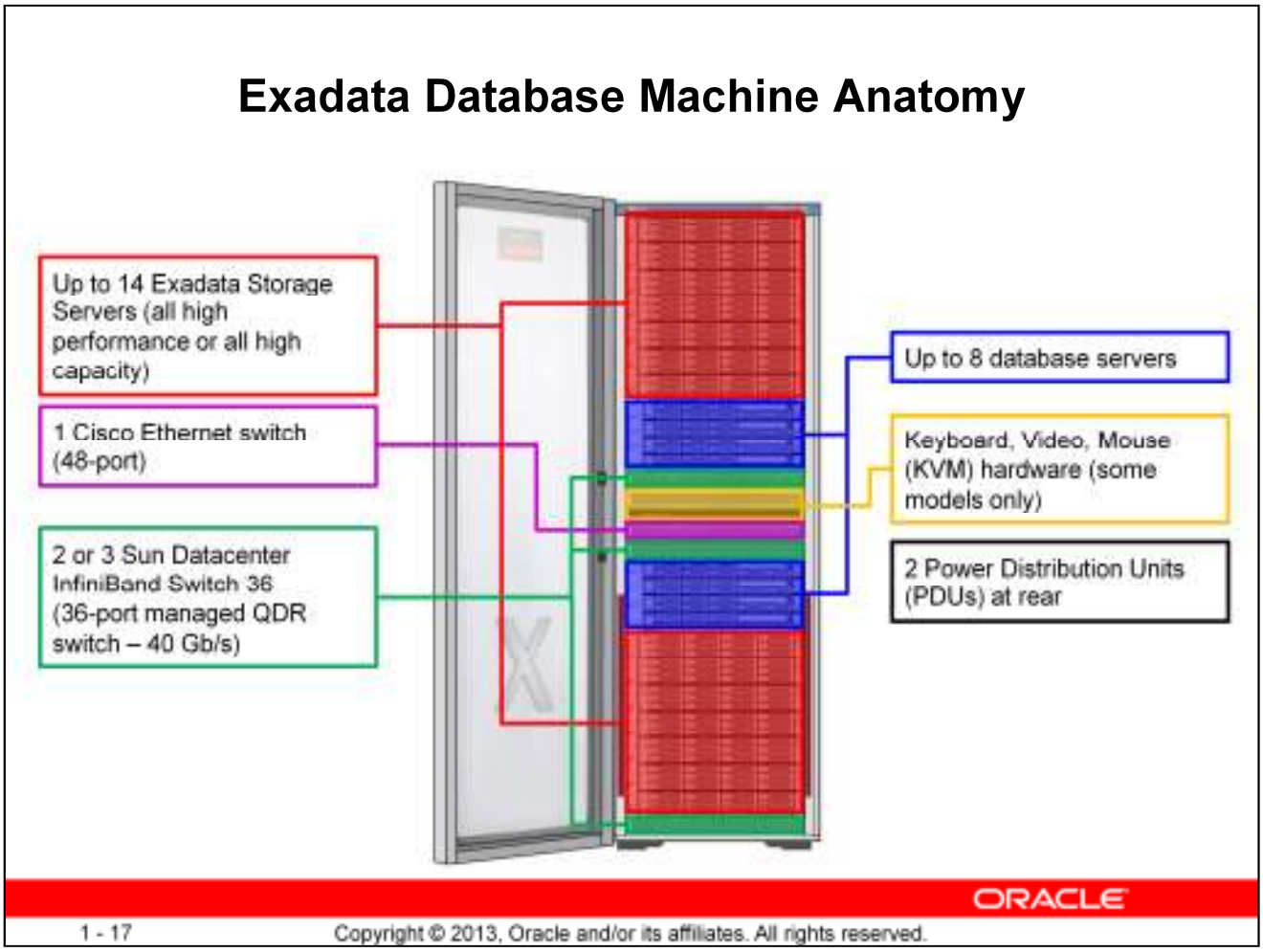
- Optimization
  - Taking advantage of all available computing resources on any given server
  - Potential for reduced workload on operations or systems staff with fewer resources to manage
- Abstraction
  - The host and application platforms in a cloud deployment become an abstraction of physical servers with installed software, breaking ties to specific physical resources.
- Flexibility
  - As long as any service-level agreements that are in place are not breached, IT resources can be deployed and used in whatever configuration and manner is desired.
  - Applications can be deployed on any environments that meet their current resource needs.
- Self service
  - Where suitable, allow end users (for example, developers) to request resources and have the cloud management system provide those resources, thereby freeing system administrators and operations teams from servicing such requests.
  - End users need only be concerned with any quotas and other restrictions imposed upon them, not where they might be able to find the resources they need as these will be provided by the private cloud.

What is an Exadata Database Machine?

- Fully integrated platform for Oracle Database
- Based on Exadata Storage Server storage technology
- High performance and high availability for all Oracle Database workloads
- Balanced hardware configurations
- Well suited as a database consolidation platform
- Simple and fast to implement

ORACLE

Oracle Exadata Database Machine is a fully integrated Oracle Database platform based on Exadata Storage Server storage technology. It provides a high-performance, highly available solution for all database workloads, ranging from scan-intensive data warehouse applications to highly concurrent OLTP applications.
Special attention has been paid to ensure that Exadata Database Machine is a well balanced platform. Throughout its hardware architecture, components and technologies have been specially matched to eliminate bottlenecks while maintaining good hardware utilization.
By using the unique clustering and workload management capabilities of Oracle Database, Exadata Database Machine is well suited for consolidating multiple databases onto a single environment. Delivered as a complete package of software, servers, and storage, Exadata Database Machine is simple and fast to implement.
Note: While Exadata Database Machine is a fully integrated platform solution comprised of specific hardware and software components, Oracle offers the hardware and software components as a series of separately purchasable items. Customers can choose from the different hardware configurations that are available. Appropriate licensing of Oracle Database and Exadata cell software is also required. In addition, Exadata Database Machine is highly complementary with clustering and parallel operations, so Oracle Real Application Clusters and Oracle Partitioning are highly recommended software options.



One Rack Fits All Exadata Database Machines

Exadata Database Machine is available in a few different configurations, all of which are assembled within a standard rack and cabinet to allow for possible future expansion. Exadata Database Machine topologies can span multiple racks if required. An Exadata Database Machine is comprised of the following components:

- 2, 4, or 8 high-performance database servers, with differing specifications depending on the model of the Exadata Database Machine
- 3, 7, or 14 Exadata Storage Servers, each with 12 high performance (15,000 RPM 600GB) or high capacity (7,200RPM 3TB) disks, and up to 1.6 TB of Exadata Smart Flash Cache
- 2 or 3 QDR (40Gb/sec) 36-port Infiniband switches
- Keyboard, Video, Mouse for console access (some models only)
- 2 Power Distribution Units

Why Use a Database Machine?

Database Machine is designed to address common issues:

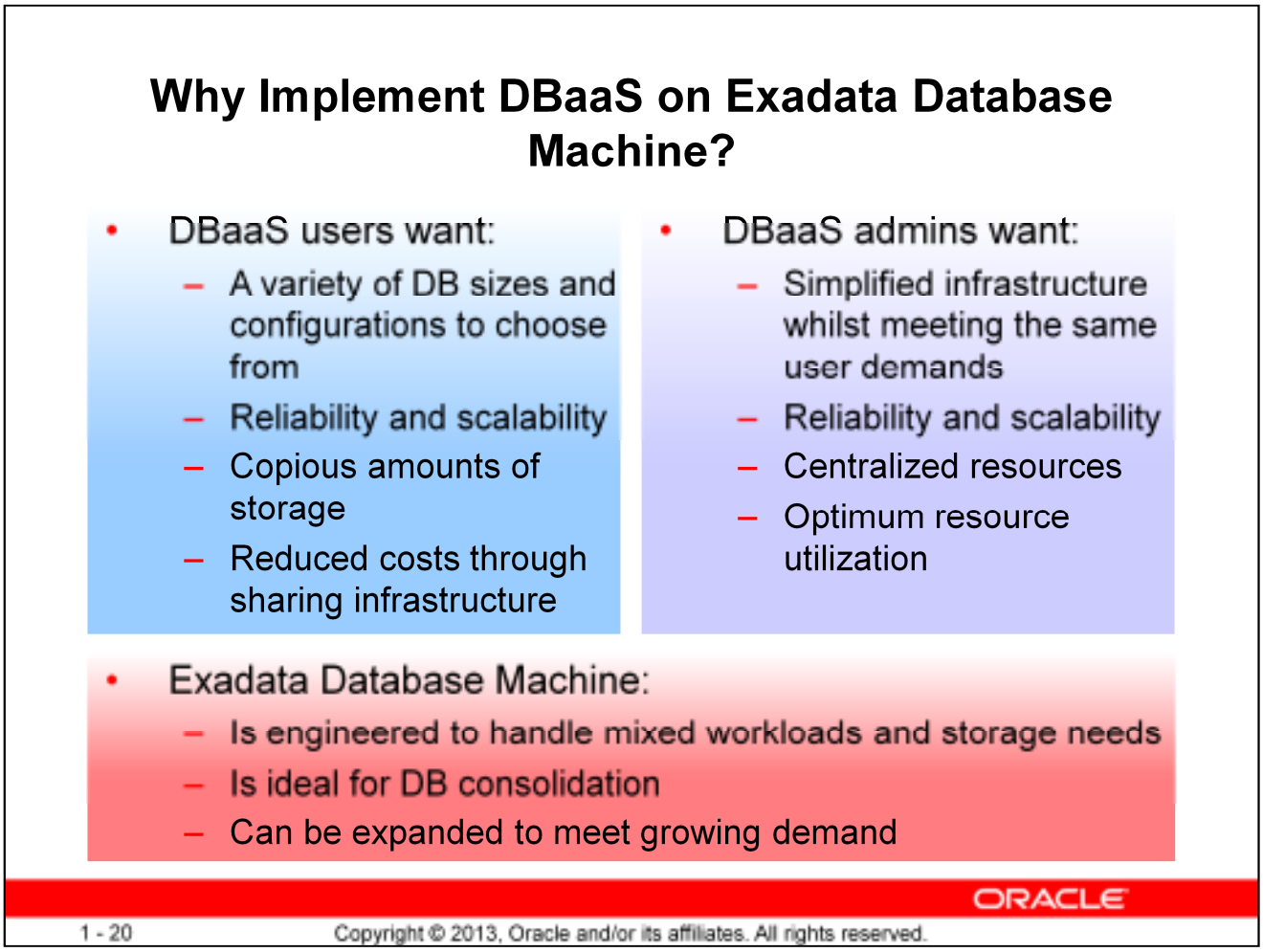
- Data Warehousing issues:
  - Supporting large, complex queries
  - Managing massive databases
- OLTP issues:
  - Supporting large user populations and transaction volumes
  - Delivering quick and consistent response times
- Consolidation issues:
  - Efficiently supporting mixed workloads
  - Prioritizing workloads
- Configuration issues:
  - Creating a balanced configuration without bottlenecks
  - Building and maintaining a robust system that works

ORACLE

Oracle Exadata Database Machine is an engineered system that is designed to address common issues faced by most database users, especially as databases increase in size and complexity. Exadata Database Machine technologies and practices address the following issues and requirements:

- Data Warehousing issues:
  - Supporting large, complex queries:
    - Getting enough I/O throughput to support massive scans
    - Driving the I/O throughput over the storage network
    - Avoiding unproductive I/O
    - Parallel processing
  - Managing massive databases:
    - Easily and effectively manage storage space
    - Use resources effectively while controlling runaway queries
    - Efficient data compression

- OLTP issues:
  - Supporting large user populations and transaction volumes:
    - Getting enough I/Os per second
    - Caching frequently accessed data
  - Delivering quick and consistent response times:
    - Minimizing I/O latency
    - Efficient commit processing
- Consolidation issues:
  - Efficiently supporting mixed workloads:
    - Supporting different workloads on the same system
    - Isolating workloads to avoid conflict
  - Prioritizing workloads:
    - Managing resources based on established priorities
    - Dynamically adjusting resource allocations based on current system configuration and
      workload observations
- Configuration issues:
  - Creating a balanced configuration without bottlenecks:
    - Hardware components matched with each other across the system
    - Hardware specifications matched to software capabilities
  - Building and maintaining a robust system that works:
    - Hardware, firmware, and software compatibility
    - Configuration best practices aiding consistency and supportability
    - Intelligent monitoring tools



DBaaS and Exadata Database Machine Complement One Another

A key aspect of a DBaaS cloud is that end users can select from a catalog of databases that meet a variety of needs. These needs could include a selection of single and multiple instance (RAC) databases, different initialization parameters, different security configurations, pre-seeded data or no data, and so forth. Bearing in mind that simplicity for one side of a user interface often implies complexity on the other, providing such a varied DBaaS self service catalog would typically require reasonably complex infrastructure to support the variety of databases represented in that catalog. The capabilities of Exadata Database Machine are ideally suited to addressing this aspect of DBaaS clouds.

Varied workloads

Exadata Database Machine is designed to handle mixed database workloads, catering to both compute intensive and IO intensive operations. There is no need to utilize servers of varying capacity and capabilities to reflect the mixture of databases available in the self service catalog.

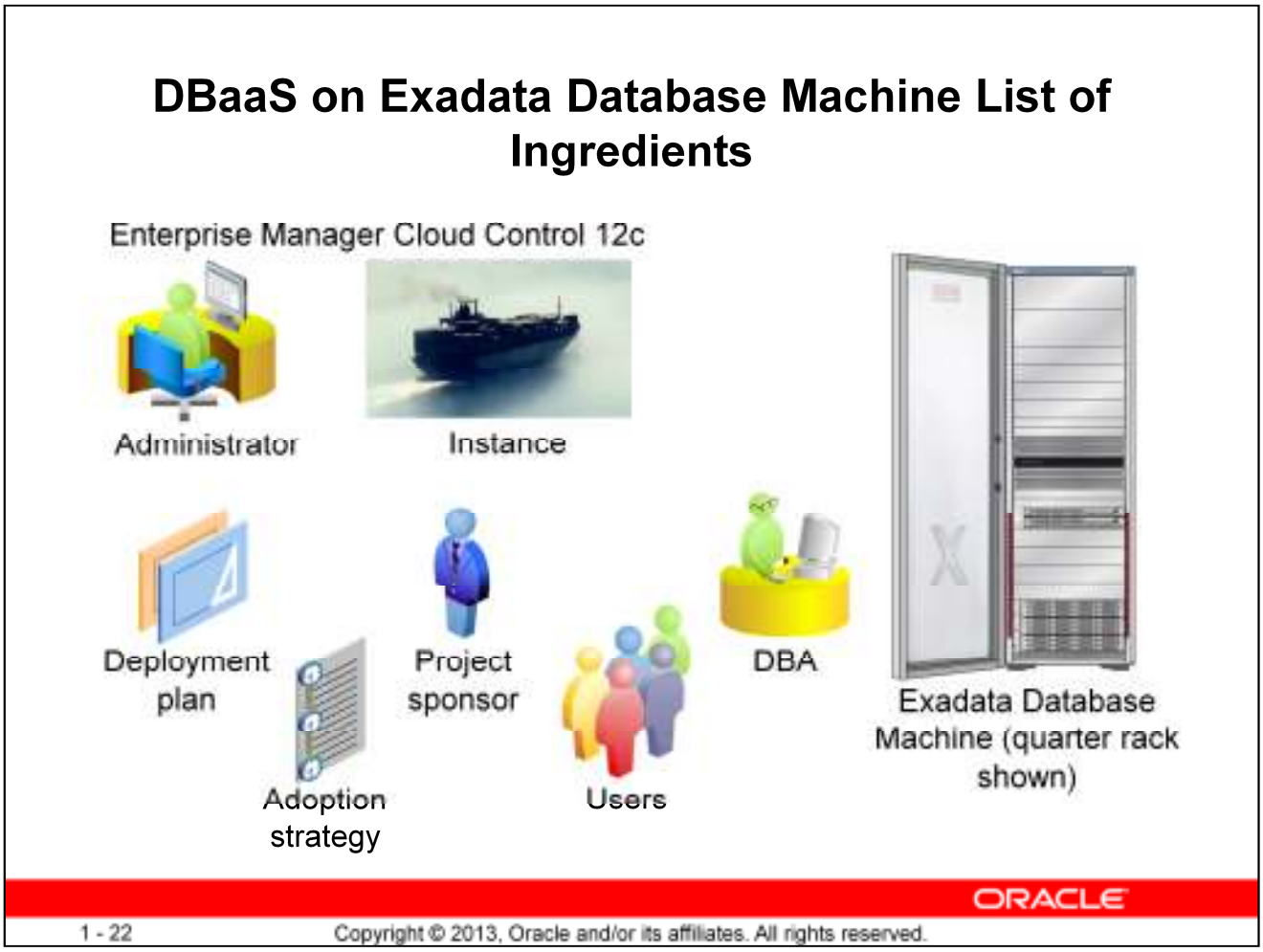
Storage

Because the number and mixture of databases requested by your DBaaS cloud users is not necessarily predictable, neither are your storage requirements. Exadata Storage Servers

provide the capacity with scope for expansion to meet all your end user demands, and, if required, a separate Exadata storage rack can be acquired.

Future friendly

As your DBaaS cloud grows in popularity and utilization, your Exadata Database Machine can potentially expand to accommodate the increased demand for resources.

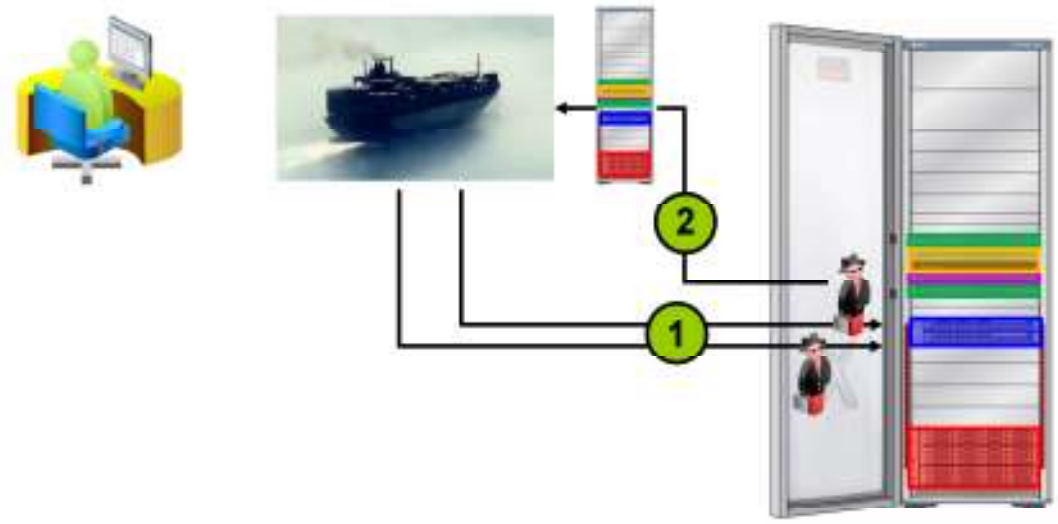


DBaaS On Exadata Database Machine List of Ingredients

To implement DBaaS on an Exadata Database Machine you will need at least the following:

- 1 x Enterprise Manager Cloud Control 12c instance
- 1 x Enterprise Manager Cloud Control 12c administrator
- 1 x Oracle Exadata Database Machine
- 1 x Oracle DBA
- 1 x a reason for implementing DBaaS on Exadata Database Machine
- 1 or more project sponsor
- 1 x deployment plan
- 1 x adoption strategy
- A handful of success criteria
- A handful of initial end users

Recipe Step 1 – Add Exadata Database Machine to Enterprise Manager Cloud Control 12c

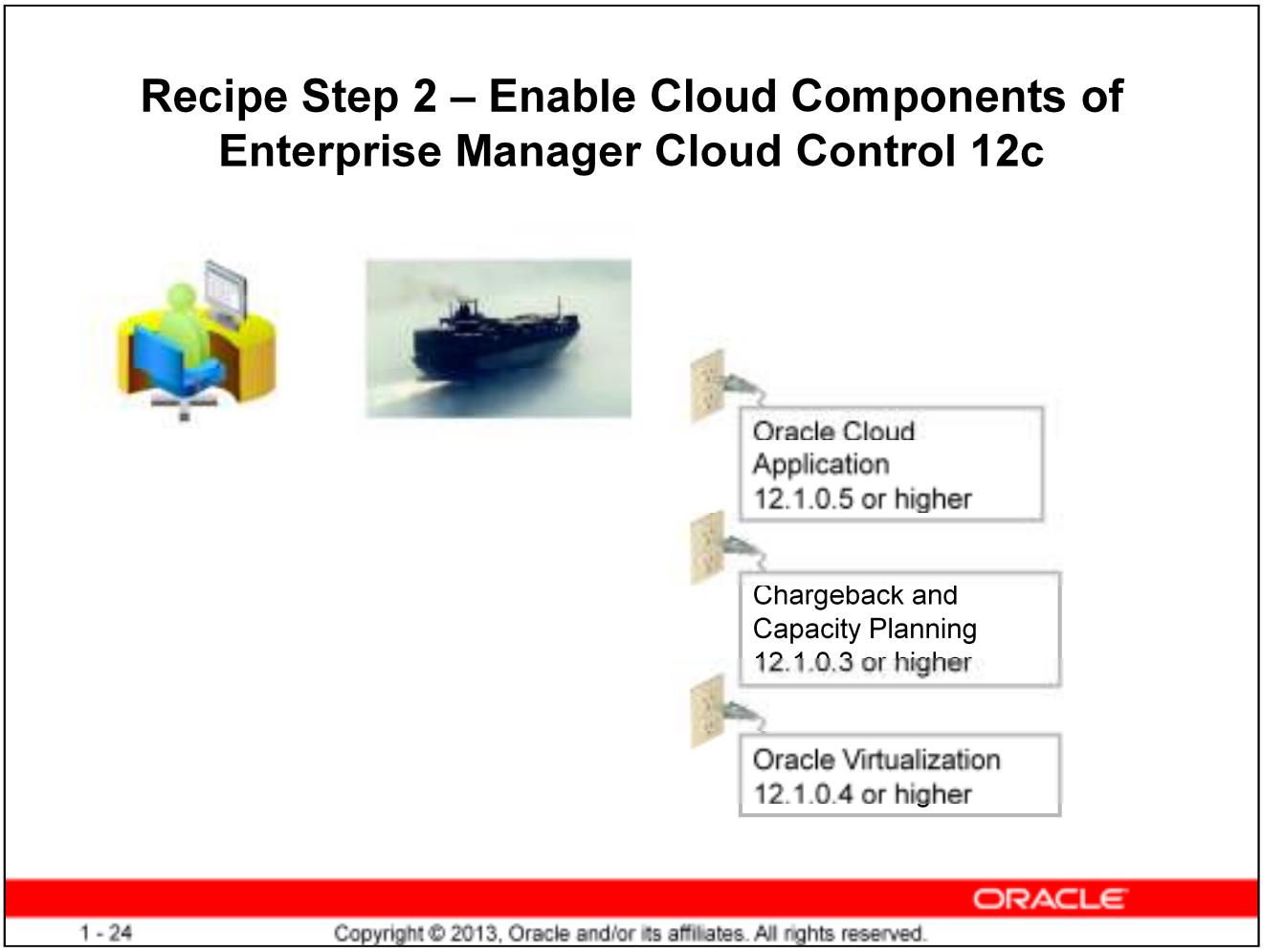


ORACLE

Step 1 – Discover Exadata Database Machine in Enterprise Manager Cloud Control 12c

This activity is comprised of two steps:

- Add the Exadata Database Machine database servers as managed hosts, resulting in the Enterprise Manager Cloud Control 12c agent being installed on the servers
- Use guided discovery with one of the agents to add the Exadata Database Machine and all its components as non-host targets

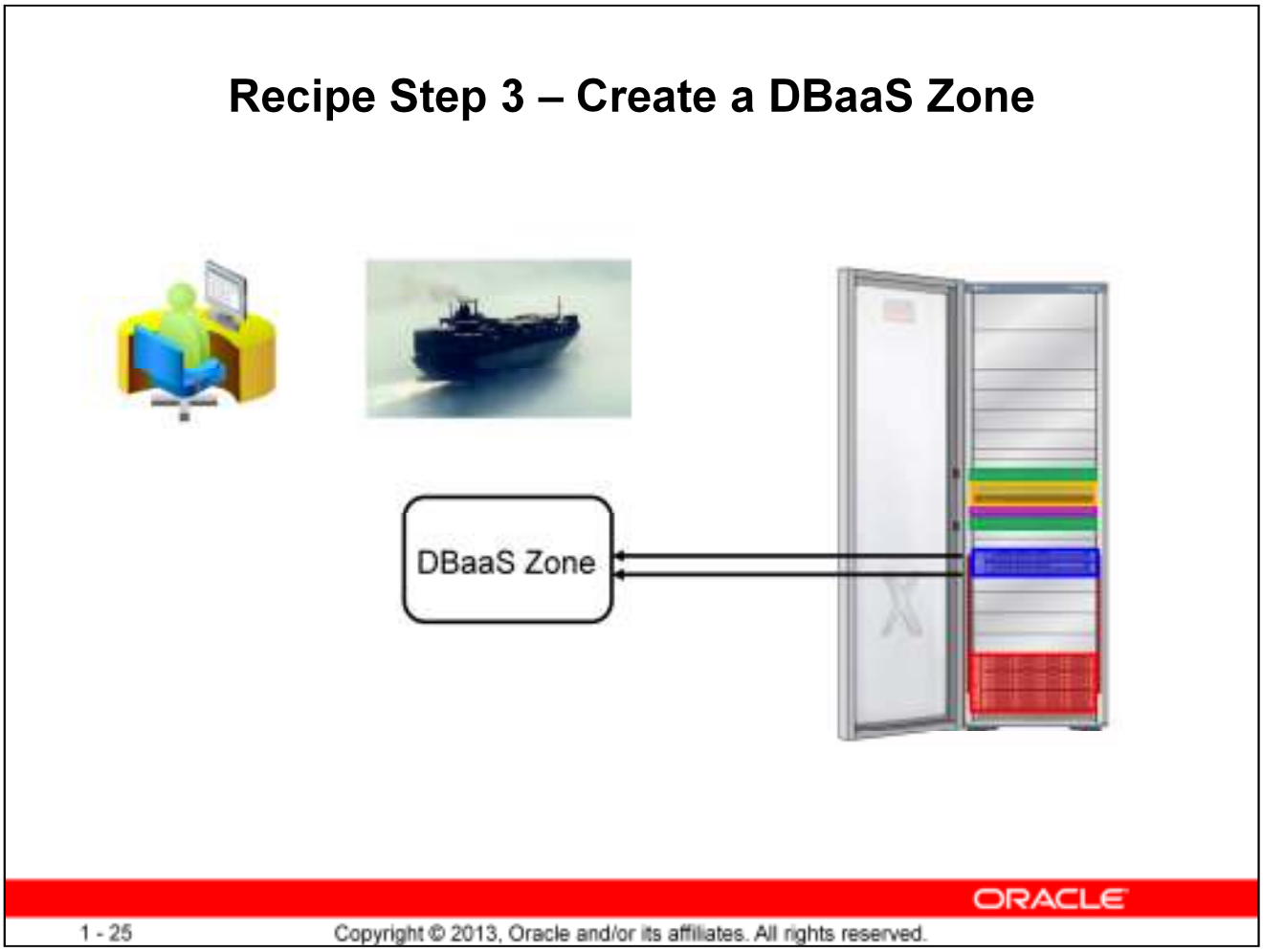


Step 2 – Deploy Cloud Plugins to Enterprise Manager Cloud Control 12c OMS

Deploy the cloud plugins to the Enterprise Manager Cloud Control 12c Oracle Management Service (OMS). These plugins provide all the cloud functionality including the setup and management pages, the self service portal, as well as creating requisite internal users and schema objects.

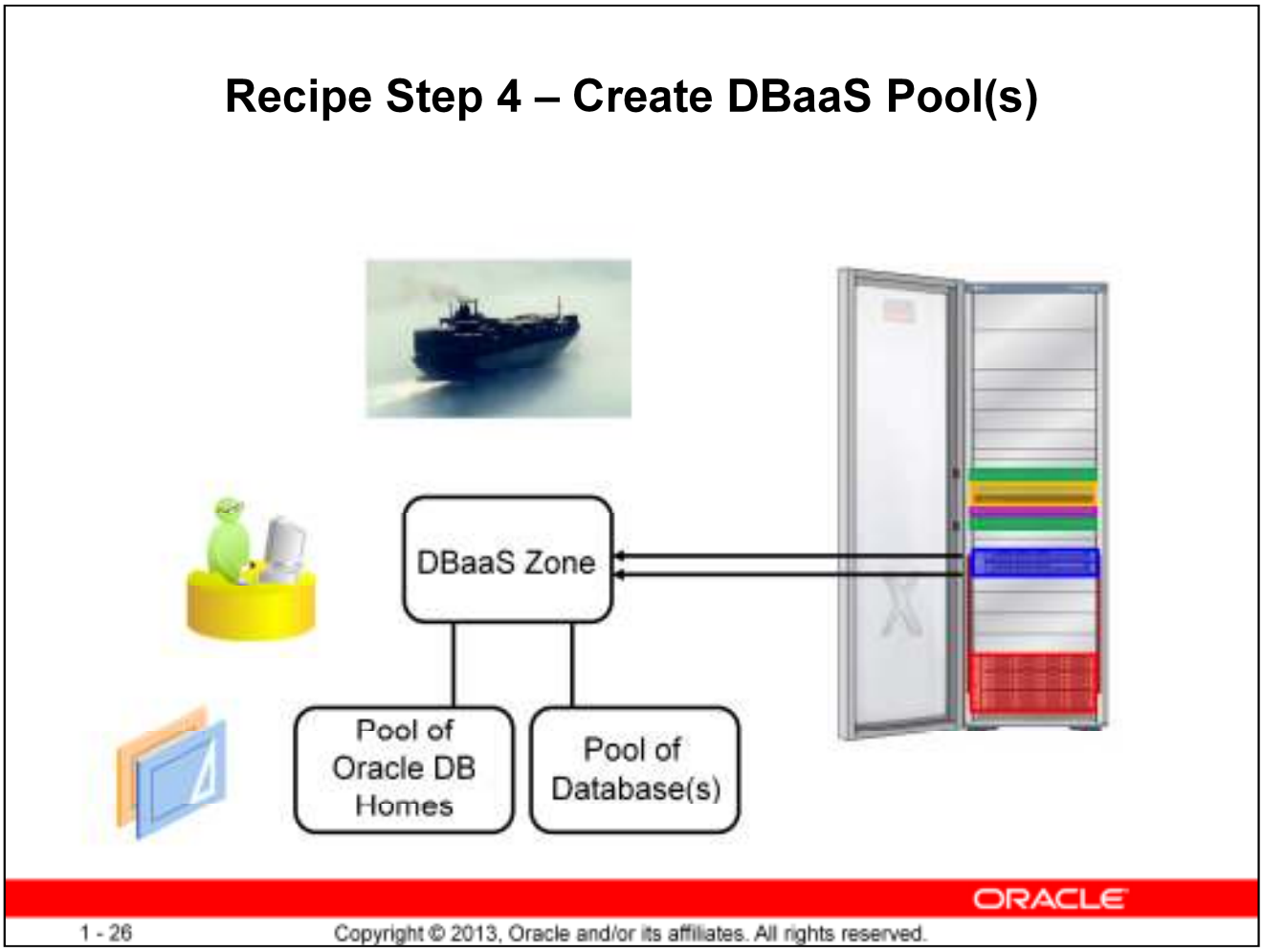
- Oracle Cloud Application – this provides the bulk of the cloud setup and management pages, including the self service portal
- Chargeback and Capacity Planning – this provides the chargeback capability
- Oracle Virtualization – although not strictly required for DBaaS, there are some internal dependencies that require this plugin to be deployed to the OMS

The Oracle Database and Oracle Exadata plugins are deployed by default, and should be at least version 12.1.0.3 and 12.1.0.3 respectively.



Step 3 – Create a DBaaS Zone From the Exadata Database Machine Database Servers

The database servers are added to a zone, from which pools will be created. A zone is a collection of one or more hosts. The DBaaS zone is strictly speaking a Platform-as-a-Service (PaaS) Infrastructure Zone, as DBaaS and Middleware-as-a-Service (MWaaS) which are both variants of the PaaS cloud type are implemented and managed from a common interface, however when implementing DBaaS on Exadata Database Machine the zone will most likely consist exclusively of Exadata Database Machine database servers.



Step 4 – Create Pools of Oracle Homes or Databases

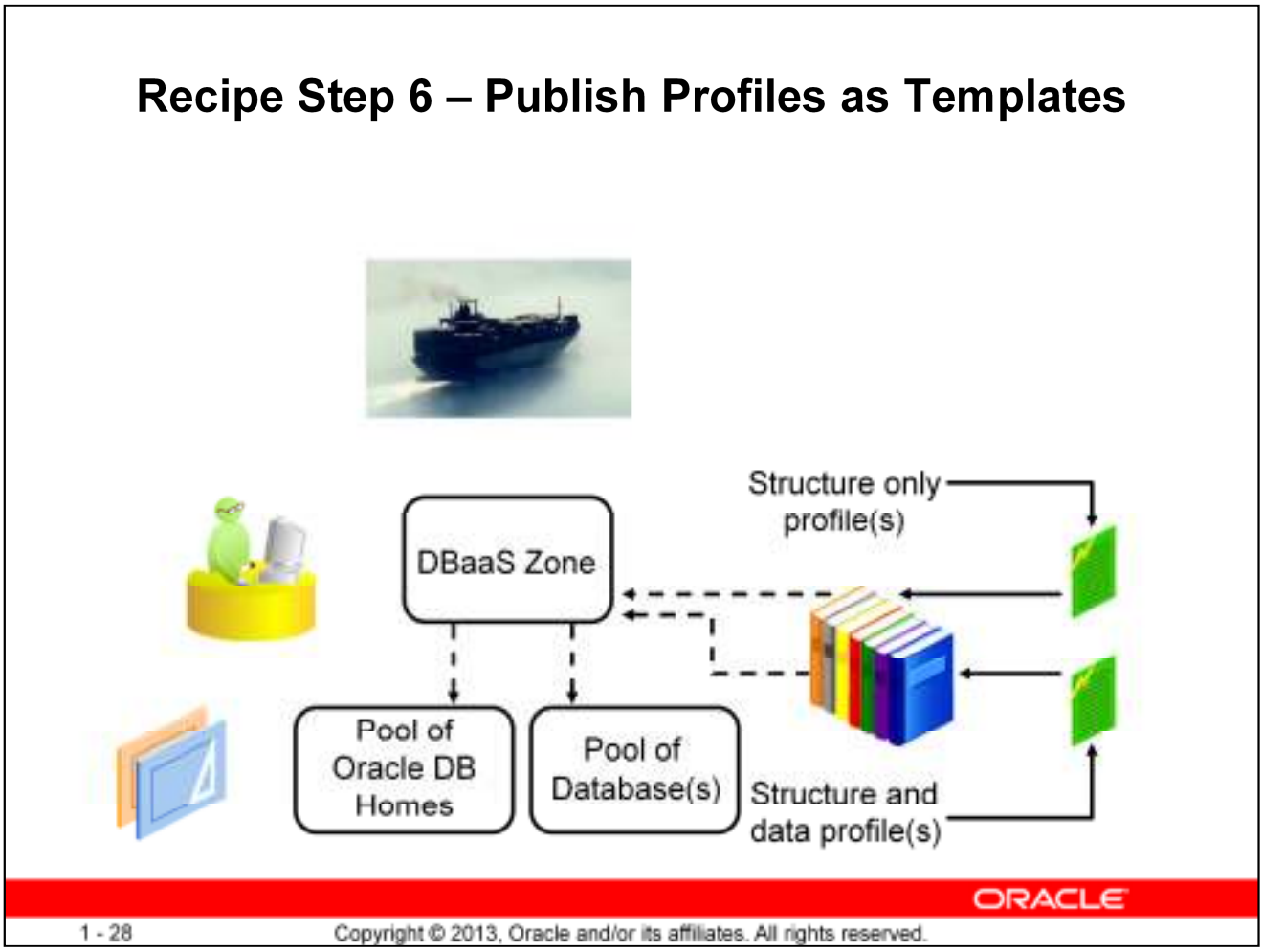
The DBA now steps in and uses the deployment plan to create the required DBaaS pool(s). A DBaaS pool may be a collection of Oracle Database Homes in the infrastructure zone that will be used to service requests for new databases, or it may be a collection of one or more Oracle Databases that are running on the servers in the infrastructure zone that will be used to service requests for new schemas.

Recipe Step 5 – Create DBaaS Profiles

ORACLE

Step 5 – Create Profiles From Reference Databases

Using reference databases, create profiles as designated by your deployment plan. You may only require structure-only profiles or structure-and-data profiles or a combination of the two. The reference databases do not have to be running on the Exadata Database Machine, they simply need to be non-host targets managed by your Enterprise Manager Cloud Control 12c and accessible by the user creating the profiles that match the specifications of the Oracle Homes on the Exadata Database Machine (platform, type, and version) if they are to be used to create full databases.



Step 6 – Publish Profiles as Templates With Configuration Information

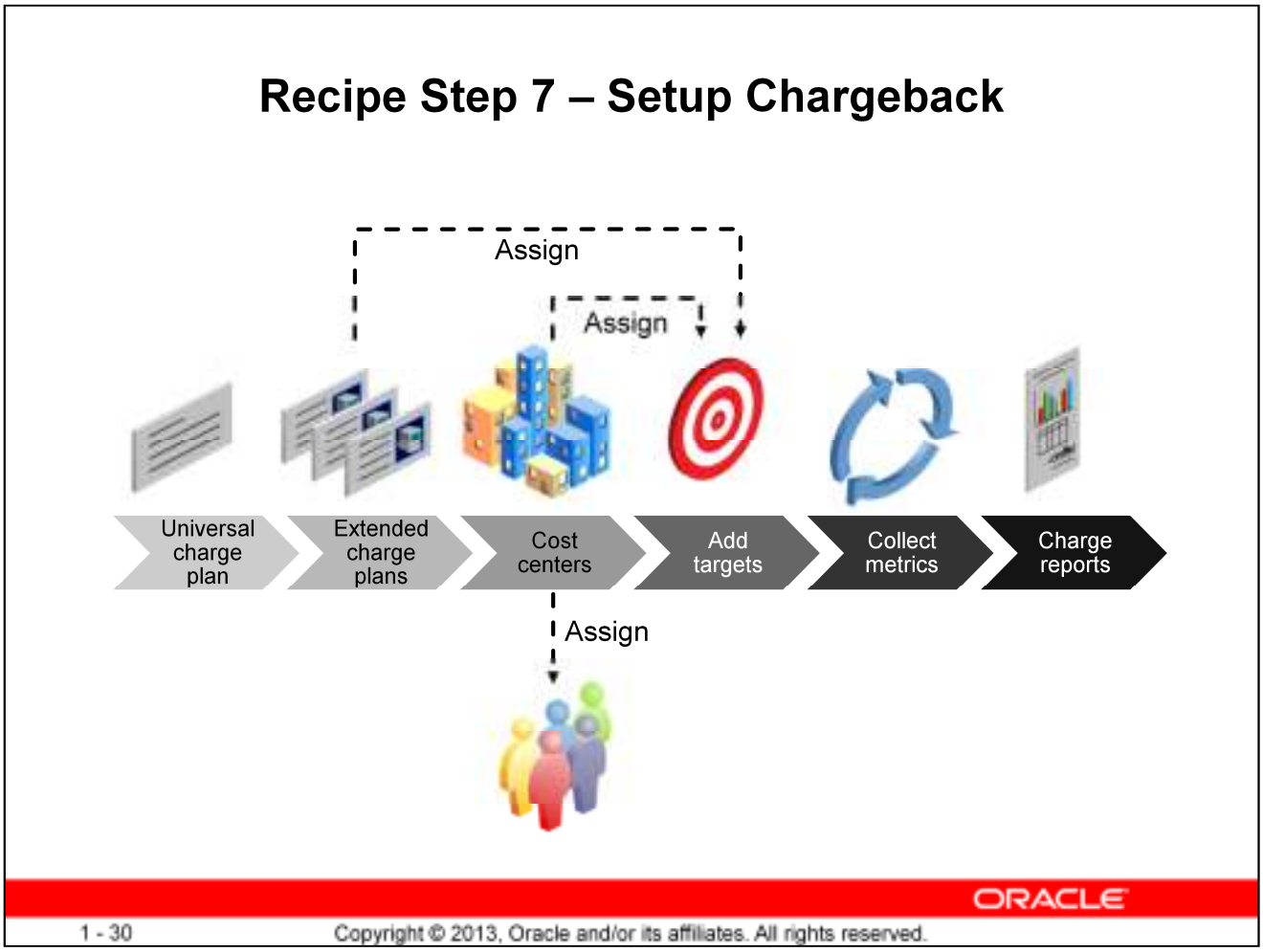
Using the profiles created from your reference database(s), you now publish those profiles as templates, specifying configuration information in the process. This gives you the potential to create many templates from one profile, each with different configuration parameters. In this way you can create a varied template catalog.

Include Database Instance Caging In Your DBaaS Catalog

Database instance caging allows you to cap the amount of CPU resources that a database instance can consume. Multiple instances on the same database server can then consume the optimum amount of CPU available to them. At first glance this simply appears to be the ideal way to squeeze every last drop of performance from a low or mid range serve. However, in the context of an Exadata Database Machine where each database server has multiple cores, instance caging allows you to logically partition your database server to optimally support a mixed workload. When considered as an aspect of a DBaaS service template catalog, instance caging allows DBaaS to place requested databases where they will “best fit”. By creating templates with different values for the cpu_count initialization parameter, you can truly build a varied template catalog that might provide a selection of single or multi instance databases, high or low CPU caps, data or no data, small to large SGA, small to large storage, and so on.

Publish to a Pool in the Zone

When publishing a template, you nominate the DBaaS zone and pool combination where it can be used as well as what self service enabled roles can use it. This allows you to segment your catalog into templates that are suitable for different types of users such as developers, testers, and user acceptance testers.



Step 7 – Setup Chargeback (an Ongoing Task)

You may want to setup chargeback before releasing your DBaaS on Exadata Database Machine cloud to your end users. This slide illustrates the generic steps involved.

- **Set rates for the universal charge plan:** The universal charge plan specifies the base metrics and associated rates that will be used for generating chargeback reports.
- **Optional: define extended charge plans:** Extended charge plans are, as the name implies, an extension of the universal charge plan and allow you to specify fees and rates that should be applied to particular targets over and above the universal charges.
- **Optional: define cost centers:** In the absence of a cost center hierarchy, all charges will be reported against the default cost center. Defining a cost center hierarchy not only simplifies separation of charges by department, but also allows you to easily track resource utilization.
- **Add targets:** Add your DBaaS zone so that all databases and schemas created within it are included in the chargeback collection cycle.
- **Collect metrics:** This automatically occurs without any administrator intervention.
- **Generate charge reports:** An Enterprise Manager job generates reports daily, but you can also generate them on-demand.

Recipe Step 8 – Bake and Serve to End Users

ORACLE

Step 8 – Release to Your Self Service Users and Monitor

Now that your DBaaS catalog is ready, you can release your DBaaS on Exadata Database Machine cloud to your end users. Both DBAs and Enterprise Manager Cloud Control 12c administrators should monitor the activity of your self service users and the performance and utilization of the Exadata Database Machine and make any adjustments.

Summary

In this seminar, you should have learned:

- The steps involved in setting up a DBaaS cloud on an Exadata Database Machine
- Why DBaaS on an Exadata Database Machine simply makes sense
- How you can transform your end users into your biggest supporters