An Expert Guide to Building Oracle Database Cloud Infrastructures

Building Database Clouds in Oracle Database 12*c*

Tariq Farooq | Sridhar Avantsa | Pete Sharman

Building Database Clouds in Oracle 12c

This page intentionally left blank

Building Database Clouds in Oracle 12*c*

Tariq Farooq
Sridhar Avantsa
Pete Sharman

Addison-Wesley

Boston • Columbus • Indianapolis • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2016936586

ISBN-13: 978-0-13-431086-2
ISBN-10: 0-13-431086-1
Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.
First printing, June 2016

Contents

Preface

Cloud Computing is all the rage these days. This book focuses on DBaaS (database-as-a-service) and Real Application Clusters (RAC), one of Oracle's newest cutting-edge technologies within the Cloud Computing world.

Authored by a world-renowned, veteran author team of Oracle ACEs/ACE directors with a proven track record of multiple best-selling books and an active presence in the Oracle speaking circuit, this book is intended to be a blend of real-world, hands-on operations guide and expert handbook for Oracle 12*c* DBaaS within Oracle 12*c* Enterprise Manager (OEM 12*c*) as well as provide a segue into Oracle 12*c* RAC.

Targeted for Oracle DBAs and DMAs, this expert's handbook is intended to serve as a practical, technical, go-to reference for performing administration operations and tasks for building out, managing, monitoring, and administering DB Clouds with the following objectives:

- Practical, technical guide for building Oracle Database Clouds in 12*c*
- Expert, pro-level DBaaS handbook
- Real-world DBaaS hands-on operations guide
- Expert deployment, management, administration, support, and monitoring guide for Oracle DBaaS
- Practical best-practices advice from real-life DBaaS architects/administrators
- Guide to setting up virtualized DB Clouds based on Oracle RAC clusters

In this technical, everyday, hands-on, step-by-step book, the authors aim for an audience of intermediate-level, power, and expert users of Oracle 12*c* DBaaS and RAC.

This book covers the 12*c* version of the Oracle DB software.

Register your copy of *Building Database Clouds in Oracle 12*c for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to informit.com/register and log in or create an account. Enter the product ISBN (9780134310862) and click Submit. Once the process is complete, you will find any available bonus content under "Registered Products."

Acknowledgments

Tariq Farooq

I would like to express boundless thanks for all good things in my life to the Almighty ALLAH, the lord of the worlds, the most gracious, the most merciful.

I dedicate this book to my parents, Mr. and Mrs. Abdullah Farooq; my awesome wife, Ambreen; my wonderful kids, Sumaiya, Hafsa, Fatima, and Muhammad-Talha; and my nephews, Muhammad-Hamza, Muhammad-Saad, Muhammed-Muaz, Abdul-Karim, and Ibrahim, without whose perpetual support, this book would not have come to fruition. My endless gratitude to them as I dedicated more than two years of my spare time to this book, most of which was on airplanes and in late nights and weekends at home.

My heartfelt gratitude to my friends at the Oracle Technology Network (OTN), my colleagues in the Oracle ACE fellowship, my coworkers, and everyone else within the Oracle community and my workplace for standing behind me in my quest to bring this project to completion, particularly Dave Vitalo.

I had been contemplating authoring a book within the Oracle Cloud domain for a few years—the project finally started in 2013. I am very proud of my coauthors' industry credentials and the depth of experience that they brought to this endeavor.

From inception to writing to technical review to production, authoring a book is a lengthy labor of love and at times a painful process; this book would not have been possible without the endless support of the awesome Addison-Wesley team. A very special tip-of-the-hat goes to Greg Doench, the executive editor, and all the other folks at Addison-Wesley. Kudos to the technical reviewers, book reviewers, and editorial teams at Addison-Wesley for an amazing job on this book.

Many appreciative thanks to my buddies, coauthors, and technical reviewers, Pete Sharman, Sridhar Avantsa, Sandesh Rao, and Dr. Bert Scalzo

Finally, I express my gratitude to you, my dear reader, for hopping on this knowledge-laden journey. My sincerest hope is that you will learn from this book and that you will enjoy reading it as much as we did researching and authoring it.

Sridhar Avantsa

This, my second book, is specifically dedicated to my family. They make everything worth it at the end of the day. They encouraged me to take on a second book, in spite of knowing that this would disrupt family time.

To my wife, Gita Avantsa, you are the best thing that has happened to me. You are the pillar of our family and an absolutely amazing mother to our children. Twenty years of marriage and companionship have just flown by.

To my elder son, soon to be 10, Nikhil Avantsa, your physical and mental endurance at such a young age, the ability to see a silver lining in any dark cloud, and the protective shield you place on your younger brother are awe-inspiring. Tarun is indeed lucky to have you as his elder brother.

To my younger son, soon to be 9, Tarun Avantsa, you are the most mature, loving, and caring elementary-school-going child I have ever known and will ever know. Your willingness to sacrifice anything for Nikhil makes you the absolute best younger brother one could ask for.

To my parents, Krishna and Manga Avantsa, your love, sacrifices, and support are the foundations upon which my success is built. To my elder brother, Srinivas Avantsa, you were there to guide me at a most critical and crucial juncture of my life.

I want to thank my coauthors, Pete and Tariq, and the technical reviewers, Bert and Sandesh, for giving me the privilege and opportunity to work alongside some of the finest Oracle technologists to be found. To the amazing supporting team Addison-Wesley, I thank you for countless hours you have invested in the book.

Pete Sharman

Authoring or coauthoring a book is not for the faint-hearted; it's an incredibly intense labor that sometimes makes you wonder why the heck you even volunteered to do it in the first place! So my thanks to the wonderful team at Addison-Wesley, to Sridhar Avantsa, and especially to Tariq Farooq for keeping me on track.

As always, I dedicate this book to my lovely wife, Ann, and my wonderful (now adult) kids, Emma, Kit, and Sandi. Love you all to the moon and back!

About the Authors

Tariq Farooq is an Oracle technologist, architect, and problem-solver and has been working with various Oracle Technologies for more than 24 years in very complex environments at some of the world's largest organizations. Having presented at almost every major Oracle conference and event all over the world, Tariq is an award-winning speaker, community leader/organizer, author, forum contributor, and tech blogger. He is the founding president of the IOUG Virtualization & Cloud Computing Special Interest Group and the BrainSurface social network for the various Oracle Communities. Tariq founded, organized, and chaired various Oracle conferences, including, among others, the OTN Middle East and North Africa (MENA) Tour, VirtaThon (the largest online-only conference for the various Oracle domains), the CloudaThon & RACaThon series of conferences, and the first ever Oracle-centric conference at the Massachusetts Institute of Technology in 2011. He was the founder and anchor/show-host of the *VirtaThon Internet Radio* series program. Tariq is an Oracle RAC Certified Expert and holds a total of 14 professional Oracle Certifications. Having authored more than one hundred articles, whitepapers, and other publications, Tariq is the coauthor of *Expert Oracle RAC 12c* (Apress, 2013), *Oracle Exadata Expert's Handbook* (Addison-Wesley, 2015), and *Oracle Database Problem Solving and Troubleshooting Handbook* (Addison-Wesley, 2016). Tariq has been awarded the Oracle ACE and ACE Director awards.

Sridhar Avantsa started his career with Oracle in 1991 as a developer. Over the years he progressed to become a DBA and an architect. Currently he runs the National Oracle Database Infrastructure Consulting Practice for Rolta AdvizeX (formerly known as TUSC). Sridhar's work as a technologist has been recognized by Oracle with partner awards on multiple occasions. His specific areas of interest and expertise include infrastructure architecture, database performance tuning, high availability/disaster recovery and business continuity planning, Oracle RAC and Clustering, and the Oracle engineering systems. Sridhar has been an active member of the Oracle community as a presenter and as a member of Oracle expert panels at conferences. Sridhar is the coauthor of *Oracle Exadata Expert's Handbook* (Addison-Wesley, 2015). Sridhar lives in Chicago with his wife and two sons.

Pete Sharman is a database architect with the DBaaS team in the Enterprise Manager product suite group at Oracle Corporation. He has worked with Oracle for the past 20 years in a variety of roles from education to consulting to development, and has used Enterprise Manager since its 0.76 beta release. Pete is a member of the Oak Table Network, and has presented at conferences around the world from Oracle Open World (both in Australia and the US), RMOUG Training Days, the Hotsos Conference, Miracle Open World, and AUSOUG and NZOUG conferences. He has coauthored two books, *Expert Enterprise Manager 12*c (Apress, 2013) and *Practical Oracle Database Appliance* (Apress, 2014), and also previously authored a book on how to pass the Oracle8i Database administration exam for the Oracle Certified Professional program. He lives in Canberra, Australia, with his wife and three children.

About the Technical Reviewers and Contributors

Dr. Bert Scalzo is a world-renowned database expert, Oracle ACE, author, chief architect at HGST, and formerly a member of Dell Software's TOAD dev team. With three decades of Oracle database experience to draw on, Bert's webcasts garner high attendance and participation rates. His work history includes time at both Oracle Education and Oracle Consulting. Bert holds several Oracle Masters Certifications and has an extensive academic background that includes a B.S., an M.S., and a Ph.D. in computer science, as well as an M.B.A. and insurance industry designations.

Sandesh Rao is a senior director running the RAC Assurance Team within RAC Development at Oracle Corporation. He specializes in performance tuning, high availability (HA), disaster recovery, and architecting cloud-based solutions using the Oracle Stack. With more than 17 years of experience working in the HA space and having worked on several versions of Oracle with different application stacks, he is a recognized expert in RAC, database internals, and solving Big Data–related problems. Most of his work involves working with customers in the implementation of projects in the financial, retail, scientific, insurance, biotech, and tech spaces. His current position involves running a team that develops best practices for the Oracle Grid Infrastructure 12*c,* including products like RAC (Real

Application Clusters) and Storage (ASM, ACFS), which includes tools like Exachk, OraChk, and the Trace File analyzer framework for Diagnostic collections. Sandesh is involved with several projects for developing, deploying, and orchestrating components in the Oracle Cloud.

1

Database as a Service Concepts—360 Degrees

To understand the basic principles and concepts of database as a service (DBaaS), we must understand the meanings of both *database* and *service* and how the two interact. We also must understand the relationship between cloud computing and DBaaS. Cloud computing encompasses the IT infrastructure resources, which include networks, storage, servers, applications, and services. DBaaS is a subset of the overall cloud concept, specifically focused on the last two resources, applications and services.

The goal of this chapter is to explain the cloud computing implementation as it relates to DBaaS. Although the concept of DBaaS is generic, this book focuses on using Oracle technologies to implement DBaaS.

Cloud Computing: Definition and Classical View

The National Institute of Standards and Terminology (NIST) defines *cloud computing*[1] as follows:

> Cloud computing is 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. [emphasis added]

1. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, p. 2.

This definition articulates the service goals that a cloud computing environment is expected to deliver:

- **Ubiquitous**: The resources are available and ready for consumption.
- **Convenient**: The consumer has easy access to the resources.
- **On demand**: Resource requests need not involve resource approval and acquisition tasks.
- **Shared pool**: The resources are shared, not dedicated, which provides mobility and flexibility in terms of assigning resources.
- **Rapid**: The time window between a resource request and its fulfillment is shortened or eliminated.

These service goals drive the physical implementation of any cloud computing model. Specifically, they provide the basis for the following core aspects of any cloud computing model, which are also interdependent on each other:

- **Roles** applicable within cloud computing—Who is sharing the resource pool?
- **Cloud type** from an infrastructure persepctive—What is the shared resource pool, and how is it deployed?
- The **security** framework within a cloud computing model—What are the basic rules that govern how the resource pool is shared?

Let us first look at roles within a cloud computing model. There are primarily two roles: end user and provider. The *end user* is the entity that uses the hardware resources and associated services that exist in a cloud environment. The *provider* is the entity that owns the physical hardware and infrastructure resources and is responsible for the services associated with delivering these resources to the end user.

Next, we look at the three types of cloud computing models prevalent: private, public, and hybrid.

- **Private cloud**: A cloud infrastructure provisioned for exclusive use by a single organization or entity, maintained by the entity and within the entity's network. The roles of provider and end user are represented by different groups within the entity.
- **Public cloud**: A cloud infrastructure provisioned in the public space such that multiple entities can use the infrastructure simultaneously. The provider is a third-party service provider that supports multiple clients or entities in the end user role.
- **Hybrid cloud**: A combination of the private and public cloud models.

The security model in a cloud environment must include the capability to define roles, responsibilities, and separation of duties for both the provider and end users. As a part of the overall cloud deployment model, the provider must develop, implement, and support a security system with proper access and privileges grants and administration in place.

The security requirements on all cloud environments follow the same basic model, with the difference that in a public cloud, the security controls have to be much wider and more stringent than in a private cloud. The security framework in both the public and private cloud models must also address data security and privacy protection between the provider and end users.

For public clouds, however, security usually requires a much stronger encryption algorithm than used in a private cloud as well as sufficient networking bandwidth to meet public needs. Furthermore, the administrator role at a provider is focused on managing the underpinning infrastructure of the private cloud itself.

To be able to define and create such a security framework, it is important to understand the roles associated with who is using the cloud. Once we understand who is using the cloud, we can translate that knowledge into requirements around access levels, roles, responsibilities, and separation of duties. Following are examples of these requirements:

- The role of a provider's cloud administrator is to manage the underpinning infrastructure of the cloud offering itself.
- The subscriber administrator role is filled individually by each subscriber entity on a cloud. The subscriber administrator manages resources and privileges for his or her own organization.
- The end user role applies to a specific user within a subscriber entity that requests and uses a subset of the resources.

Note

To the provider cloud administrator, the subscriber admnistrator role is the equivalent of an end user role with elevated privileges and rights.

To understand the impact and implementations of the security framework in a cloud environment, we also need to understand the models in which cloud services may be deployed. There are three main models:

- **Software as a service (SaaS)**: SaaS allows the consumer to use the provider's applications running on cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure—not the network, servers, operating systems, storage, or even individual application capabilities—with the possible exception of limited user-specific application configuration settings.

- **Platform as a service (PaaS)**: PaaS allows the consumer to deploy any software or application onto servers deployed on cloud infrastructure. These applications may be consumer-created applications or consumer-acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- **Infrastructure as a service (IaaS)**: The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications and possibly has limited control of select networking components (e.g., host firewalls).

DBaaS—A Special Case of Cloud Computing

DBaaS is a very specific implementation of cloud computing. Now that we have defined cloud computing at a generic high level, let's examine the characteristics that distinguish DBaaS.

To understand the principles of "database as a service," it's helpful to look at the terms *database* and *service* individually and then look at how they interact.

The Merriam-Webster online dictionary provides a definition of *database* that we are all familiar with: "a usually large collection of data organized especially for rapid search and retrieval (as by a computer)."

The term *service* is also one that we are familiar with. Merriam-Webster has quite a few definitions for this word, and following are some of the relevant ones that apply even in the specific case of DBaaS:

- "The occupation or function of serving"
- "The work performed by one that serves"
- "A facility supplying some public demand" such as "telephone service" or "bus service"

But these definitions are generic in nature and context. The core concepts of the definitions still apply in terms of an IT infrastructure as well—but with a few context-sensitive tweaks. In this section, we further explore the meanings and implications of *database* and *service*.

First, just to be clear, the concept we are discussing is not called "Oracle database as a service"—it is just "database as a service." Conceptually speaking, we can deploy DBaaS using Microsoft SQL Server, DB2, PostGres, MySQL, *or* Oracle. They are all software technologies that we can use to build and deploy DBaaS. The design and implementation of DBaaS includes choosing the underlying technologies that we use to implement the service. Oracle is a leader in database technology, and Oracle 12*c* focuses largely on feature sets, utilities, and functionality that enable cloud computing, making Oracle 12*c* a leading contender in terms of implementing DBaaS.

Now let us look at the service element of DBaaS. We must understand what services the end user expects and what components the provider must manage and maintain in order to deliver the expected services.

Let's start with the end users' expectations of DBaaS. Based on NIST's definition of cloud computing, we take the generic expectations implied by "services" and frame them around DBaaS-specific expectations.

Resource Utilization—Usage Instrumentation and Self-Service

One of the fundamental concepts of cloud computing is to provide end users of the cloud service the capability to monitor cloud resource usage and consumption. Therefore, a good cloud service should provide end users visibility into their resource usage, analytics, and chargeback.

With DBaaS specifically, the cloud service should provide the functionality or self-service capabilities to view resource usage and consumption as it applies to databases. The resources would include CPU consumption, storage consumptions, backup service consumption, and network bandwidth consumption.

Broad Network Access

By its definition, a core component of a cloud service is network access, or bandwidth. The cloud service network should be accessible over multiple devices and heterogenous platforms.

From a DBaaS perspective, the focus is on the protocols, quality, and efficiency of network access to the database. Network access concerns are applicable from an application standpoint as well as from an administration and management standpoint.

Resource Pooling

The reason a cloud service provides resource utilization statistics and analytics is to allow end users to tune consumption specifically to their needs. Remember, in a cloud, all the resources come out of a pool. End users should therefore be able to request additional resources when needed and reduce resource allocation and usage as needed. Remember, a cloud service must be able to optimize resources across the entire platform and at the same time maintain performance and availability according to service level agreements (SLAs) between the provider and end user.

Consequently, the service provider must enable end users to provision and decommission resources as needed without having to request resource increases or reductions through the provider. Based on resource usage statistics and business needs, end users should be able to manage and administer all resources, including CPU, network, storage, and backups.

Multitenancy is a key construct of the implementation of any cloud service. The service provider supports multiple clients within a cloud solution. The type of the cloud solution (private, public, or hybrid) determines who the allowed clients (end users) are.

In a DBaaS environment, multitenancy means that the cloud solution supports multiple databases across multiple clients, and each client has one or more databases.

From a DBaaS perspective, the end users, as consumers of services, do not manage the availability of resources or capacity-related issues. These concerns fall under the service provider's responsibility. The service provider must manage the resources at a holistic level across all the clients it supports.

However, there is a caveat here. The preceding statement is true as long as the end user requirements around security, privacy, and compliance are met. The provider has to ensure that the solution design meets the end user criteria.

Rapid Elasticity

Rapid elasticity is the logical next step and evolution of resource pooling. The cloud solution must be able to dynamically allocate or deallocate resources as requested by the end user. The service provided includes the ability to dynamically add or remove resources based on workload volume and nature. In other words, the solution needs to be adaptive and flexible to adjust resource requirements on the fly.

Measured Service

The culmination of all of the preceding concepts of resource usage intrumentation, pooling, and elasticity is the ability to measure resources across an entire service and at each individual cloud subscriber level. Therefore, resource usage should be monitored, controlled, and reported upon, providing transparency for both the provider and the consumers of the utilized service.

Another way to put this is that the cloud solution is based on multitenancy—multiple clients being supported from a single cloud solution, which leads to the question of who should be charged for resources. Obviously, the cost should be based on what resources are actually used. In other words, the cloud solution must be able to support *chargeback* capabilities. The chargeback can be based on what is provisioned or, more popularly, on what resources are actually used.

Services Applicable to DBaaS

We now have a solid foundational understanding of what cloud computing is and how it applies to DBaaS. In this section, we outline the specifics around services as they relate to DBaaS and what they mean to end users as well as to the provider in a cloud computing environment.

The services offered by the DBaaS provider to the end user fall into three main categories: provisioning, administrative, and reporting. Some of these services are optional, and others are mandatory.

Provisioning services provided to end users include some or all of the following:

- The ability to requisition new databases.
- The ability to choose database options as needed (partitioning, advanced security, Real Application Cluster, etc.).
- The ability to add resources (storage, CPU, network bandwidth, etc.) to existing databases. This includes the ability to scale up as well as to scale down.
- Database backup capability using provided backup resources.

Administrative services include some or all of the following:

- The ability to perform on-demand database restores and recoveries
- The ability to perform database clones using existing database backups
- Database monitoring capabilities, including basic 24/7 incident reporting management capabilities

Reporting services include some or all of the following:

- Performance management, which is the ability to look at a database from a performance and tuning standpoint, whether in the form of reporting or in the form of application and GUI database restore and recovery capability

- Resource consumption and usage reports, which let end users compare the resources provisioned and the actual usage so they can fine-tune resource needs to accommodate workload
- The ability to view resource chargeback based on resource allocation and consumption
- The ability to track provider compliance to the SLAs

The ability to track provider compliance to the SLAs is an especially critical point to understand. To ascertain whether or not the requested services are being provided at an appropriate level, end users must first define what "appropriate level" means. For each service, there may be more than one SLA. The higher the SLA, the more technology and resources are needed to satisfy the SLA. Pricing is also affected by the level of service detailed in the SLA.

For example, for I/O performance guarantees, the SLA would specify the input/output operations per second (IOPS) and megabytes per second (MBps) would specify I/O service times. Based on the SLA, the provider determines the actual storage layer provided to the end user. It is the provider's responsibility to ensure that the service delivered to the end user is within the accepted limits.

If we look at I/O performance as an example, the SLAs could be structured as follows:

- **Bronze standard**: Small block average I/O service times equal to or under 15 ms
- **Silver standard**: Small block average I/O service times equal to or under 10 ms
- **Gold standard**: Small block average I/O service times equal to or under 5 ms
- **Platinum standard**: Small block average I/O service times equal to or under 1 ms

Based on these SLAs, the provider may choose to

- Place bronze customers on low-end storage arrays using primarily serial advanced technology attachment (SATA) disks
- Place silver customers on high-performance storage arrays using serial-attached SCSI (SAS) drives
- Place gold customers on high-end storage arrays with a combination of SAS drives and solid-state drives (SSDs)

- Place platinum customers on high-end storage arrays based entirely on SSDs or flash memory

The key is that, once end users make their choice, the provider has to

- Define the exact key performance indicators (KPIs) required to meet the service level expectation
- Ensure that the KPIs required for the SLA are measured and monitored
- Plan for expansion to continue to be able to meet and provide the expected KPI metrics both now and in the long term
- Provide end users with reports that support or, if necessary, justify the provider's service performance capabilities

Architecture of an Oracle-Based DBaaS Implementation

DBaaS started primarily as a consolidation exercise for reducing capital expenditures (CAPEX), but as it evolved, organizations started looking into other key drivers, such as self-service, showback, and chargeback. Before we look at the details of how to implement DBaaS, we need to have some understanding of the underlying consolidation models and deployment issues that are common to all DBaaS flavors and some of the terminology that we use when defining DBaaS.

Consolidation Models

The various consolidation models that can be used to provide DBaaS are shown in Figure 1.1. The simplest and most prevalent form of consolidation exists around server virtualization. Server virtualization offers a simple way of running multiple operating system instances on the same hardware. A better model, platform consolidation, consolidates multiple databases on the same operating system, or a *cluster*. However, in both cases, database sprawl is still an issue that invariably leads to larger administrative overheads and compliance challenges. An even better consolidation model is the capability to host multiple schemas from different tenants within the same database, using Oracle Database 12*c*'s multitenant architecture.

Before we describe such methodologies, however, it is important to have a common understanding of the components that make up the underlying architecture.

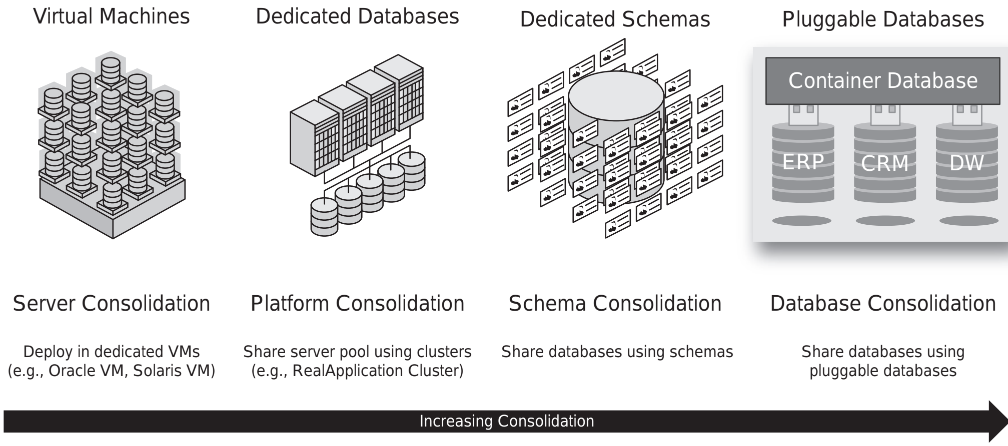

Figure 1.1 Consolidation models

Architecture and Components

In Oracle terminology, hosts containing monitored and managed targets are grouped into *logical pools*. These pools are collections of one or more Oracle database homes (used for database requests) or databases (used for schema requests). A pool contains database homes or databases of the same version and platform—for example, a pool may contain a group of Oracle Database 12.1.0.1 container databases on Linux x86_64.

Pools can in turn be grouped into *zones*. In the DBaaS world, a zone typically comprises a host, an operating system, and an Oracle database. In a similar vein, when defining middleware as a service (MWaaS) zones, a zone consists of a host, an operating system, and an Oracle WebLogic application server. Collectively, these MWaaS and DBaaS zones are called *platform as a service* (PaaS) zones. Users can perform a few administrative tasks at the zone level, including starting and stopping, backup and recovery, and running chargeback reports for the different components making up a PaaS zone.

In the DBaaS view of a PaaS zone, self-service users may request new databases, or else new schemas in an existing database can be created. The databases can be either single instance or a Real Application Cluster (RAC) environment, depending on the zones and service catalog templates that a user can access.

Diagrammatically, these components and their relationships are shown in Figure 1.2.

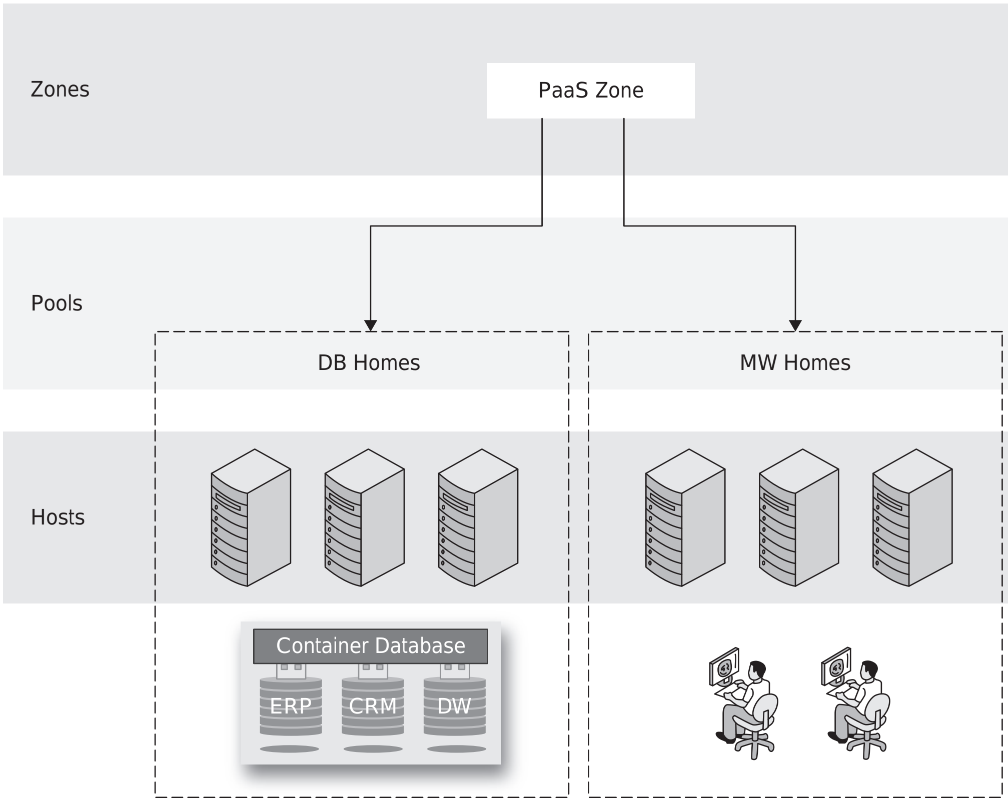

Figure 1.2 Components of a PaaS zone

Deployment Issues

Now that we understand the architecture and components that are used in the different consolidation models, let's examine some standard deployment issues that need to be addressed. These include security, operational, resource, and fault isolation issues, as well as scalability and high availability. It is very important to understand that delivery services and the SLAs around those services will drive the actual architecture, design, and implementation. Therefore, architecture, design, and implementation also play directly into the chargeback and metering aspect of the services.

Security Isolation

Security isolation is often the first point that management worries about in any cloud model. Is my data safe? What options do I have for securing my consolidated infrastructure? How can I prevent the cloud database administrator from accessing and viewing my data? How can I ensure that my network traffic is secure? Can I ensure I meet compliance regulations?

With all of these questions, security isolation has become an essential component of any cloud deployment. Security breaches can arise not only externally but also internally, so all aspects of your cloud infrastructure must be secure.

Operational Isolation

Operational isolation in a DBaaS cloud requires that any maintenance being performed on a database or on the environment the database operates in affects the smallest number of other databases in the same pool. Meeting this requirement clearly becomes more problematic for operating system or grid infrastructure maintenance, though the impact can be minimized by rolling upgrades where allowed. Isolation for patching an Oracle database kernel can be provided by minimizing the number of databases per Oracle home, but adding Oracle homes also increases management overheads. Database startup and shutdown would normally be considered database-dependent operations, but administrative errors such as setting the wrong `ORACLE_SID` can lead to unforeseen impacts on other databases. Again, isolation can be provided at the `ORACLE_HOME` level and by having different user IDs and group IDs at the kernel level, but this also leads to more management overhead, and, it must be said, more likelihood of human error.

Resource Isolation

In a DBaaS cloud, resource isolation deals with the allocation and segregation of resources such as CPU, memory, network (public and private), and storage (I/O per second and overall capacity). Management concerns include questions such as How does the CPU usage of my database affect other databases in the DBaaS cloud? How much memory should I allocate to a specific database? Can I restrict the network utilization, both at the public network and interconnect levels, to not impact other databases? Likewise, how can I guarantee storage capacity and IOPS for my databases?

Fault Isolation

Fault isolation in a DBaaS cloud is normally provided at the database level, since that is the unit of granularity in the multitenant architecture. Each database and its associated instance (or instances, in RAC environments) need to be isolated from other databases. Even when all databases are run from a single `ORACLE_HOME`, database faults are normally isolated to a failing instance, so fault isolation is maintained by fencing off the offending instance. However, other failures may require handling at different levels. For example, concerns include how to deal with a server, network, or storage failure. Such failures are normally handled by some form of redundancy such as multinode setups, active/passive switches, bonded networks, or redundant storage such as Automatic Storage Management (ASM) redundancy.

DBaaS Scalability

Scalability is a fundamental characteristic of DBaaS architectures by virtue of their support for self-service, elasticity, and multitenancy. Oracle's database technologies provide a number of ways to support scalability when delivering database services, including resource management and quality of service, addition of extra storage through functionality such as multiple Exadata Database Machine frames, horizontal scaling via RACs when service demands increase beyond the capabilities of a single machine, and scalable management resources where Oracle Enterprise Manager can add management nodes as the number of targets under management grows.

DBaaS High Availability

Not all consumers require the same level of availability in a cloud environment. Oracle's DBaaS self-service catalog allows the capability to include different levels of availability using a metals model, as shown in Table 1.1.

For example, the bronze standard provides a single-instance database service (possibly via RAC One Node), whereas the other extreme, platinum, would normally include a RAC database with multiple standbys. These standbys might include a near standby in the same data center as your RAC database and a far standby in a completely separate remote data center. These measures help to improve the high-availability and disaster recovery goals you have for that database. In Oracle Enterprise Manager 12.1.0.4, with the added support for Data Guard, you now have the ability with just a few clicks to provision the primary and multiple standbys across different data centers. The standbys can be either single instance or a RAC configuration.

Table 1.1 Availability Levels

Service Level	Description
Bronze standard	Single instance databases/RAC One Node databases
Silver standard	Single instance with standby
Gold standard (HA)	RAC databases
Platinum standard	RAC with standby

Business and Technology Benefits of Having DBaaS Enabled

DBaaS, or a database cloud, is becoming a very popular concept with organizations of all sizes across the spectrum of industry. Placing database infrastructure concerns with the DBaaS provider frees an organization's IT and technology departments to focus at an organizational level rather than at an application or department level. With the focus at an organizational level, the IT and technology teams are more closely aligned with the organizational and business needs. The fundamental requirements of an organization have never really changed—they have always aimed for lowered operational expenses (OPEX) and total cost of ownership (TCO). What has changed is the emergence of new platform architecture and software technologies that, working together, deliver on those needs. The opportunity to reduce OPEX and TCO is precisely what is driving the improved acceptance and adoption rate of DBaaS.

Let's look at some of the intrinsic benefits of deploying DBaaS, which include the basic benefits associated with any cloud solution:

- **Time-to-market**: The nimbleness with which a company reacts and adapts to changing market conditions, competition, and consumer needs and expectations is critical. A core component of any cloud solution is self-service and automation. With a well-planned cloud solution, there is no need to deploy hardware for new projects, and with self-service and automation, the business units become more self-reliant.
- **Scalability**: The combination of inherent concepts of elasticity, consolidation, and resource pooling at a wider organizational level drives scalability in a cloud computing environment. For custom built solutions, the value and benefit of this automatic scaling is even more potent and impressive.
- **Empowerment**: Cloud computing solutions typically have a web-based interface for users. They can be accessed by employees, customers, and partners no matter where they are. With a cloud database, everyone gets to work with the same set of information, and spreadsheet chaos is a thing of the past.
- **Availability**: Combining the benefits of standardization (hardware, software, procedural best practices) and empowerment (self-service, on demand scalability) automatically delivers improved availability.

Let's go a step further and look at why databases are worthy of their own class in the cloud solution world. We do not see phrases such as "application servers as a cloud" or "web servers as a service" or "exchange servers as a service." Logically and technically, these concepts can exist, but they do not. Why is that?

Databases are used to store data. As we are all aware, the amount of data being generated, used, and stored is growing exponentially. This ever-growing volume of data needs to mined and analyzed to generate intelligent, actionable information. Now, more than ever, data means everything—it drives financial, operational, and tactical decisions and strategies in every business. But along with all this data come the headaches of tasks such as managing performance, scaling capacity, and backup and recovery strategies.

Databases are often considered the single point of serialization of application processing and logic, usually because application design is not focused on how databases work or the best way to use them. What this means is that designing, managing, and performance tuning databases represents a unique set of skills and talents.

From a computing perspective, resource consumption characteristics and performance needs of a database are unique in nature. Databases, especially untuned databases, can be resource hogs when it comes to storage, CPU, and network resources.

Scaling of databases also presents unique challenges. Scaling can directly impact expenditure on multiple components of the platform and infrastructure, including on the storage subsystem (due to storage volume or performance) and on throughput (in IOPS or MBps).

Databases are a complex component of the application stack. Consequently, the underlying database technology can potentially have a severe impact, positive or negative, on the overall scalability, availability, business continuity, and performance aspects of any given application. When the applications in question are business critical and/or revenue generating, the potential for impact makes the databases a very visible, highly scrutinized component.

From an economics perspective, databases can prove to be one of the costliest, if not *the* costliest, component of any given application deployment. The database's application stack, for example, can drive the overall solution cost in the following ways:

- Database licensing costs and annual support costs.
- Database-specific infrastructure costs, especially those driven by performance initiatives, such as high-performance compute servers, high-performance storage, and in some cases even high-performance networking.
- Staffing and resourcing costs for maintaining the database (design, administration, performance tuning, etc.).
- Cost of high-performance backup management, storage systems, and infrastructure based on the uptime, recovery point objectives (RPOs), and RTO expectations. (In today's age of data explosion, databases tend be quite large, and database backup and recovery becomes key.)

Cloud computing, as a concept and a solution, is aimed at resolving these economic concerns. When you add the uniqueness of databases to the mix, you can see the value of deploying a database cloud, or in other words, deploying a DBaaS solution.

Great First Step for Transitioning into the Cloud

Moving an organization's IT infrastructure from the old server-based model to a cloud-based model can be a daunting task, regardless of whether the destination is a private cloud or a public cloud. Implementing a cloud solution on a very focused, self-contained technology stack, such as database technology, can be a very useful first step into cloud computing.

The toolkit available for database technologies is wide, extensive, mature, and multivendor. The same is true for the infrastructure components, such as the server, storage, and backup infrastructures. Dedicated, fully contained, engineered appliances have been a part of the database technology stack for a while now.

Another key aspect to consider is the significant amount of automation that exists in the database arena. This is primarily due to the unique and complex nature of databases plus the sizes of the databases that are common nowadays.

Security is an important aspect of any cloud solution and is yet another consideration that has long been a part of any overall database solution. Databases have their own dedicated security model that is mature and can fairly easily integrate into the larger organization model (single sign-on [SSO]- and Lightweight Directory Access Protocol [LDAP]-based authentication and integration, etc.). Database security models have matured to include data encryption for data backups, data at rest, as well as data in flight.

Finally, the amount of data existing within organizations is huge, and its rate of growth is exponential. Almost every application deployed will need a data repository or data store of some type. This growth in data must be supported by corresponding growth in infrastructure.

The combination of the mature toolkit, the engineering inherent to database solutions, the preexisting automation especially in the administration aspects of databases, and the existence of a mature security model provide a solid foundation upon which organizations can build and deploy their first cloud solution.

The existing domain knowledge and the highly experienced skill set available provide the technical basis for learning and fine tuning the various aspects of cloud computing.

According to some reports and surveys, database technology–related expenditure for midsize to large-size companies can be up to 40 percent or more of the annual IT budget. Having a defined organization-wide strategy for databases will help organizations manage the growth of data and at the same time keep database costs down. The fact that databases can drive up to 40 percent of the IT budget makes the database a very attractive focus area to use to kick off cloud computing as a long-term IT strategy.

Summary

Cloud computing is a generic architectural concept that encompasses the entire gamut of technology as it relates to infrastructure. Cloud computing is more than just another fancy term for "virtualization." All of the new "as a service" models are implementations of cloud computing. Infrastructure, platform, database, software, network, and storage as a service all are implementations focusing on specific concepts of the technology stack within infrastructure. These terms are sometimes used interchangeably, but in reality, cloud computing is a concept, whereas the as-a-service models are implementations.

The very definition of "cloud computing" has introduced a fundamental change in thinking when it comes to ownership, roles, responsibilities, and expectations. This is not to say that ownership, roles, responsibilities, and expectations were missing or lacking before the advent of cloud computing. They have always existed, but cloud computing has changed the lens through which they are seen.

Introducing the core concept of "service" into the overall architecture brings about these changes. We saw that in order to deliver a service that is meaningful, cloud computing had to introduce "elasticity, flexibility, and rapid and easy deployment" into its core concept and architecture.

DBaaS implementations are not much different from other cloud implementations. Database clouds have some unique challenges when it comes to cloud implementations, driven by their complex and temperamental nature. We need to understand these core concepts specifically as they apply to databases in order to deploy a successful and meaningful DBaaS.

This chapter is the beginning of understanding the cloud computing framework, specifically when it comes to database clouds or DBaaS.

This page intentionally left blank

Index

V

W

X

Z