



The Cloud DBA-Oracle

Managing Oracle Database in the Cloud

First Edition

Abhinivesh Jain
Niraj Mahajan



Apress®

The Cloud DBA-Oracle

Managing Oracle
Database in the Cloud

First Edition



**Abhinivesh Jain
Niraj Mahajan**

Apress®

The Cloud DBA-Oracle: Managing Oracle Database in the Cloud

Abhinivesh Jain
Pune, Maharashtra, India

ISBN-13 (pbk): 978-1-4842-2634-6
DOI 10.1007/978-1-4842-2635-3

Niraj Mahajan
Pune, Maharashtra, India

ISBN-13 (electronic): 978-1-4842-2635-3

Library of Congress Control Number: 2017933318

Copyright © 2017 by Abhinivesh Jain and Niraj Mahajan

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Celestin Suresh John
Development Editor: Laura Berendson
Technical Reviewer: Sudip Das
Coordinating Editor: Sanchita Mandal
Copy Editor: Kezia Endsley
Composer: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover image designed by Freepik

Distributed to the book trade worldwide by Springer Science+Business Media New York,
233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail
orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC
and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc).
SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-2634-6.
For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my parents, my wife Bhumika, and my son Aarjav,
for their unconditional love, support, and encouragement in writing this book.*

—Abhinivesh Jain

*To my parents, my wife Aboli, and my daughters Swarali and Sarisha,
for their continuous support and love.
My father is my inspiration for writing this book.*

—Niraj Mahajan

Contents at a Glance

About the Authors.....	xiii
About the Technical Reviewer	xv
Acknowledgements	xvii
Introduction	xix
■ Part I: Cloud Computing Fundamentals	1
■ Chapter 1: Introduction to Cloud Computing	3
■ Chapter 2: Introduction to Database as a Service	11
■ Part II: Database Administration in Cloud	23
■ Chapter 3: Provisioning	25
■ Chapter 4: High Availability Options.....	51
■ Chapter 5: Disaster Recovery Options	71
■ Chapter 6: DB Security	87
■ Chapter 7: DB Migration to the Cloud	99
■ Chapter 8: Backup and Restore.....	139
■ Chapter 9: Manage and Monitor	173
Epilogue.....	211
Index.....	213

Contents

About the Authors.....	xiii
About the Technical Reviewer	xv
Acknowledgements	xvii
Introduction	xix
■ Part I: Cloud Computing Fundamentals	1
■ Chapter 1: Introduction to Cloud Computing	3
Overview	3
What Is Cloud Computing	3
Benefits of Cloud Computing	4
Challenges of Cloud Computing.....	4
Cloud Computing Service Models	5
Infrastructure as a Service (IaaS).....	6
Platform as a Service (PaaS).....	6
Software as a Service (SaaS)	6
Cloud Deployment Models.....	6
Public Cloud.....	6
Private Cloud	8
Hybrid Cloud	8
Community Cloud	9
Metering and Chargeback	10
Summary.....	10

■ Chapter 2: Introduction to Database as a Service	11
What Is Database as a Service?	11
DBaaS Public Cloud Offerings	12
Oracle Public Cloud Offering.....	12
AWS Public Cloud Offering.....	16
Microsoft Azure Public Cloud Offering.....	18
DBaaS in Private Cloud.....	19
DBaaS in Managed Services Model	19
Getting Started with Database as a Service.....	20
Selecting Your DBaaS Vendor	20
Selecting the DBaaS Services	20
Choosing the Right DBaaS Subscription.....	20
Getting Acquainted with the Basic Terminology in Database as a Service.....	21
Oracle Cloud Terminology	21
Amazon Web Services Terminology	21
Microsoft Azure Terminology	22
Summary	22
■ Part II: Database Administration in Cloud	23
■ Chapter 3: Provisioning	25
Database Provisioning Overview	25
Database Provisioning in Oracle Cloud.....	26
Creating Oracle Cloud Subscription	26
Selecting Oracle Database Cloud Offering.....	29
Creating the Service	30
Accessing Your Database	36

Database Provisioning in AWS.....	39
Selecting the Proper Engine	39
Specifying the DB Environment.....	41
Specifying the DB Details	41
Configuring the Advanced Settings	44
Accessing Your AWS RDS Instance.....	47
Summary.....	50
■Chapter 4: High Availability Options.....	51
Need for High Availability with Cloud-Based DBs.....	51
Database High Availability Overview	51
HA Options in Oracle Cloud	53
Oracle RAC Provisioning in Oracle Cloud.....	53
Connecting to Your RAC Database Servers Using PuTTY.....	59
Connecting to Your RAC Database Using SQL Developer.....	60
RAC HA Testing.....	64
HA Options in AWS.....	64
Overview of Oracle RAC Provisioning in AWS	64
Overview of the Multi-AZ Deployment Option	65
Multi-AZ Deployment Method.....	66
Failover in Multi-AZ Deployment	69
Summary.....	70
■Chapter 5: Disaster Recovery Options.....	71
Need for Disaster Recovery with Cloud-Based DBs	71
Database Disaster Recovery (DR) Overview.....	71
DR Options in Oracle Cloud	72
Provisioning of DataGuard in Oracle Cloud.....	73

DataGuard Administration in Oracle Cloud	75
Checking the DG Configuration Details.....	76
Performing DB Switchover Using the Console.....	76
Performing DB Switchover Using dbaascli	78
Performing DB Failover Using the Console.....	82
Reinstating Failed Primary Database	83
DR Options in AWS.....	85
Summary.....	85
■ Chapter 6: DB Security	87
Need for Database Security with Cloud-Based DB.....	87
Cloud Security Model	87
Security Configurations in Oracle Cloud.....	88
Transparent Data Encryption (TDE).....	89
SQL*Net Encryption	90
Security Configurations in AWS	90
Security at the Account Level	91
Security for Oracle Running on Amazon RDS	93
Security for Oracle Running on Amazon EC2.....	95
Security Best Practices	98
Summary.....	98
■ Chapter 7: DB Migration to the Cloud	99
DB Migration Key Considerations.....	99
Migration Lifecycle	101
Migration Approach	102
Backup.....	102
Transfer.....	102
Restore	103

Migration Options	103
Migration Using Tools	103
Traditional Migration Methods	105
Lesser Known Migration Methods	106
Comparison of Migration Methods	107
Detailed Migration Steps for Data Migration from the Local Database to Amazon RDS	107
Detailed Migration Steps for Data Migration Using Oracle Data Pump Utility	113
Sample Comparison Between SQL Developer and Data Pump Migration.....	116
Migration Using RMAN Backup Restore	117
Migration By Cloning a PDB into the Cloud	118
Multitenant Setup On-Premise (Local).....	119
Create Oracle 12c PDB Database in Oracle Public Cloud	120
Clone LocalPDB (Optional)	120
Open the Cloned LocalPDB in READWRITE Mode (Optional)	122
Close LocalPDB.....	123
Convert LocalPDB into READONLY Mode	123
Unplug LocalPDB	124
Copy the .XML and Data Files to the Cloud DB Server.....	124
Using EM Express to Plug LocalPDB into the Cloud.....	129
Open LocalPDB in the Cloud	131
Migration Using DMS.....	132
Migration Using AWS Snowball	135
Migration Best Practices	137
Summary.....	137

Chapter 8: Backup and Restore	139
Overview of Database Backup and Restore from the Cloud Perspective	139
Using the Cloud as a Backup Destination for an On-Premise Database.....	140
Using Backup and Restore Options (Within the Cloud) for a Cloud Database	141
DB Backup and Restore in Oracle Cloud	141
Oracle Database Cloud Backup Service	142
DB Snapshots	147
Oracle Database Backup Service	149
Database Recovery in Oracle Cloud	155
DB Backup and Restore in AWS.....	162
On-Premise Backup to Amazon Cloud Using Amazon S3	162
DB Backup and Restore in Amazon RDS.....	167
DB Backup and Restore in Amazon EC2	172
Backup and Restore Best Practices	172
Summary	172
Chapter 9: Manage and Monitor	173
Overview of Cloud DB Monitoring and Management.....	173
DB Monitoring and Management in Oracle Cloud	174
DBaaS Monitor.....	176
EM Express	185
Command-Line Interfaces	194
SQL Developer	197

DB Monitoring and Management in AWS.....	199
AWS RDS Monitoring Using OEM.....	201
AWS CloudWatch	204
DB Monitoring and Management Best Practices.....	209
Summary.....	209
Epilogue.....	211
Index.....	213

About the Authors



Abhinivesh Jain is Oracle ACE Associate, AIOUG (All India Oracle User Group) Evangelist, and founder member of AIOUG, Pune Chapter.

He has more than a decade of database experience with expertise in datacenter migration, transformation, database migration, upgrade, performance tuning, solution designing, DBaaS, and Exadata.

He is a frequent presenter at Sangam, OTN Yathra, AIOUG Pune Chapter, Tech Days, and other Oracle events. He is an active member of Oracle community and one of the leading members of the Exadata community.



Niraj Mahajan has more than 15 years of experience in managing Oracle databases. He is a founding member of AIOUG, Pune Chapter, and a frequent speaker at the AIOUG annual conference, Sangam. His core skills are Oracle licensing, DB cloud adoption, solution architecture and tuning, database design and administration, OEM12c, etc.

He is Oracle Certified Professional DBA (OCP) for Oracle 8i, 9i, 10g, 11g, and OCE 10g RAC, Oracle 12c implementation specialist, and Oracle Exadata implementation specialist. He is also certified for SQL Server, VMware, and ITIL.

About the Technical Reviewer



Sudip Das has over 21 years of diverse, all round experience in IT Infrastructure strategy, Cloud solutions, solution architecture, consulting, delivery, automation, operations and innovation in large projects of global organizations specializing in database administration. Won multiple large deals over 100MUSD. Over six years of international experience with global organizations and customers in Germany, United States, South Africa and Europe. Rich experience in developing and implementing service line strategy. Adept at engaging with customers and partners to provide consultative approach towards building proactive solutions addressing client requirements.

Acknowledgments

Converting an idea into a book is a long journey and this can't be done without the help of so many individuals.

First of all, thanks to my co-author Niraj, without whom I could not have completed this book. Besides writing four chapters, he was instrumental in solving many show-stopper issues and keeping the momentum going.

Thanks to our technical reviewer, Sudip, whose expert comments helped in improving the quality of this book.

Special thanks to our friends, Harichand Kapgate and Luv Ranadive, who constantly inspired us and motivated us to finish this book.

Thanks to all our family members for their support and encouragement.

Last but not the least, thanks to all our readers!!!

Introduction

Cloud adoption in the database world is growing very rapidly. Various research and studies forecast the DBaaS and cloud database service market to grow at a compound annual growth rate (CAGR) of more than 65% by 2019.

Currently available books cover either cloud computing or database administration, but not both. This book bridges the gap. The book is divided into two parts. Part I covers the cloud computing concepts and database as a service overview, whereas Part II covers everything that is required for you to become a cloud DBA.

Part II starts from your first database provisioning in the cloud and then moves to setting up Oracle RAC and DataGuard in the cloud. The chapter on database migration to the cloud gives you in-depth details around how to plan and execute DB migration. The chapter on DB security touches on the key security aspects that you should take care for your cloud-based database. The backup and recovery chapter covers various backup and recovery options in Oracle cloud and AWS. The manage and monitor chapter covers details on all the tools that are useful for performing day-to-day monitoring and administration. Part II covers DB administration aspects from the two most prominent cloud providers—Oracle Cloud and Amazon Web Services (AWS).

This book has the right balance of theory and practical examples, along with the best practices in each given area.

Database administrators, DB architects, and DB operations managers can use this book to learn and understand the process of running Oracle database in the cloud.

Thanks for purchasing this book; we are confident that the knowledge you acquire by reading this book will be worth your time and money. For any comments, suggestions, or feedback, write to us at feedback@theclouddba-oracle.com.

PART I



Cloud Computing Fundamentals

To become Cloud DBA, you need to first understand the cloud concepts. Cloud computing is very vast area and there are numerous books available in market. Objective of Part I of this book is to give you “just enough” details of cloud computing so that you can get started towards your journey to become The Cloud DBA.

In Part I, there are 2 chapters.

Chapter 1 covers the overview of cloud computing. This is explained using cloud computing definition, benefits, challenges, service models, deployment models etc.

Chapter 2 covers overview of Database as a service (DBaaS) and how DBaaS is provided in IaaS, PaaS, Public cloud and Private cloud. It also makes you aware of key terminologies related to DBaaS.

CHAPTER 1



Introduction to Cloud Computing

This chapter explains the definition and benefits of cloud computing. It also covers cloud computing deployment models and service models. Finally, it covers the metering and chargeback concepts. Tons of resources are available on cloud computing, hence the objective here is to give you a high-level overview.

Overview

The origin of cloud computing is not known, and it has nothing to do with real “cloud”. So why is it called *cloud computing*? The word “cloud” is a metaphor to indicate the abstraction between the end user and the resources. Cloud computing is more of a pay-as-you-go model, compared to a do-it-yourself (DIY) model. This can easily be understood from an electricity example. You consume electricity and pay on the basis of the number of units consumed. While using it, you are not concerned about how the electricity is produced, distributed, or managed. Similarly, in cloud computing, you just pay for computing resources for the duration of your use. You are not concerned with how these computing resources are hosted or managed.

In database terms, you will get your database up and running very quickly. You don’t need to worry about procurement of server, configuration, and installation of various software, or with, maintaining the datacenter, infrastructure, hardware, or operating system.

In this book, you will first learn about the cloud computing concepts, followed by “database as a service” details, and then you will learn about the major aspects of database administration, including provisioning, HA and DR setup, monitoring, day-to-day administration activities, and backup and recovery, all from the cloud perspective.

What Is Cloud Computing

Defining cloud computing is complex and many definitions exist. The official definition is given by the National Institute of Standards and Technology (NIST).

It's essentially a web-based service for computing resources, including server, network, storage, and applications. This service is ubiquitous and provided as on-demand basis.

Benefits of Cloud Computing

Companies have traditionally believed in owning the assets and managing them on their own, but this model meant large capital expenditure (CAPEX) and operating expenditure (OPEX) costs. Another challenge with this traditional approach is related to meeting the higher computing requirements due to a single business event. Buying extra hardware and resources for handling spikes means waste of resources during the non-peak times.

The biggest benefit of cloud computing is saving on CAPEX related to infrastructure costs. The various benefits are listed in Table 1-1.

Table 1-1. *Cloud Computing Benefits*

Category	Benefits
Provisioning	Faster provisioning makes cloud computing an ideal platform for test and dev environments
Scalability	Scale up and scale out on-demand
Resource release	Faster resource release/scale down
CAPEX	No investment required for setting up the infrastructure
OPEX	Lower OPEX due to pay-as-you-go model
Availability	Highly available
Skill requirements	Lower skill required due to built-in automation
Accessibility	Accessible using web-based portal, hence it is ubiquitous

Challenges of Cloud Computing

There are many challenges related to cloud computing, some of the key ones are listed here:

- *Cost of service:* While cloud computing is good for ad hoc usage, it becomes more costly if you try to adopt it as the only solution for all your hosting requirements.
- *Laws of the land:* Local laws in various countries don't allow you to keep data on servers that aren't in direct control of the data owner. In a few cases, keeping data outside of a given country is also not allowed.
- *Cloud interoperability:* Moving things between different cloud service providers is a very difficult task.

- *Geographical presence:* Cloud vendors do not have a presence in all geographies, which means cloud computing becomes infeasible for some customers.
- *Application certification:* Not all applications are certified to run in a cloud; this is one of the key challenges in cloud computing adoption.
- *Lack of an integrated solution provider:* Currently no single vendor provides all the cloud services that any given organization needs. For example, converged infrastructure as a service is provided by Oracle, whereas physical hardware as IaaS is provided by IBM SoftLayer.
- *Security:* Security used to be the biggest challenge for cloud computing, but this is less the case nowadays. Cloud service providers get all security certifications and have a proven track record. For example, AWS hosts the American Security agency's cloud computing infrastructure. Many organizations still perceive cloud security as a major challenge.

Cloud Computing Service Models

There are three commonly used service models (see Figure 1-1):

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

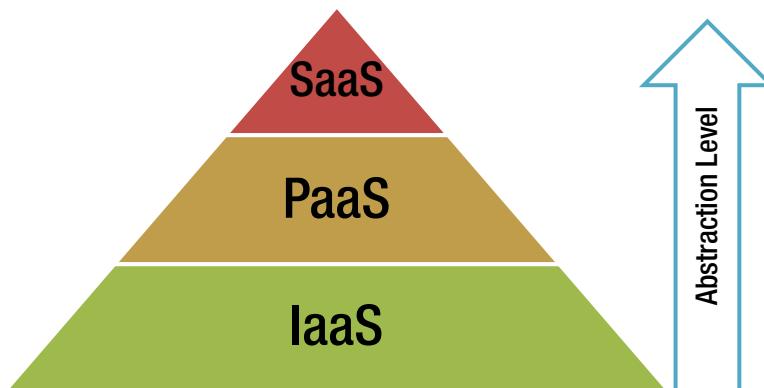


Figure 1-1. Cloud computing service models

Infrastructure as a Service (IaaS)

IaaS providers supply physical/virtual machines, storage, firewall, load balances, VLANs, etc. These are provided from a vast pool of resources. In IaaS, the consumer is responsible for patching and maintaining the operating system and the application software. In Oracle database context, IaaS means that the cloud service provider gives you the server, network, storage, OS, and other software that is required to run Oracle. Sometimes, Oracle software is preinstalled or the server comes with Oracle binaries. You are allowed to bring your own software image.

Platform as a Service (PaaS)

PaaS providers give the dev environment like Dev toolkit (for example, Microsoft Azure and Google App engine). In PaaS, the consumer is not responsible for managing the underlying infrastructure and doesn't administer the underlying cloud components such as the operating system, database, etc. In Oracle database context, PaaS means Oracle database as a service. Here, you don't need to worry about installing Oracle or managing the database server.

Software as a Service (SaaS)

In the SaaS model, the application is placed in the cloud by the SaaS provider. The consumer is not responsible for managing the infrastructure and platform. In Oracle context, SaaS means getting the entire database service from the cloud provider; you just pay for usage. The schema as a service offering from Oracle Cloud is one example of SaaS.

Cloud Deployment Models

Cloud computing can be deployed in many ways and it all depends on the placement of computing resources at the consumer's location (on-premise), at cloud service provider's location, or at both locations. The current cloud deployment models are as follows:

- Public cloud
- Private cloud
- Hybrid cloud
- Community cloud

Public Cloud

This is most common deployment model. In this model, computing resources are present in the cloud service provider's datacenter and are shared with various consumers in a multi-tenant architecture (Figure 1-2). The major advantage of this deployment model is

that you don't need to invest in H/W and effort in setting up the cloud. The disadvantage is that you don't have full control of your computing resources. You can't use this model when there are local laws that prevent you from keeping your data outside your premises.

The public cloud model is good option for startups and any organization that wants to avoid CAPEX costs related to DB servers.



Figure 1-2. Public cloud example

Oracle database can be run on many public cloud providers; however the prominent players are as follows:

- Oracle Cloud
- Microsoft Azure
- Amazon Web Services (AWS)

In the public cloud model, database related services are provided in all three service models (IaaS, PaaS, and SaaS). Oracle provides the following services in Oracle cloud:

- *Oracle Database Cloud Service, Virtual Image*: IaaS offering to run Oracle database
- *Oracle Database Cloud Service*: IaaS/PaaS offering to run Oracle database
- *Oracle Database Exadata Cloud Service*: IaaS offering for running Oracle on Exadata machines
- *Schema as a Service*: SaaS offering to run Oracle database

Microsoft Azure provides IaaS service where you can run Oracle database on virtual machines hosted in their cloud.

AWS provides the following services for Oracle database:

- *EC2 Instance*: IaaS offering
- *Relational Database Services (RDS)*: SaaS offering to run Oracle database

Cloud service providers continuously add enhancements, hence it is recommended that you check the cloud service provider's portal to get the latest offerings.

Private Cloud

In this deployment model, computing resources are placed on-premise. There is one more option here in which computing resources are placed at the cloud provider's premises, but all of these are dedicated for consumers (Figure 1-3). The major advantage of this model is that you have full control of your resources and you can meet any local laws requiring data be kept in your datacenter. The disadvantage is related to the effort needed for private cloud setup.

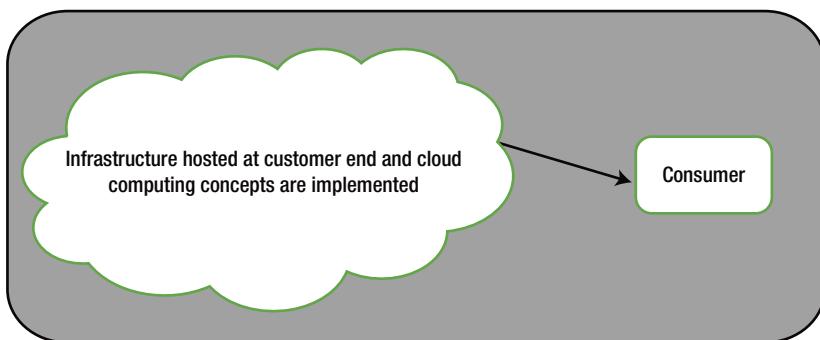


Figure 1-3. Private cloud example

Database private cloud setup is done mostly in house, where companies use commodity server or converged infrastructures like Oracle Exadata, VCE Vblock, and IBM Pure app to host DB servers in a consolidated fashion. Oracle OEM 12c/13c cloud control is used to mimic a cloud-like setup where a self-service portal is created for quick provisioning and a chargeback module is used for metering and billing. In this model, planning, implementation, and on-going maintenance is handled by the company itself.

Private cloud is good option for organizations that want to get ROI from their CAPEX investments done and where hosting in a public cloud is not an option due to compliance requirements.

Hybrid Cloud

This deployment model provides the best of the public and private cloud options. In this model, consumers use both a public cloud and private cloud to cater to different requirements (Figure 1-4). For example, some applications can't move to a public cloud since they are running on end-of-life software, so they remain in a private cloud.

This model is the future of cloud computing and world is moving toward it.

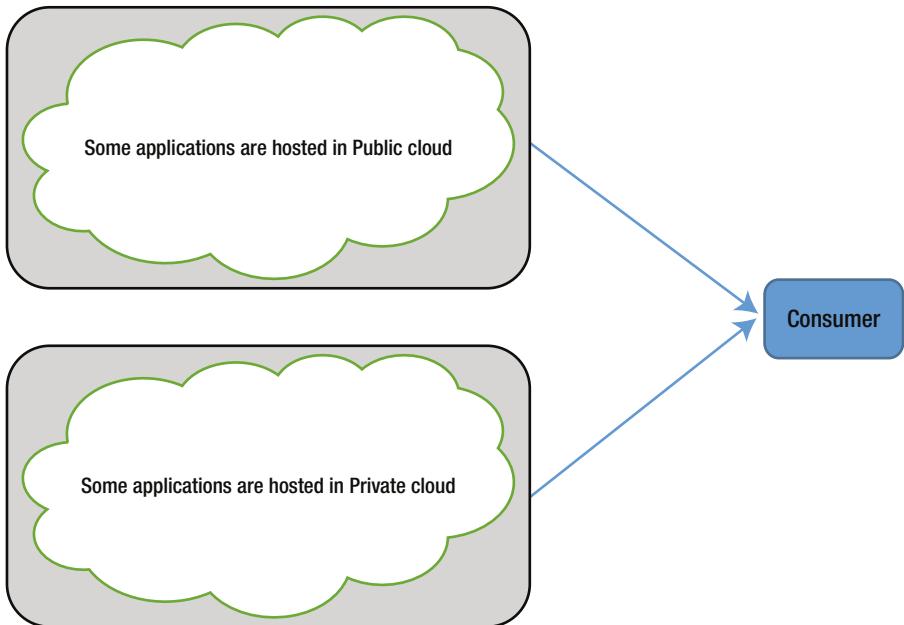


Figure 1-4. Hybrid cloud example

In the hybrid cloud model, some of the databases run on-premise and some are hosted in a public cloud.

The public cloud does not offer support for all database versions. You can't run Oracle 10g on Oracle cloud, so you have to keep such databases in your own private cloud/datacenter.

Similarly, you might want to keep your extremely complex and mission-critical databases in your datacenter rather than host them in a public cloud.

A hybrid cloud is good for situations where you can selectively move some of your workload into a public cloud while retaining others on-premises.

Community Cloud

In the community cloud model, cloud infrastructure is provisioned for the community. Here *community* refers to organizations that have shared objectives. This cloud infrastructure might be managed by one or more organizations that are part of community or by some third-party provider.

This cloud deployment model is not very popular, but it does fall into the official NIST definition, hence it is mentioned here.

Metering and Chargeback

Metering refers to the measurement of resource use, whereas *chargeback* is related to charging based on that metering. These two ideas are essential characteristics of cloud computing. They are always present in the public cloud deployment model, but are optional in the private cloud deployment model.

The metering and chargeback concepts can best be understood by considering the analogy of electricity use in your home. You have a meter that monitors all your energy consumption and then you are charged for the number of units consumed in a given period. While electricity consumption has only one metric, cloud computing has many, including the number of virtual machines, the type of virtual machines, storage, I/O, etc.

The metering and chargeback model in cloud computing is similar to the pay-as-you-go/pre-paid model used in the mobile industry.

Consider this Oracle example. Oracle database pricing (chargeback) is divided into two categories:

- *Metered*: A pay-as-you-go model where no upfront payment is required and invoices are generated on a monthly basis for actual use on an hourly basis.
- *Non-metered*: A monthly pricing. Per hour pricing is not applicable here.

Within each of these categories, there are three options:

1. Oracle Database as a Service
2. Oracle Database Exadata cloud service
3. Oracle Database Virtual Image

In Oracle Database as a Service, Oracle software is installed and Oracle database is created using the option you provide. The database management tools are available for backup, recovery, and patching.

In Oracle Database Virtual Image, you get Oracle software pre-installed on Oracle cloud virtual machine. Database instances are not created automatically, but you can do that using DBCA or manually. Here, you don't get any additional cloud tools.

In the Exadata cloud service, you get Exadata quarter, half, and full rack in a hosted environment.

Summary

This chapter covered the basics of cloud computing, including the deployment models—public cloud, private cloud, hybrid cloud, and community cloud—as well as the service models—IaaS, PaaS, and SaaS. Finally, it covered the metering and chargeback concepts.

The next chapter considers these concepts from the database perspective.

CHAPTER 2



Introduction to Database as a Service

In the last chapter, you learned the basics of cloud computing, so now let's see what cloud computing means from the database perspective. In this chapter, you will learn about the database as a service (DBaaS) concept, including ways to implement it, public cloud offerings around it, and private cloud offerings around it. You will learn about DBaaS as Infrastructure as a service, DBaaS as Platform as a service, and DBaaS in the managed service model. You also learn about the basic terminologies related to prominent cloud providers and the methods and guidelines for choosing the right cloud provider for your DBaaS implementation.

What Is Database as a Service?

As described in the previous chapter, there are three service models for cloud computing—IaaS, PaaS, and SaaS. Recently many new “as-a-service” terms came into existence and cloud computing has become a synonym of XaaS, where X is anything.

Database as a service, also known as DBaaS and very occasionally DaaS, is the most important from a DBA perspective. As the name suggests, it is about providing database services, such as DB servers, database and schema as a service. DBaaS is offered in IaaS, PaaS, and SaaS models.

There are many cloud service providers that allow you to host Oracle databases. The key players are as follows:

- Oracle
- Amazon Web Services (AWS)
- Microsoft Azure

Note More and more providers are being added, so it is recommended you check the latest updates before doing cloud implementation.

You can provision Oracle database with any of these mentioned cloud providers. It is recommended that you use a public cloud provider that is certified and supported by Oracle. Refer to Oracle-approved cloud service providers at <http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf>.

In next section, you will see various offerings provided by these vendors. These are explained using the examples of Oracle, AWS, and Azure cloud, which provide Oracle databases in the cloud.

DBaaS Public Cloud Offerings

DBaaS in the public cloud is offered both as an Infrastructure as a Service (IaaS) and as a Platform as a Service (PaaS). The following section explains the current offerings from Oracle and AWS. These are current as of writing this book. It is said that cloud offerings change on weekly basis, hence it is advised that you check the latest offerings before selecting anything.

Oracle Public Cloud Offering

Oracle offers the following three DB-related services in the cloud. They are all categorized into Platform as a Service, except for Oracle Database Virtual Image option, which is an IaaS offering.

- Schema as a Service
- Database as a Service
- Oracle Database Cloud Exadata Service

Let's take a deeper look at these three offerings.

Schema as a Service

In this scenario, you can get one schema of 5, 20, or 50GB. This schema is encrypted and created on an 11g (11.2.0.4) database. This is fully managed by Oracle. As a DBA, you don't need to perform any administration tasks.

To access and work on this schema, you get three options—via APEX, by using RESTful web services, or by SQL Developer. You don't get direct connectivity using SQL*Net. You can load data in this schema using SQL developer or APEX, or you can export the data from this schema.

This service runs on Exadata and it is very good for faster application development and deployment where you don't want to manage any database aspects. It is highly scalable and available.

Some of the limitations of this offering are as follows:

- No instance-level changes allowed
- No access to underlying hardware or OS
- No storing of files on a filesystem
- No installation of any application
- No access using Oracle net services

Database as a Service

In this service, you have two service levels (Figure 2-1).

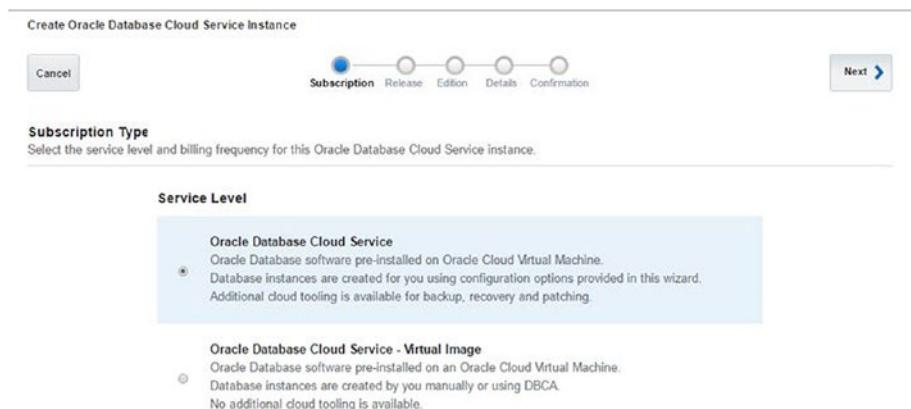


Figure 2-1. Service levels in Oracle database cloud service

Oracle Database Cloud Service: Virtual Image

In this offering, you get Oracle software pre-installed on Oracle cloud virtual machine. You can get started by simply creating the database using DBCA or using a silent installation. This is more of IaaS offering, but Oracle classifies it as PaaS since Oracle binaries are also preinstalled. Here, all the DB maintenance operation is your responsibility. Besides using Oracle's given tools, you can use third-party tools and custom scripts to manage your database.

Oracle Database Cloud Service

In this offering, you get software along with Oracle database. This database is created as per the specifications you provide. This option also gives you cloud tools for doing backups, recovery, patching, upgrading, and monitoring. These tools are as follows:

- `bkup_api`
- `orec` subcommand of the `dbaascli` utility
- `dbpatchm` subcommand of the `dbaascli` utility
- DBaaS monitor
- `raccli` utility

Database provisioning is explained in the next chapter and the tools are covered in Chapter 9, “Manage and Monitor”.

In both of these service levels, you have two billing options:

- Hourly
- Monthly

From a DB version perspective, you have three choices:

- Oracle database 11g R2 (11.2.0.4)
- Oracle database 12c R1 (12.1.0.2)
- Oracle database 12c R2 (12.2.0.1)

From an Edition perspective, you have four choices:

- *Standard Edition (SE)*: Full database instance with up to 16 OCPUs
- *Enterprise Edition (EE)*: Comes with all standard features of Enterprise Edition (EE), plus DataGuard and Transparent Data Encryption (TDE)
- *High Performance*: Comes with all EE packs and options except RAC, RAC One node, in-memory, and Active DataGuard
- *Extreme Performance*: Comes with all EE packs and options except RAC One node

From a compute perspective, you have several combinations of RAM and OCPUs (Oracle CPUs) to choose from. See Table 2-1.

Table 2-1. Oracle Virtual Image Shapes

Category	Shape Name	Size
General Purpose	OC3	1 OCPU, 7.5GB RAM
General Purpose	OC4	2 OCpus, 15GB RAM
General Purpose	OC5	4 OCpus, 30GB RAM
General Purpose	OC6	8 OCpus, 60GB RAM
General Purpose	OC7	16 OCpus, 120GB RAM
High Memory	OC1M	1 OCpus, 15GB RAM
High Memory	OC2M	2 OCpus, 30GB RAM
High Memory	OC3M	4 OCpus, 60GB RAM
High Memory	OC4M	8 OCpus, 120GB RAM
High Memory	OC5M	16 OCpus, 240GB RAM

Oracle Database Cloud Exadata Service

In this offering, you get database hosting on Exadata machines on Oracle cloud. This is very useful for mission-critical databases where you need consistently good performance.

There are three configuration options:

- *Quarter rack*: Two compute nodes and three storage cells
- *Half rack*: Four compute nodes and six storage cells
- *Full rack*: Eight compute nodes and 12 storage cells

Each compute node comes with a fixed amount of memory, network, and storage; however, you are allowed to enable extra CPU cores as per your need.

As per IaaS offering norms, you don't get access to underlying hardware, storage cells, and InfiniBand.

In Exadata service, you can deploy as many instances as you want. It comes with all Enterprise Edition features, options, and packs, along with cloud tools for performing common database administration tasks.

You have full control of the database and you are responsible for performing database backups, recovery, and all other DB administration tasks. Oracle is also offering this Exadata Service under a cloud platform with the option to host Exadata on-premise. This is really helpful for expanding the scope of public cloud based DBaaS to the business-critical applications addressing the concerns related to security and data hosting locations.

Oracle Cloud Choices

Table 2-2 explains Oracle Database Cloud options with scenarios.

Table 2-2. When to Choose What in Oracle Cloud

Oracle Cloud Offering	Ideal For
Schema as a Service	When you need database platform for faster application development or deployment and you don't want to do database administration related tasks
Oracle Database Cloud Service	When you need a database server with preconfigured Oracle instance and various tools for automated maintenance tasks
Oracle Database Cloud Service: Virtual Image	When you need a database server with Oracle software installed but no database created
Exadata Service	When you want high performance and various Exadata features in the cloud or you are migrating your database hosted in on-premises Exadata

AWS Public Cloud Offering

Amazon Web Services (AWS) is a cloud computing platform from Amazon. Oracle database was first made available in AWS in 2006. Oracle on AWS can be deployed in two ways:

- On Amazon Elastic Compute Cloud (EC2)
- On Amazon Relational Database Services (RDS)

All the DBaaS offerings, from Oracle Cloud or from AWS, have on-demand scalability that helps in meeting storage, processing power, and memory requirements. Since database is hosted in the cloud, you have high-availability options like Multi-AZ, which is discussed in Chapter 4.

Oracle DB on Amazon EC2

Amazon EC2 is an IaaS offering, hence you get full control of the OS, Oracle software installation and configuration, and all the other database administration tasks. This is good if you want to run old Oracle database versions (e.g., Oracle 10g) or you want to have full control over database deployment. The following are key highlights of this offering:

- Easy to set up
- Highly reliable
- High availability
- Full control for the consumer

- Easy integration with other AWS offerings such as S3
- Easy scalability

Oracle DB on Amazon RDS

Amazon RDS is the relational database services offering from Amazon. The following are key highlights of this offering:

- *Easy to set up*: You can bring up Oracle instance in a few minutes.
- *Easy administration*: AWS manages your installation, backup, recovery, software patching, minor upgrades, monitoring, snapshots, and automatic host replacement.
- *Easy scalability*: Quick scalability by clicking of few buttons.
- *High availability*: Uses Multi-AZ (availability zones).
- *Good performance*: Availability of 32vCPU and 244GB RAM compute machine, along with SSD backend storage, helps meet performance requirements.
- *Security*: Uses Transparent Data Encryption and Virtual Private Cloud.

The following are some of key Oracle features that are not supported in Amazon RDS. For a complete list and the most up-to-date information, refer to <http://aws.amazon.com/rds/oracle/>.

- DataGuard and Active DataGuard
- Real Application cluster (RAC)
- ASM
- Database Vault
- Oracle Enterprise Manager (OEM) Grid Control

Supported Versions and Editions

Only Oracle 11gR2 and Oracle 12c are supported in Amazon RDS.

The following Oracle editions are supported in Amazon RDS. These are dependent on the licensing model.

- Standard Edition: Bring your own license (BYOL)
- Standard Edition One: License included and BYOL
- Standard Edition Two: BYOL
- Enterprise Edition: BYOL

AWS Choices

Table 2-3 describes the scenarios when you should choose hosting Oracle DB on EC2 or RDS.

Table 2-3. When to Choose What in AWS

Choose	When
Amazon EC2	<ul style="list-style-type: none"> • You want to run older software versions that are not supported in RDS. • You want full control over database installation and configuration. • Your database size exceeds the current maximum allowed in RDS. • You have DBAs who can manage the database in the cloud. • You want to bring your own license (BYOL) for running Oracle database in EC2.
Amazon RDS	<ul style="list-style-type: none"> • You do not want to do regular database administration tasks and want to focus only on performance tuning and application-related tasks. • You do not want to spend effort in manually configuring the DR solution, like Multi-AZ. • You want to use Oracle Standard edition One with the license-included model. There are many edition options available with BYOL, but this particular option is only available in Amazon RDS.

Microsoft Azure Public Cloud Offering

Oracle database is certified and supported on Microsoft Azure Cloud. Microsoft Azure platform provides only an Infrastructure as a Service (IaaS) offering for running Oracle databases. As part of this, Microsoft provides options for selecting virtual machines, storage, and virtual networks.

Supported Versions and Editions

You get these two options in Azure (Figure 2-2):

- Oracle database 12.1.0.2 Enterprise Edition
- Oracle database 12.1.0.2 Standard Edition

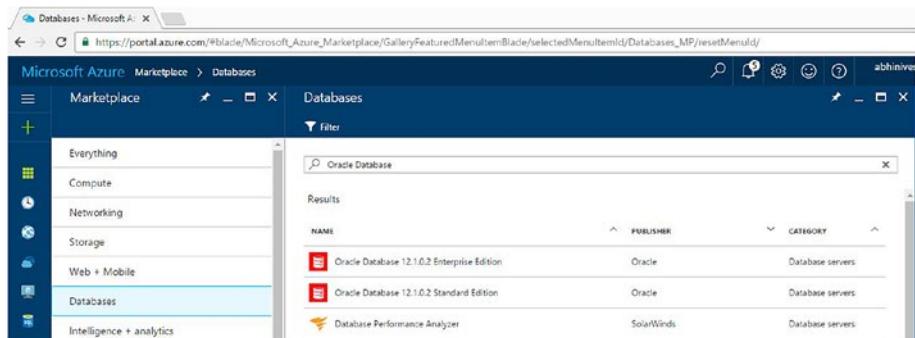


Figure 2-2. Microsoft Azure offerings for Oracle database

Microsoft Azure stopped the pay-as-you-go model for Oracle DB and now only supports bring-your-own-license (BYOL). Since Azure Cloud has less functionality for Oracle database hosting, it is not covered in this book.

DBaaS in Private Cloud

DBaaS in a private cloud means setting up your own cloud using your own infrastructure. This is useful because you have full control of the entire infrastructure and the end user gets all the cloud computing benefits.

DBaaS in a private cloud can be set up either by some consulting partner such as Oracle consulting or by your own in-house IT team.

Oracle Enterprise Manager 12c (OEM) and OEM 12 Cloud management pack features and options are used to enable self-service portal, metering, and chargeback, as well as managing the database lifecycle.

Oracle database 12c multi-tenant feature is a very good option for setting up a private cloud.

A private cloud can be set up on a traditional infrastructure, on a converged infrastructure like Vblock, or on engineered system like Exadata, Supercluster, etc. Oracle Virtual Machine (OVM) can be taken for virtualization. The DBaaS in a private cloud offering is very challenging compared to DBaaS in Public cloud since you are responsible for configuration and for on-going maintenance.

This is good option for organizations that can't move their databases to the cloud due to security and compliance issues.

DBaaS in Managed Services Model

Managed services means that cloud service providers are responsible end to end for managing and maintaining your database.

Oracle Database Exadata Cloud Service and AWS's RDS offering are examples of managed services. As of the writing of this book, Oracle is yet to release managed services offerings for Oracle databases in a non-Exadata environment.

Getting Started with Database as a Service

Now that you understand the DBaaS concepts and various offerings from Oracle, Microsoft, and AWS, it's time to look at what you need to get started. Before you start your DBaaS implementation, you need to do the following three things:

1. Select your vendor.
2. Select the services.
3. Choose the right subscription.

Selecting Your DBaaS Vendor

Vendor selection depends on many factors, but the following three factors are the most important when choosing the right vendor:

- Services offered by vendor
- Features supported by vendor
- Price

Cloud provider selection requires detailed analysis and it is beyond the scope of this book.

Selecting the DBaaS Services

Each cloud provider provides many offerings around Oracle database, so it becomes very important to choose the right services.

Table 2-4 details the offerings around various cloud service models.

Table 2-4. DBaaS Offerings Around Various Cloud Service Models

Cloud Service Model	Offerings
IaaS	Oracle Database Virtual Image, Oracle Database Exadata cloud service, AWS EC2 instance, Azure Virtual machine
PaaS	Oracle Database as a Service
SaaS	Oracle Schema as a Service, Amazon RDS

Choosing the Right DBaaS Subscription

Each cloud service provider has its own method of metering and chargeback, hence it is extremely important to choose the subscription based on your business needs. For example, for learning purposes, trials or free subscriptions come with certain restrictions. For long-term use, the pre-paid option is usually cheaper than the pay-as-you-go model.

Getting Acquainted with the Basic Terminology in Database as a Service

This section lists the basic terminology and descriptions of each term. These terms are frequently used in the DBaaS world, hence you should make yourself familiar with them before you start using DBaaS. These terms are explained in relevant chapters as well.

Oracle Cloud Terminology

- *Service levels:* Refers to the options that Oracle cloud offers
- *Virtual image:* The virtual machine
- *OCPU:* Refers to Oracle CPU, which is equivalent to one physical core of an Intel Xeon processor
- *Cloud storage:* Storage option present in the cloud
- *Subscription:* Registration for Oracle cloud service
- *Region:* Refers to the geography where the datacenters are present
- *Compute:* Refers to CPU, memory, network, and storage
- *Console:* GUI to access and manage Oracle cloud service
- *Shape:* Virtual image sizes

Amazon Web Services Terminology

- *EC2 instance:* Refers to the virtual machine
- *RDS:* Relational Database Services
- *Region:* Refers to the geography where the datacenters are present
- *EBS:* Block Storage in AWS
- *AZ:* Availability Zone
- *DB Instance Class:* Same as *shape* in Oracle Cloud
- *BYOL:* Bring your own license
- *DB Engine:* Standard Edition or Enterprise Edition
- *VPC (Virtual Private Cloud):* Virtual datacenter in AWS

Microsoft Azure Terminology

- *Virtual Machine:* Compute resources provisioned using hypervisor
- *Storage account:* Storage provided by Azure; requires you to create one or more storage account
- *Subscription:* Registration for Azure cloud services
- *Classic/old portal:* <https://manage.windowsazure.com/> is referred to as the classic portal.
- *New portal:* <https://portal.azure.com/> refers to the new portal

Summary

In this chapter, you learned the meaning of DBaaS, read an overview of DBaaS offerings given by the top three cloud service providers, and learned about the benefits and use cases of each of these offerings. You learned about how to choose which offering, read about the supported versions and editions, and learned the basic terminology related to DBaaS.

The next chapter covers the provisioning options in Oracle cloud and Amazon cloud.

PART II



Database Administration in Cloud

Part II of this book covers all aspects of Oracle database administration in cloud. It consists of 7 chapters, brief overview of each of the chapter is given below-

Chapter [3](#) covers the provisioning of Oracle database in cloud.

Chapter [4](#) covers the High availability options like- RAC setup in cloud.

Chapter [5](#) covers the Disaster recovery options like- DataGuard setup, Multi AZ deployment in cloud.

Chapter [6](#) covers Database security aspects.

Chapter [7](#) covers Database migration to cloud.

Chapter [8](#) gives details around Database backup and restore approaches for cloud based database.

Chapter [9](#) covers Database management and monitoring approaches and different tools available for performing Database management activities.

CHAPTER 3



Provisioning

In this chapter, you go thru database provisioning in Oracle Cloud and AWS. You will learn about the various options that you can select during database provisioning. Some configurations, like backup configuration for Oracle Cloud, if not done during database creation, will never be available for modification. GUI-based provisioning is a simple and quick way to create your database with AWS and Oracle Cloud.

Database Provisioning Overview

Creating Oracle database and making it available for end users is called *provisioning* the database. Provisioning can be as per specific requirements or can be completed for a standard offering. In the context of DB provisioning for cloud based offerings, it is usually a standard offering with very little room for customization. All the customization can be done by the respective support team after it is made available to them.

You might have done DB provisioning multiple times for non-cloud based databases. In such cases, the approach could be completely manual or partially automated. It can be completely automated as well and is called a factory-based approach, where the standard set of activities are performed via automation and a standard DB installation is completed.

For cloud-based DB provisioning, you must get the requirements from the user/client with minimal interaction and in the minimum amount of time. A self-service portal is key to meeting this requirement.

Also for non-cloud based provisioning, the turnaround time can be long, as it may involve budgeting, purchase, hardware shipping, OS, storage, and other required configurations. Some of these steps can be skipped if the target is to create the DB on a Virtual Machine (VM) or on a logical partition (LPAR). For cloud-based DB provisioning, on the other hand, you need to be ready with requirements for CPUs, OS, storage, and other hardware and software needs. On receipt of the request, the DB provision is completed quickly, usually in minutes, not days.

In the next section, you go thru a step-by-step process for provisioning Oracle database with two prominent cloud service providers—Oracle Cloud and Amazon Web Services.

Most of the cloud service providers do not support Oracle database 10g or below. The cloud-based provisioning is limited to 11g or 12c databases only. There are various offerings, including:

- Database instance (11g)
- Pluggable database (12c)
- Schema service

Database Provisioning in Oracle Cloud

This section describes the step-by-step process for provisioning Oracle database using Oracle Cloud services. At a high level, this can be divided into the following steps.

1. Create Oracle Cloud subscription.
2. Select Oracle Database Cloud offering.
3. Create the service.

Let's look at these steps in detail.

Creating Oracle Cloud Subscription

To start with, you learn how to provision a database using Oracle Cloud service and create a pluggable database. As the first step, you need to subscribe to Oracle Cloud service.

Go to <https://cloud.oracle.com/home>

Login to your Oracle account or create one if you don't have Oracle account. Oracle Cloud subscription is used by Oracle to promote new offerings and make subscribers aware of new features added to the service. You may also get invites to webcasts and events, in order to keep updated.

To create a database, you need to click on Database under the Platform (PaaS) category, as shown in Figure 3-1.

Platform (PaaS)



Figure 3-1. Oracle Cloud PaaS offering

There are two offerings available. The Database as a Service option will let you create database instances, whereas the Database Schema Service option will let you create a schema to be used as per your requirements.

You need to provision a database, so select Database as a Service, as shown in Figure 3-2. Note that Oracle offers a free trial service to help you explore many features without any cost.



Figure 3-2. Oracle Cloud Database as a Service offering

As the next step, you need to sign up for the subscription of Oracle Cloud services. Fill in the required details to move to the next step, as shown in Figure 3-3.

 A screenshot of the Oracle Cloud subscription form. The top navigation bar includes 'ORACLE CLOUD' and 'Preferences'. Below it, a large blue circle contains the text 'Sign Up for a Trial Subscription to Oracle Public Cloud Services'. The main area starts with a 'Hello!' greeting and a note about verifying contact details. It features a horizontal menu with links for Oracle Database, Database Backup, Java, Storage, Compute, Application Container, Developer, and SOA. The 'Contact Details' section is highlighted with a blue header and contains fields for First Name (Niraj), Last Name (Mahajan), Company (Cloud DBA), and Country (India).

Figure 3-3. Oracle Cloud subscription form: contact details

Also provide the correct contact details, like your phone number, which is required for verification. Provide the name for your domain, which will be used for the database as well as for other Oracle Public cloud services, as shown in Figure 3-4.

The screenshot shows the 'Verification Code' step of the Oracle Cloud subscription process. It includes fields for 'Country Calling Code' (India (+91)), 'Mobile Number' (+91 1234567890), a 'Request Code' button, and a 'Verification Code' field (1234). A note states: 'This verification code will be sent in a text message to your mobile phone. Standard text messaging rates apply. Oracle may also contact you at this phone number if we have questions about your trial.' Below this is the 'Trial Details' section, which asks for a 'Name' (Create New Account) and provides a preview of the My Services URL (https://myservices.oraclecloud.com/mycloud/clouddb...). A checkbox for accepting Oracle's Terms and Conditions is checked, and there are 'Cancel' and 'Sign up' buttons.

Figure 3-4. Oracle Cloud subscription form: contact verification details

Once you have provided all the mandatory information in the required format, you will receive a success message, as shown in Figure 3-5.



Figure 3-5. Oracle Cloud subscription form completion

This will trigger an e-mail to the account you provided earlier, where you will have the link for your account and service administration. You will also be provided with a temporary password, which needs to be changed upon the first login. See Figure 3-6 for the details.

Access Details

User Name and Temporary Password



Use these credentials to sign in to My Services and your Service Console URL.

Username: nirajmahajan@yahoo.com
 Temporary Password: jKFZ7w3j
 Identity Domain: usclouddb

My Account Administration



Sign in to My Account to manage your trials and all your Oracle Cloud services.

My Account URL: <https://myaccount.cloud.oracle.com/mycloud/faces/dashboard.jspx>
 Username: Use your Oracle account user name.
 Password: Use your Oracle account password.

My Services Administration



Sign in to My Services to monitor this trial environment, including usage data and status.

My Services URL: <https://myservices.us.oraclecloud.com/mycloud/usclouddb/faces/dashboard.jspx>
 Identity Domain: usclouddb

Figure 3-6. Oracle Cloud subscription: ACCESS details

You can also set a security question, your timezone, and any language preferences and then continue.

Selecting Oracle Database Cloud Offering

Now you are ready to create your database using Oracle Public cloud services. Select Oracle Database Cloud Service, as shown in Figure 3-7 from the multiple service offerings that are available.



Oracle Database Cloud Service

Subscription: Trial (Expires:27-Jul-2016 12:17 PM IST)
 Data Center: US Commercial 2
 Identity Domain: usclouddb
 Cloud Services Account: usclouddb
 Category: Oracle Database Public Cloud Services

Figure 3-7. Choose Oracle Cloud Service

You can choose to follow the tutorials or directly jump to DB creation by clicking Go to Console button, as shown in Figure 3-8.



Figure 3-8. Oracle Database cloud: welcome page

Creating the Service

Once you're inside Oracle Database Cloud Service, you have to click on the Create Service button to start the DB provisioning process, as shown in Figure 3-9.

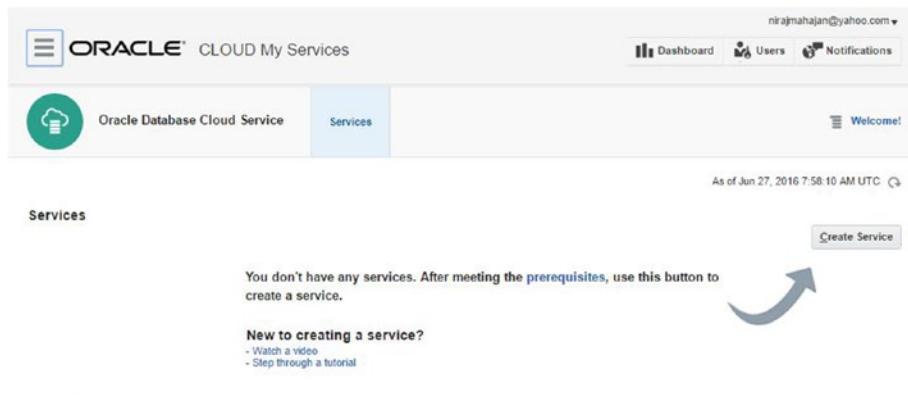


Figure 3-9. Oracle Cloud: create service process

Service creation is five-step process:

1. Subscription
2. Release
3. Edition
4. Configuration Details
5. Confirmation

Let's look at each of these steps in detail.

Subscription

As shown in Figure 3-10, there are options to either create a virtual image and then create the DB manually using the preinstalled binaries, giving you high flexibility for setting your database configurations. Otherwise, you can opt for Oracle Database Cloud Service, where you will provide your requirements via a wizard and create a DB instance quickly. For now, let's choose the wizard based approach with the monthly billing frequency.

The screenshot shows the Oracle Cloud My Services interface for creating a new Oracle Database Cloud Service instance. The process is a five-step wizard:

- Step 1: Subscription** (selected)
- Step 2: Release**
- Step 3: Edition**
- Step 4: Details**
- Step 5: Confirmation**

Subscription Type: Oracle Database Cloud Service

Service Level: Oracle Database Cloud Service - Virtual Image

Billing Frequency: Monthly

Figure 3-10. Oracle Cloud: service level and billing process

Release

As the name suggests, in this step you have to decide on the DB release to be used, as shown in Figure 3-11. The supported release versions are 11g and 12c only, both running on Oracle Linux platform.

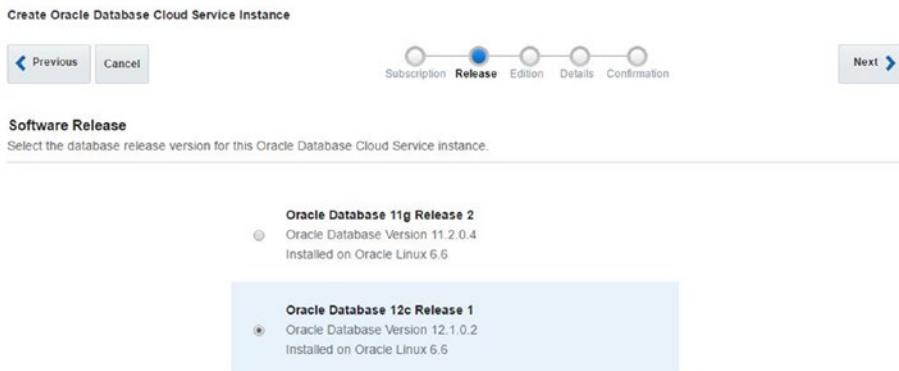


Figure 3-11. Oracle Cloud: selecting the release

Edition

The next step is to select the database edition; you have four options:

- *Standard Edition*: Use the Standard Edition features for your cloud DB
- *Enterprise Edition*: Contains all the components of Oracle DB
- *Enterprise Edition - High Performance*: Use the Enterprise Edition features for your cloud DB with enterprise management pack, excluding RAC, RAC One Node, In-Memory, and Active DataGuard
- *Enterprise Edition - Extreme Performance*: Use all the Enterprise Edition features for your cloud DB with enterprise management pack, excluding RAC One Node

Make a selection, as shown in Figure 3-12.

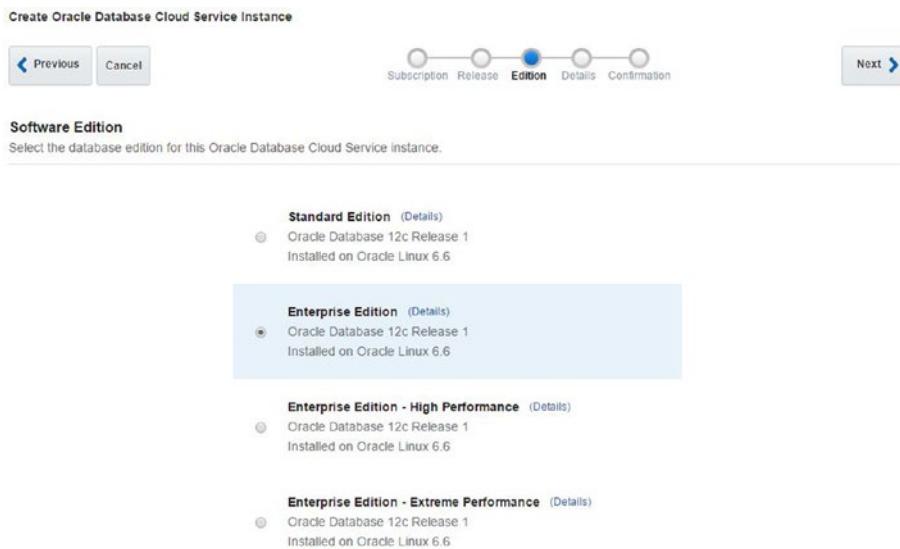


Figure 3-12. Oracle Cloud: software edition selection

The next stage is where all the details are collected for the DB to be created. This includes service, database, backup, and recovery configuration, along with a few more details.

Service Configuration Details

This is where you provide the service name, description, hardware requirement (*shape*), timezone, and SSH public key for secure connectivity. See Figure 3-13.

The 'Service Configuration' form includes fields for Service Name (cloud-db01), Description (cloud-db01), Shape (OC3 - 1 OCPU, 7.5 GB RAM), Timezone (UTC+05:30 India Standard), and SSH Public Key (Niraj_Public_key.pub). An 'Edit' button is shown for the SSH key field.

* Service Name	cloud-db01	?
Description	cloud-db01	?
* Shape	OC3 - 1 OCPU, 7.5 GB RAM	▼
* Timezone	(UTC+05:30) India Standard	▼
* SSH Public Key	Niraj_Public_key.pub	Edit ?

Figure 3-13. Oracle Cloud: service configuration

There are multiple standard options (as shown in Figure 3-14) for choosing hardware configuration, which is called *shape*, in Oracle public cloud context.

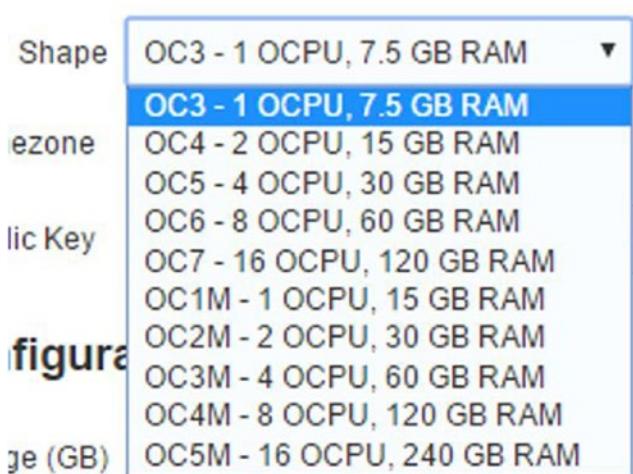


Figure 3-14. Oracle Cloud: shape option

Next is database configuration, which is where you provide the storage requirements, password for administration, SID, PDB name, and character set information, as shown in Figure 3-15.

Database Configuration

* Usable Database Storage (GB)	100
Total Data File Storage (GB)	171
* Administration Password
* Confirm Password
* DB Name (SID)	clouddb
* PDB Name	testdb1
* Character Set	AL32UTF8 - Unicode Univ
* National Character Set	AL16UTF16 - Unicode UTF

Figure 3-15. Oracle Cloud: database configuration

The next section, as shown in Figure 3-16, is about backup and recovery. This information is covered in detail in Chapter 8, “Backup and Restore”.

The screenshot shows the 'Backup and Recovery Configuration' section of the Oracle Cloud interface. It includes fields for Backup Destination (Cloud Storage Only), Cloud Storage Container (usclouddba/backup), Cloud Storage User Name (cloud_dba), and Cloud Storage Password (a masked password). A tooltip 'Enter a password user' is visible next to the password field.

* Backup Destination	Cloud Storage Only
* Cloud Storage Container	usclouddba/backup
* Cloud Storage User Name	cloud_dba
* Cloud Storage Password

Figure 3-16. Oracle Cloud: backup and recovery configuration

Once you provide all this information, a summary, as shown in Figure 3-17, is displayed so that you can review the provided details before submitting the actual database creation request.

Confirmation

Confirm your responses and create this Oracle Database Cloud Service instance.

The confirmation screen displays the following database configuration details:

- Service Level: Oracle Database Cloud Service
- Billing Frequency: Monthly
- Software Release: Oracle Database 12c Release 1
- Software Edition: Enterprise Edition
- Service Name: cloud-db01
- Description: cloud-db01
- Shape: OC3 - 1 OCPU, 7.5 GB RAM
- Timezone: (UTC+05:30) India Standard Time(Asia/Calcutta)
- Key: Niraj_Public_key.pub
- Usable Database Storage: 100
- Total Data File Storage: 171
- DB Name (SID): clouddb
- PDB Name: testdb1
- Character Set: AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit
- National Character Set: AL16UTF16 - Unicode UTF-16 Universal character set
- Standby Database with Data Guard: No
- Disaster Recovery: No
- Include "Demos" PDB: No
- Include GoldenGate: No

At the bottom, it shows the backup configuration:

- Backup Destination: Cloud Storage Only
- Username: cloud_dba
- Cloud Storage Container: usclouddba/backup

Figure 3-17. Oracle Cloud: summary screen

You are now at final lap for provisioning your database in Oracle Public cloud, as shown in Figure 3-18.



Figure 3-18. Oracle Cloud: create option

Once you click the Create button, the request is submitted for creating your database. The progress is updated under the Services page, as shown in Figure 3-19.

Services	OCPUs	Memory	Storage	Public IPs
1	1	7.5 GB	180 GB	1

Services	
Enter a full or partial service name	<input type="button" value="Search"/>
cloud-db01 Status: In Progress Version: 12.1.0.2 Edition: Enterprise Edition	Submitted On: Jun 27, 2016 8:13:52 AM UTC OCPUs: 1 Memory: 7.5 GB Storage: 180 GB

Figure 3-19. Oracle Cloud: service creation status

It takes around half an hour to get your database ready. Once it's ready, public IP and port information is available at the service page, as shown in Figure 3-20.

cloud-db01 Public IP: 129.144.16.63	SQL *Net Port: 1521 SID: clouddb PDB Name: testdb1	OCPUs: 1 Memory: 7.5 GB Storage: 180 GB	
---	--	---	--

Figure 3-20. Oracle Cloud: service details

Accessing Your Database

You can access your database with different methods, including local SQL Plus, SQL Developer, DBaaS monitor, APEX, and EM Console. See Figure 3-21.

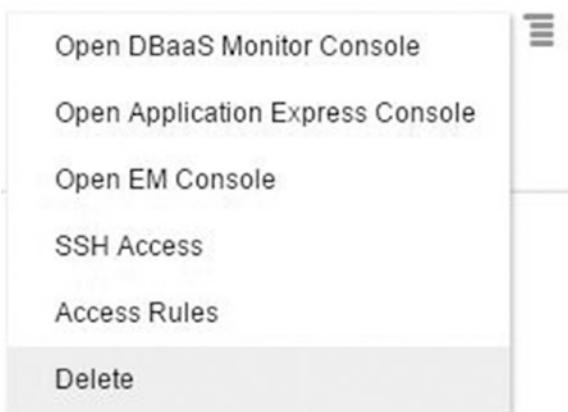


Figure 3-21. Oracle Cloud: database access option

You now need to enable access rules for DBaaS and DB Listener so that you can access your database using DBaaS Monitor and SQL Developer, as shown in Figure 3-22.



Status	Rule Name	Source	Destination	Ports	Description	Rule Type	Actions
Success	ora_p2_ssh	PUBLIC-INT...	DB	22		DEF/ Enable	☰
Success	ora_p2_dblistener	PUBLIC-INT...	DB	1521		DEF/ Disable	☰
Success	ora_p2_http	PUBLIC-INT...	DB	80		DEF/ Delete	☰
Success	ora_p2_httpsol	PUBLIC-INT...	DB	443		DEFAULT	☰
Success	ora_p2_httpadmin	PUBLIC-INT...	DB	4848		DEFAULT	☰
Success	ora_p2_dbconsole	PUBLIC-INT...	DB	1158		DEFAULT	☰
Success	ora_p2_dbexpress	PUBLIC-INT...	DB	5500		DEFAULT	☰
Success	ora_trusted_hosts_dblistener	127.0.0.1/32	DB	1521	DO NOT MODIFY: A seerule to allow speci...	DEFAULT	☰

Figure 3-22. Oracle Cloud: enable access rules

The next step is to access your database in Oracle Public cloud as any other on-premise database using SQL Developer by providing the proper connection details. Your hostname is the public IP of your database. See Figure 3-23 for more details.

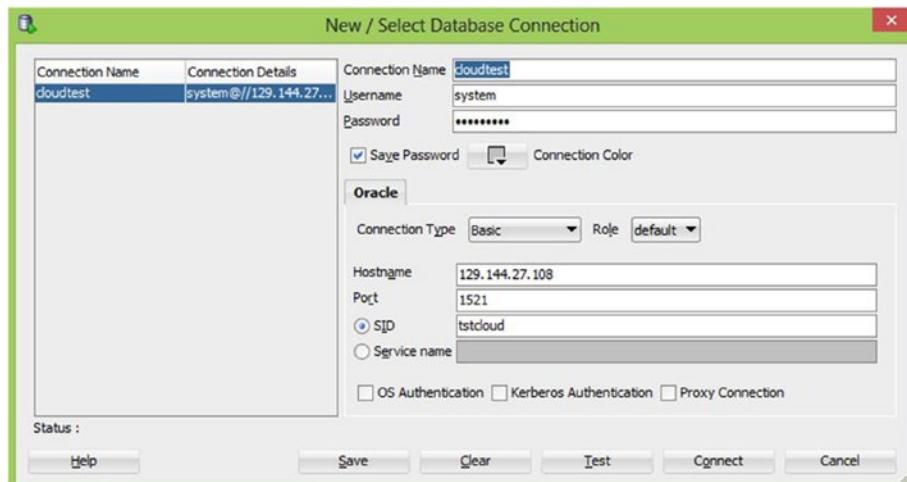


Figure 3-23. Oracle Cloud: DB access using SQL developer

Now you can access your database like any other database and run queries, as demonstrated in Figure 3-24.

FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS
1 /u02/app/oracle/oradata/TstCloud/users01.dbf	6	USERS	5242880	640
2 /u02/app/oracle/oradata/TstCloud/undotbs01.dbf	4	UNDOTBS1	592445440	72320
3 /u02/app/oracle/oradata/TstCloud/system01.dbf	1	SYSTEM	838860800	102400
4 /u02/app/oracle/oradata/TstCloud/sysaux01.dbf	3	SYSAUX	702545920	85760

Figure 3-24. Oracle Cloud: query execution in DB

Database Provisioning in AWS

In this next section, you learn about how to provision Oracle database using Amazon Web Services (AWS).

AWS has an offering under Relational Database Services (RDS) where you can create your Oracle database. There is also option to create your database using EC2.

For this example, DB creation using RDS is covered. RDS is a managed service offering where Amazon handles most of the administration tasks, including DB backup, space allocation, etc.

To provision a database using AWS RDS, go to <http://aws.amazon.com/> and sign up or sign in by providing the required details.

As the next step, select RDS under Database from the Amazon Web Services page. RDS is the managed Relational Database Service offering from AWS. Select Launch DB Instance from the Instances menu to start the DB provisioning steps, as shown in Figure 3-25.

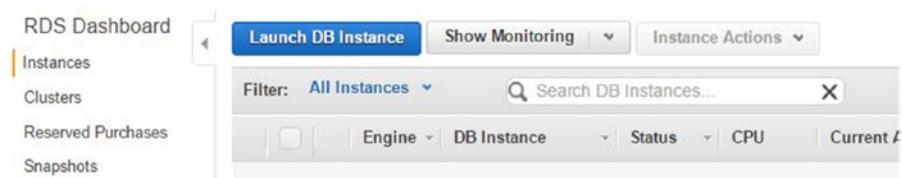


Figure 3-25. AWS RDS: launching the DB instance

It is four-step process

1. Engine Selection
2. DB environment specification
3. Specifying the DB details
4. Configuring the advanced settings

The details of each of these steps are covered in the following sections.

Selecting the Proper Engine

Select the DB engine that you would like to use, as shown in Figure 3-26. You can select Enterprise Edition or select one of the Standard Edition options.

Select Engine

To get started, choose a DB Engine below and click Select.

 Amazon Aurora	Oracle EE Oracle Database Enterprise Edition <hr/>  MySQL <hr/>  MariaDB <hr/>  PostgreSQL <hr/>  ORACLE <hr/>  Microsoft SQL Server	<input type="button" value="Select"/>
	Oracle SE Oracle Database Standard Edition <hr/>  Oracle <hr/>  Microsoft SQL Server	<input type="button" value="Select"/>
	Oracle SE One Oracle Database Standard Edition One <hr/>  Oracle <hr/>  Microsoft SQL Server	<input type="button" value="Select"/>
	Oracle SE Two Oracle Database Standard Edition Two <hr/>  Oracle <hr/>  Microsoft SQL Server	<input type="button" value="Select"/>

Figure 3-26. AWS RDS: select the proper engine based on your needs

Only SE One has the option of license included; for all others, you need to bring your own Oracle DB license (BYOL). When you are migrating your Oracle database from on-premises to the cloud, you can utilize your investment in license purchases by using the existing or spare licenses with the BYOL option.

As shown in Figure 3-27, this is part of the Specify DB Details section.

Instance Specifications

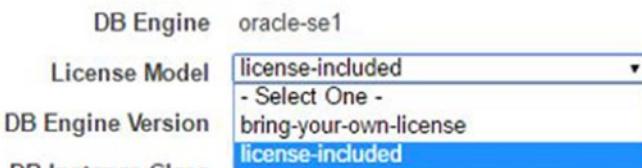


Figure 3-27. AWS RDS: instance specification

Specifying the DB Environment

You can choose bring-your-own-license (BYOL) and use the DB instance for non-production purposes with no need to pay for a license. If you intend to use this database for production purposes, declare this fact in the next step, as shown in Figure 3-28.

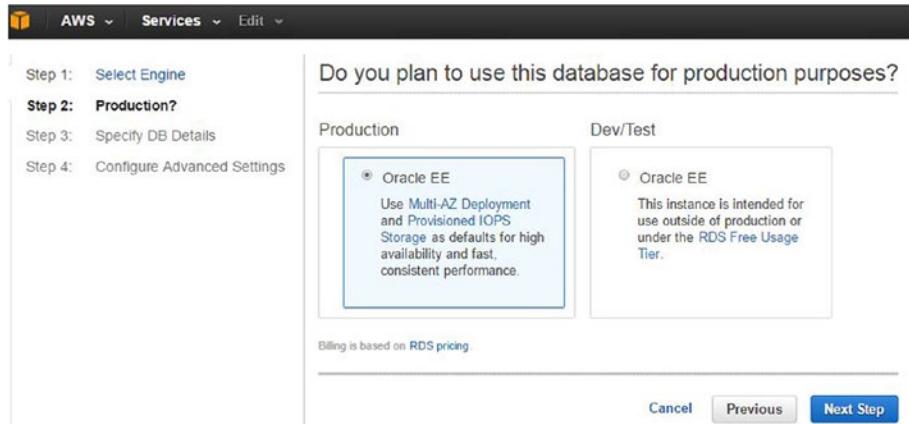


Figure 3-28. AWS RDS: DB environment specification

Specifying the DB Details

Next, you need to provide the database specifications like version, CPU and memory details, storage type and size, along with a username and password for DB administrator, as shown in Figure 3-29.

Instance Specifications

DB Engine	oracle-ee
License Model	<input type="text" value="bring-your-own-license"/>
DB Engine Version	<input type="text" value="12.1.0.2.v4"/>
DB Instance Class	<input type="text" value="db.t2.micro — 1 vCPU, 1 GiB RAM"/>
Multi-AZ Deployment	<input type="text" value="No"/>
Storage Type	<input type="text" value="General Purpose (SSD)"/>
Allocated Storage*	<input type="text" value="10"/> GB



Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. [Click here](#) for more details.

Settings

DB Instance Identifier*	<input type="text" value="First_AWS_DB"/>
Master Username*	<input type="text" value="cloud_dba"/>
Master Password*	<input type="text" value="....."/>
Confirm Password*	<input type="text" value="..... "/>

Figure 3-29. AWS RDS: specify DB Details

The warning message in Figure 3-29 is due to the fact that storage performance in AWS is linked to its size. The bigger the size, the larger the I/O bandwidth.

You can choose from many DB engine versions on the offering, starting from 11.2.0.2 to 12c. Note that neither Oracle Cloud nor AWS RDS offers 10g or below versions, as shown in Figure 3-30.

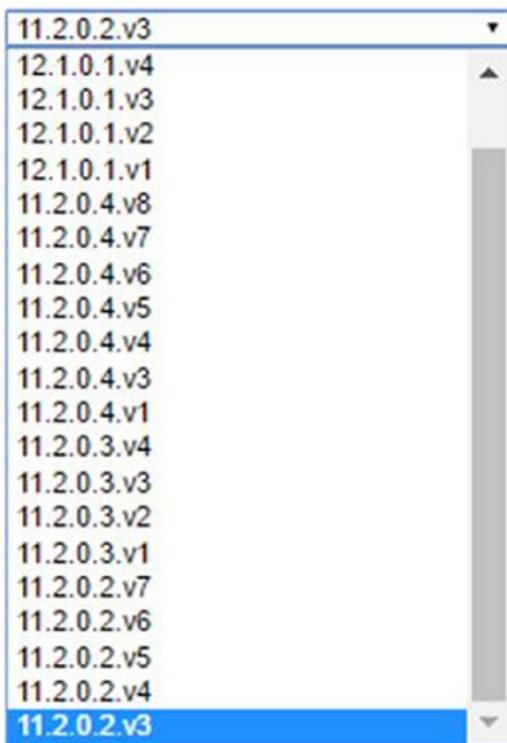
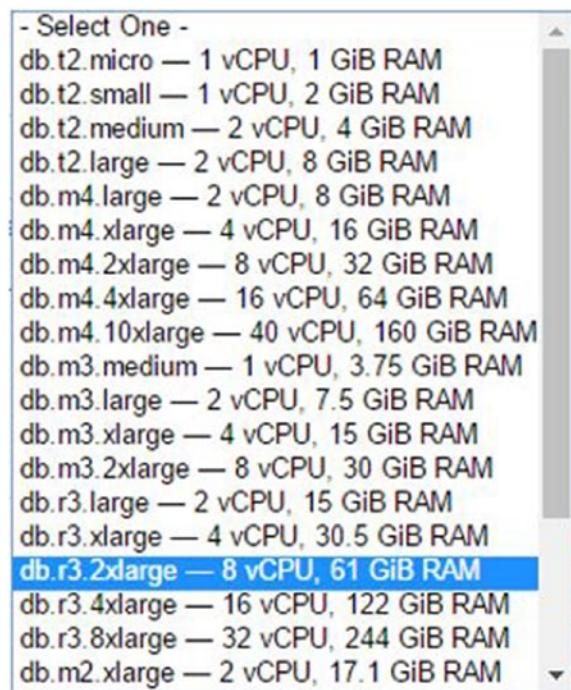


Figure 3-30. AWS RDS: DB version options

As shown in Figure 3-31, the DB Instance Class allows you to select the CPU and memory requirements for your instance. There are more than 10 options to choose from, based on the DB size categorization, including these:

- Micro
- Small
- Medium
- Large
- Extra Large

This is comparable to the shape option in Oracle Cloud.

**Figure 3-31.** AWS RDS: DB server options

Note that the DB instance identifier must contain only letters, numbers, and hyphens. Otherwise, you can expect the error message shown in Figure 3-32.

**Figure 3-32.** AWS RDS: DB instance name

Configuring the Advanced Settings

Now you are about to configure the advanced settings, where you can go with default virtual private cloud or let AWS create a new one for you. The public accessibility setting needs to be set to Yes or No, based on the connectivity and access requirements. You should select Yes if you want the public IP address to be assigned to your DB server so that it is accessible from outside of the VPC.

Select the database options and backup settings, which are self-explanatory, as shown in Figure 3-33.

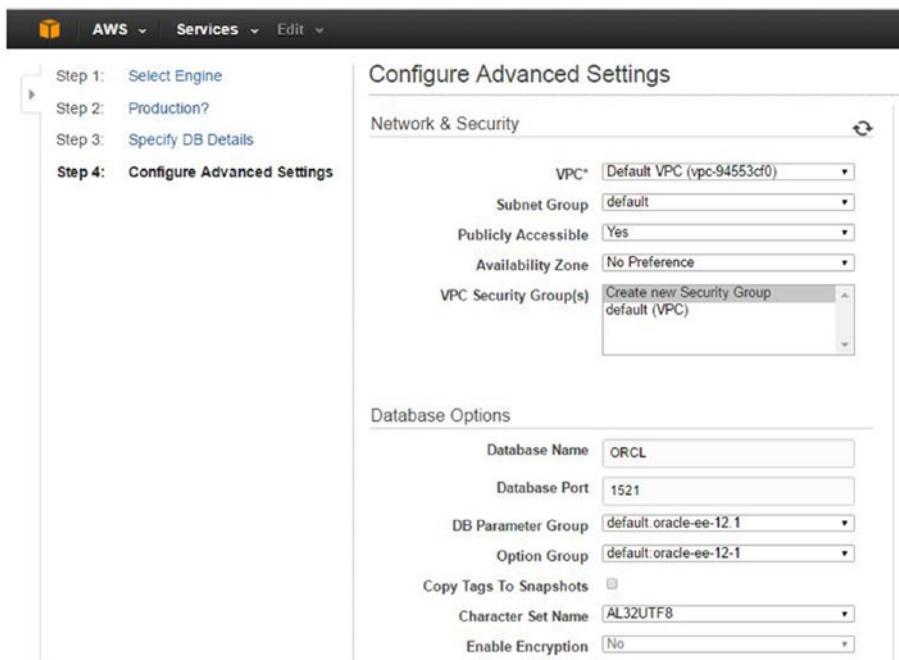


Figure 3-33. AWS RDS: Configure Advanced Settings

Once you submit the request for DB provisioning, the instance creation process will start, as shown in Figure 3-34.

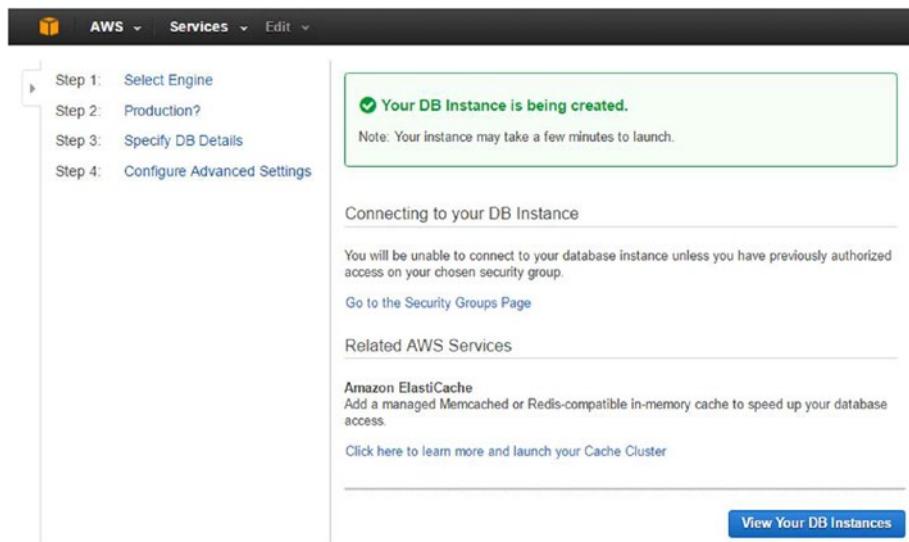


Figure 3-34. AWS RDS: DB instance creation

You can check the progress/status of your instance, as shown in Figure 3-35. Once it's ready for access, the status will be Available.

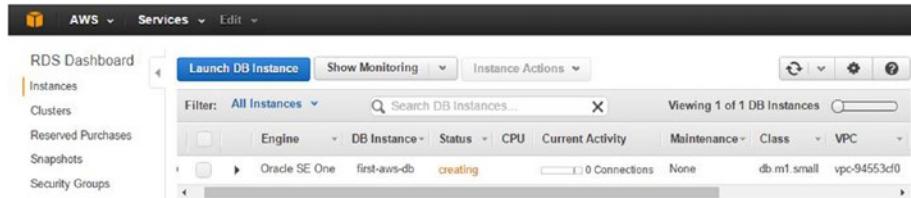


Figure 3-35. AWS RDS: DB instance creation status

After your instance is created, as shown in Figure 3-36, you can start monitoring it online or access it remotely using SQL Developer and other tools.

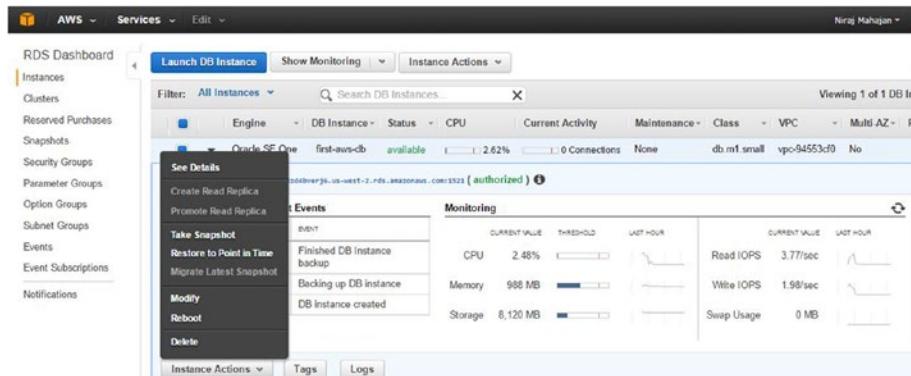


Figure 3-36. AWS RDS: DB instance availability page

Unlike Oracle Cloud, where you can start, stop, or restart your DB service, AWS RDS does not have an option to stop your DB instance. You can delete it, but you'll stop it and lose it forever.

As RDS is a managed service offering, there are lots of metrics provided by AWS in terms of ready-made monitoring, as shown in Figure 3-37. You can monitor CPU utilization, read-write operations, storage space, session count, free memory, and more.

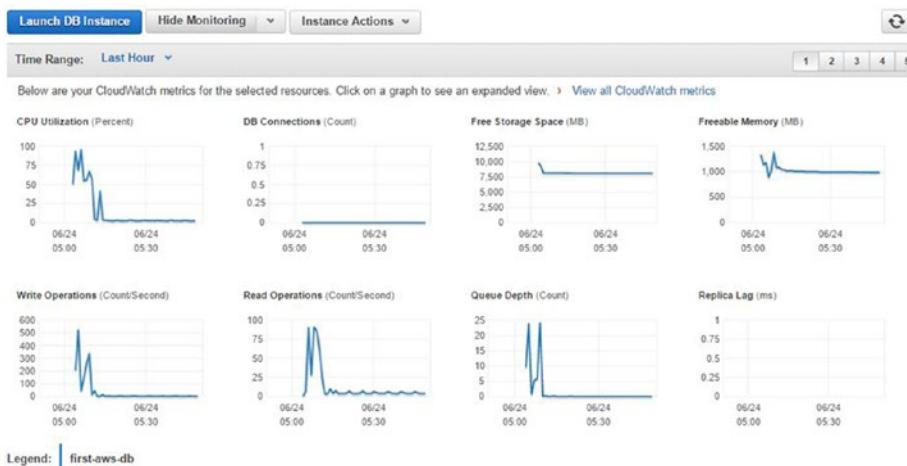


Figure 3-37. AWS RDS: DB instance monitoring, page 1

A few more metrics are shown in Figure 3-38.

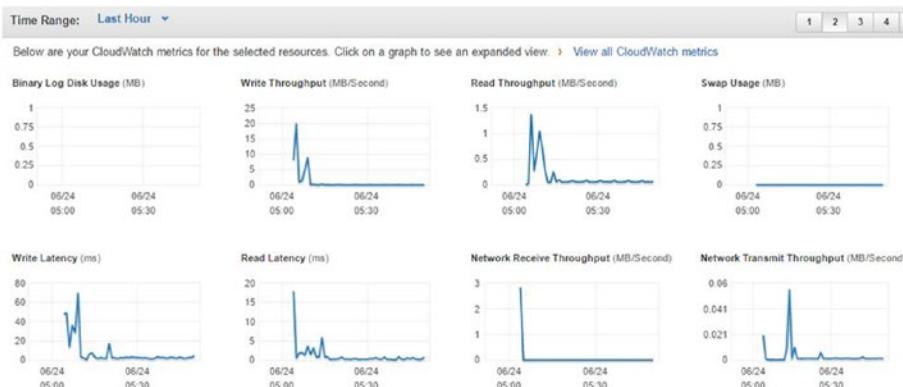


Figure 3-38. AWS RDS: DB instance monitoring, page 2

Accessing Your AWS RDS Instance

To access your newly created database instance remotely, its Publicly Accessible configuration should be set to Yes. You need to use the endpoint string (as shown in Figure 3-39) hostname when connecting using SQL Developer.

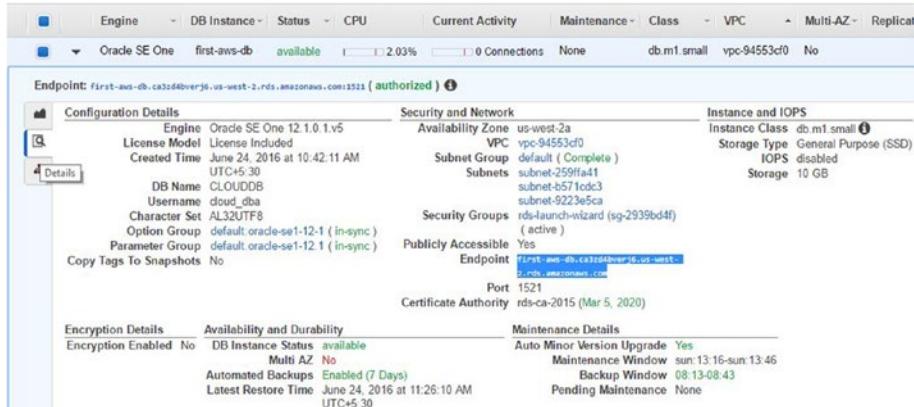


Figure 3-39. AWS RDS: DB instance endpoint

Additionally, you need to edit the inbound rules to allow specific IPs for the port where the listener services are running, as shown in Figure 3-40. Inbound settings can be opened using the following navigation path. Choose EC2 Dashboard ► Network & Security ► Security Groups and then click the Inbound tab on the lower half of the page.

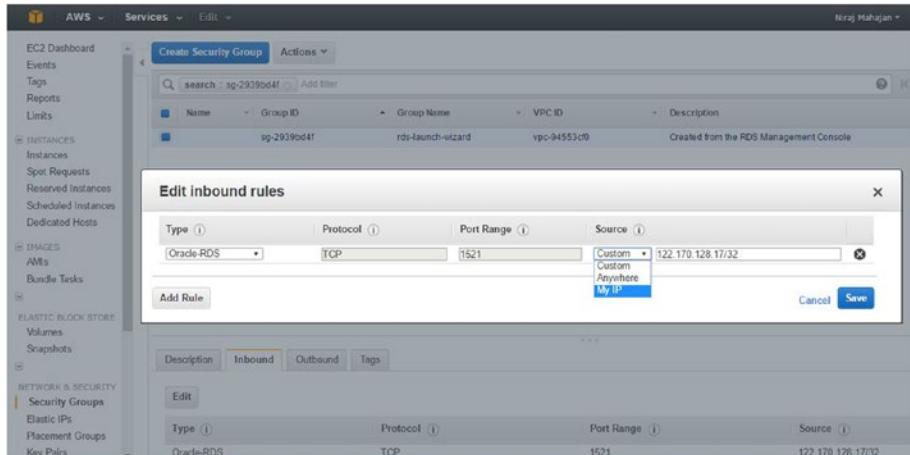


Figure 3-40. AWS RDS: inbound rule setting

After providing the correct information in the SQL Developer new connection option, as shown in Figure 3-41, you can test your connection to the instance at AWS RDS.

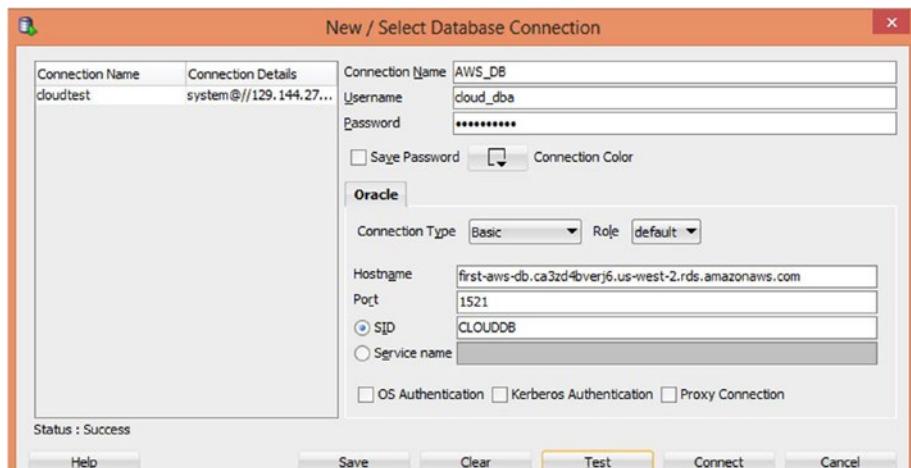


Figure 3-41. AWS RDS: connecting to the DB instance using SQL Developer

With the successful test, you can now access your DB as any other DB, as shown in Figure 3-42.

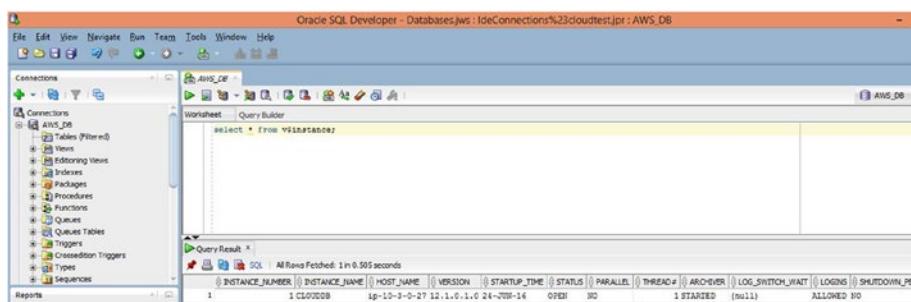
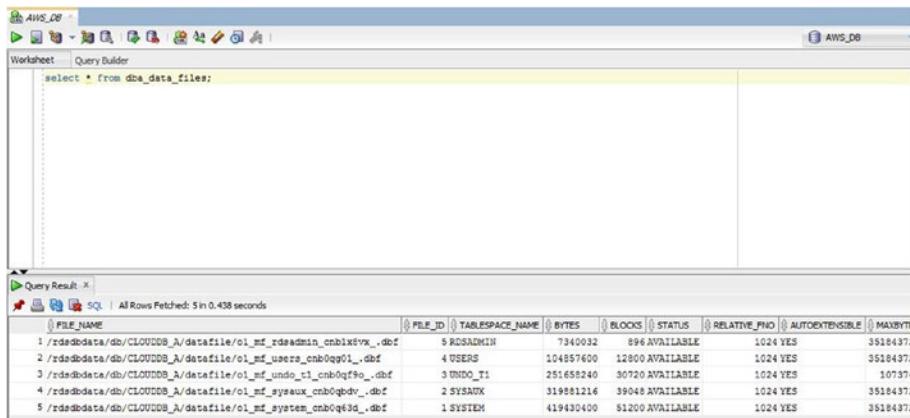


Figure 3-42. AWS RDS: DB instance query execution, example 1

You can even check the instance details and data files or create your own objects, as shown in Figure 3-43.



The screenshot shows a database management interface for an AWS RDS instance named 'AWS_DB'. In the 'Query Builder' tab, the following SQL query is entered:

```
select * from dba_data_files;
```

The results are displayed in a table titled 'Query Result' with the following data:

FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS	RELATIVE_FNO	AUTOEXTENSIBLE	MAXBYTE
1 /rdbddbdata/db/CLLOUDB_A/datafile/o1_mt_rdeadmin_cnbkx7v_.dbf	5	RSADMIN	7340032	896	AVAILABLE	1024	YES	35184371
2 /rdbddbdata/db/CLLOUDB_A/datafile/o1_mt_users_cnb0qg91_.dbf	4	USERS	104857600	12800	AVAILABLE	1024	YES	35184371
3 /rdbddbdata/db/CLLOUDB_A/datafile/o1_mt_undo_t1_cnb0qgf9o_.dbf	5	UND0_T1	251658240	30720	AVAILABLE	1024	YES	10374
4 /rdbddbdata/db/CLLOUDB_A/datafile/o1_mt_sysaux_cnb0qdv_.dbf	2	SYSAUX	319811216	39048	AVAILABLE	1024	YES	35184371
5 /rdbddbdata/db/CLLOUDB_A/datafile/o1_mt_system_cnb0q63d_.dbf	1	SYSTEM	419430400	51200	AVAILABLE	1024	YES	35184371

Figure 3-43. AWS RDS: DB instance query execution, example 2

Summary

In this chapter, you learned the step-by-step method for provisioning your DB instance with Oracle Cloud and with AWS RDS. This chapter covered details on Oracle Cloud subscription, selecting Oracle Database Cloud offering, database service creation, and ways to access your database. For database provisioning in AWS, you went through various engine options, DB environment specification, advanced configuration, and accessing your AWS RDS instance.

In subsequent chapters, you will learn about database administration tasks on databases provisioned in the cloud. The next chapter covers the high availability configuration for a database running in the cloud.

CHAPTER 4



High Availability Options

In this chapter, you will learn about the need of high availability (HA), including HA options for your database in Oracle Cloud and AWS, the details of setting up Oracle RAC in Oracle Cloud, connecting to Oracle RAC database server using PuTTY, connecting to Oracle RAC instances using SQL Developer, RAC HA testing, Multi-AZ deployment in AWS, and running failover tests in a Multi-AZ deployment.

Need for High Availability with Cloud-Based DBs

Cloud service providers have built-in high availability at every layer, including the hardware level, the network level, the storage level, and so on, so the obvious question that comes to one's mind is, why do we need to consider Oracle database high availability for our cloud databases?

Although the infrastructure provided by Oracle Cloud, AWS, and other cloud providers is highly available, it doesn't protect you from all kind of failures. For example, consider an availability zone failure in AWS, which would cause your DB to go down if your database is running in only one availability zone. Similarly, a region-level failure (admittedly a very unlikely event) would cause your DB to go down even if you have used a Multi-AZ deployment.

To protect from such failures and to keep your application available, you need to think about high availability (HA) and disaster recovery options. Choosing the right HA option is similar to what you do with your on-premise database, but HA options available in the cloud are not as broad as the on-premise options. In next sections of this chapter, you will go through all the available HA options. Let's first look at the database high availability overview.

Database High Availability Overview

Availability is usually defined in terms of accessibility of an application. In today's world, applications have become global and are being accessed from around the world using mobile connections or the Internet. This scenario brings the need for 24x7x365 application availability. Non-availability (usually referred to as *downtime*) of an application causes loss of revenue and unhappy customers. Since your database is an integral part of any application, it needs to be highly available. To make your database highly available, you need to know the causes of downtime and then define HA architectures to handle all such downtime situations.

The following are key terms related to high availability:

- *Recovery Time Objective (RTO)*: Specifies the time that is allowed for recovery. Higher RTO is good for DBAs but critical businesses require near zero or zero RTO by using Active-Active configuration across datacenters.
- *Recovery Point Objective (RPO)*: This specifies the amount of data loss that is allowed. For mission-critical financial systems, for example, an RPO value would be zero. However, for systems such as business warehouses, a few hours of data loss could be manageable since data can be reloaded from the source.

Oracle database has several HA options and along with this, there is a Maximum Availability Architecture (MAA) framework that provides you with the blueprint of various HA architectures. Oracle provides the following HA options. The DataGuard option is a DR option, but it can also be used as an HA option to handle site failures.

- OS-level cluster
- Oracle clusterware (Cold Failover Cluster)
- Oracle Real Application Clusters (Oracle RAC)
- Oracle RAC One Node
- Oracle RAC on Extended Clusters
- DataGuard
- Oracle Clusterware and DataGuard
- Oracle RAC and DataGuard
- Oracle Streams
- Oracle Golden Gate

The method for choosing the right HA option is out of the scope of this book, as our focus is to understand HA from the cloud perspective. However, the following are key considerations for choosing the right HA option:

- RTO and RPO requirements around unplanned outages and planned maintenance
- Total cost of ownership (TCO)
- Enterprise architecture and IT architecture strategy
- Return on investment (ROI)

In the next sections, you will read about the HA options that are available in Oracle Cloud and AWS.

HA Options in Oracle Cloud

Higher RTO, rolling upgrades and patching, transparent application failovers, and many more features make Oracle RAC, one of the best HA options for Oracle database. Oracle RAC is available in Oracle Cloud.

The following section describes the step-by-step process for provisioning Oracle RAC database in Oracle Cloud.

Oracle RAC Provisioning in Oracle Cloud

As of the writing of this book, Oracle supports RAC on a two-node cluster only. There are a few specific requirements related to software edition and shape, which are explained in the provisioning steps. At very high level, there are nine steps:

1. Log in to Oracle Database cloud service to launch the service console.
2. Click on Create Service.
3. Select the service level.
4. Select the software release.
5. Select the software edition.
6. Select the database clustering with RAC option.
7. Provide details for the service configuration and database configuration.
8. Verify the details and click on Create.
9. Check the RAC DB creation status.

Step 1: Log in to Oracle Database Cloud service and launch the service console by providing your domain, username, and password (Figure 4-1).

<https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jspx>

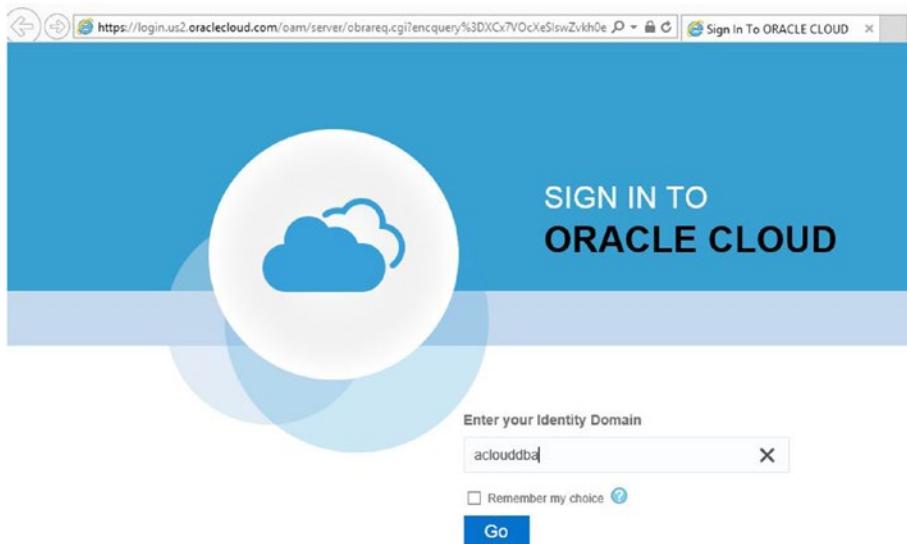


Figure 4-1. RAC provisioning in Oracle Cloud: login page

Step 2: Click on Services and choose the Create Service option (Figure 4-2).

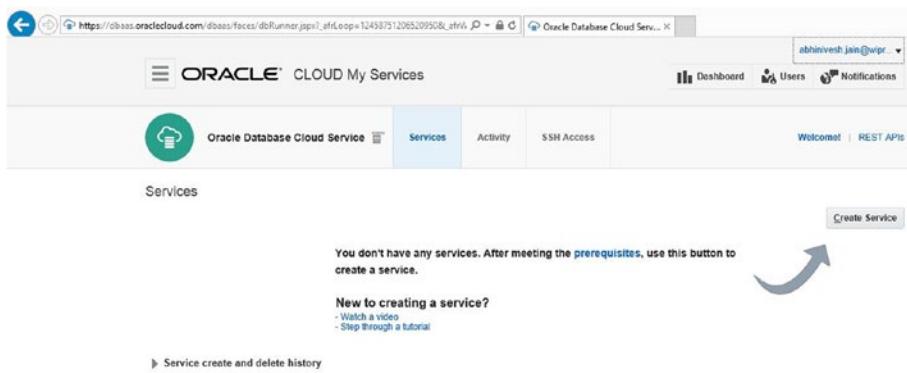


Figure 4-2. RAC provisioning in Oracle Cloud: create service

Step 3: Select Oracle database cloud service level and click on Next (Figure 4-3).

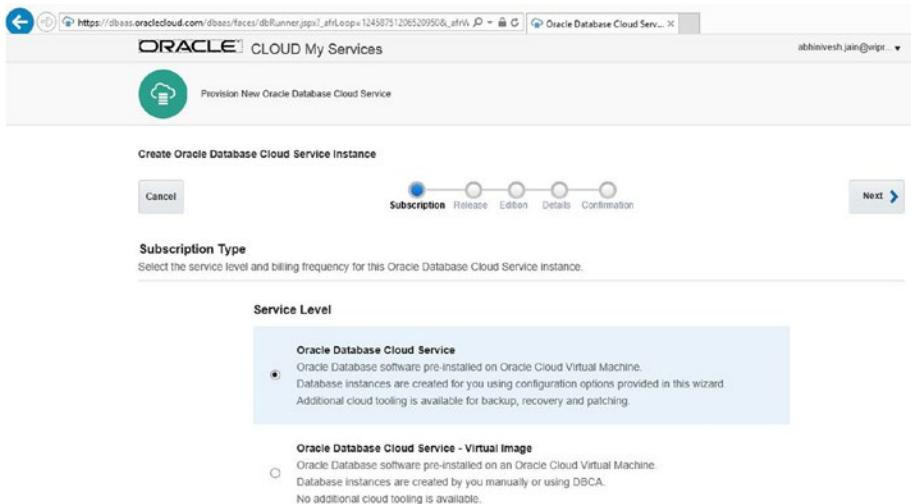


Figure 4-3. RAC provisioning in Oracle Cloud: select the service level

Step 4: Select Oracle database 12c release1 and click Next (Figure 4-4). Oracle RAC option is only available in Oracle 12c, hence you can't select Oracle database 11g.

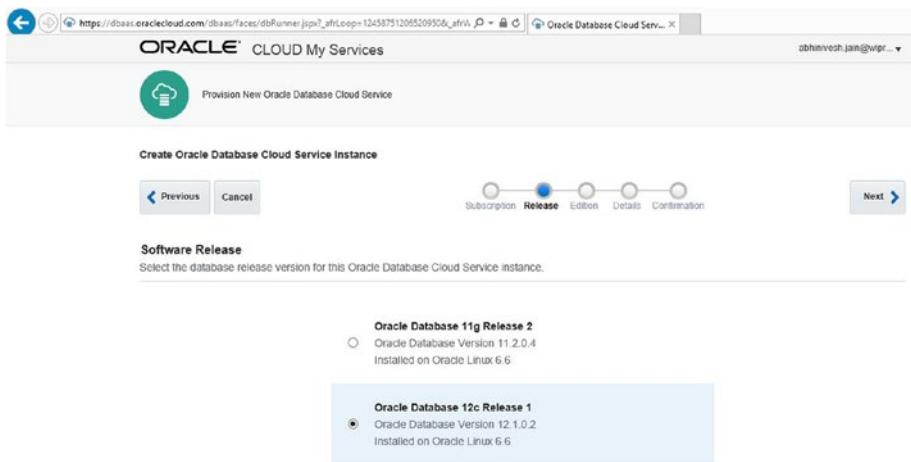


Figure 4-4. RAC provisioning in Oracle Cloud: select the software release

Step 5: Select Enterprise Edition Extreme performance option (Figure 4-5), since RAC database is only available for this software edition type.

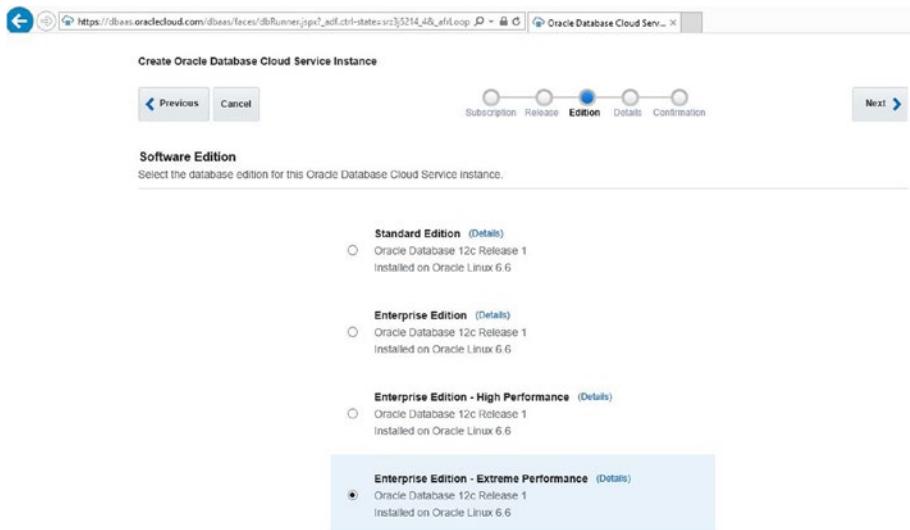


Figure 4-5. RAC provisioning in Oracle Cloud: select the software edition

Step 6: Select the Database Clustering with RAC option, which is present under Database configuration on the Service Details page. Provide details for the SSH public key (that can be generated using PuTTYGen software by clicking on Generate and saving the key files) and for an administration password (Figure 4-6). Other parameters can be selected based on your requirements. For demo purposes, all the default values are taken along with the No Backup option. You can refer to Chapter 3, “Database Provisioning,” to learn more about these options.

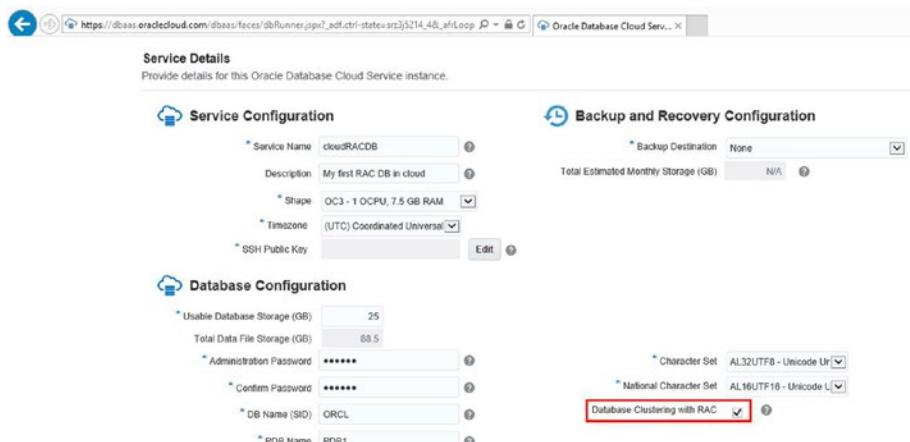


Figure 4-6. RAC provisioning in Oracle Cloud: select the RAC option

Step 7: If you are using a trial account, select Shape with two OCPUs; otherwise you will get the error as shown in Figure 4-7. If you are provisioning using a non-trial account, select the OCPU based on your requirements.

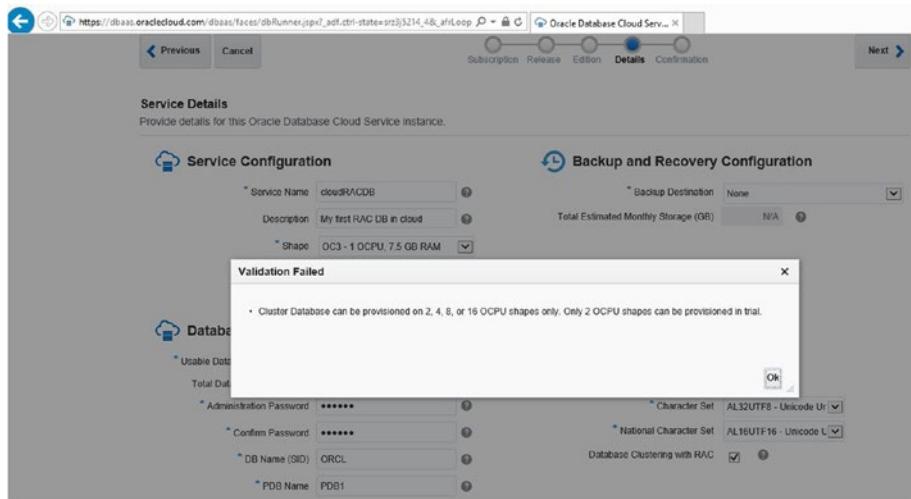


Figure 4-7. RAC provisioning in Oracle Cloud: shape related validation failure

Step 8: Verify all the inputs in the confirmation section and, if everything looks good, click on Create (Figure 4-8).

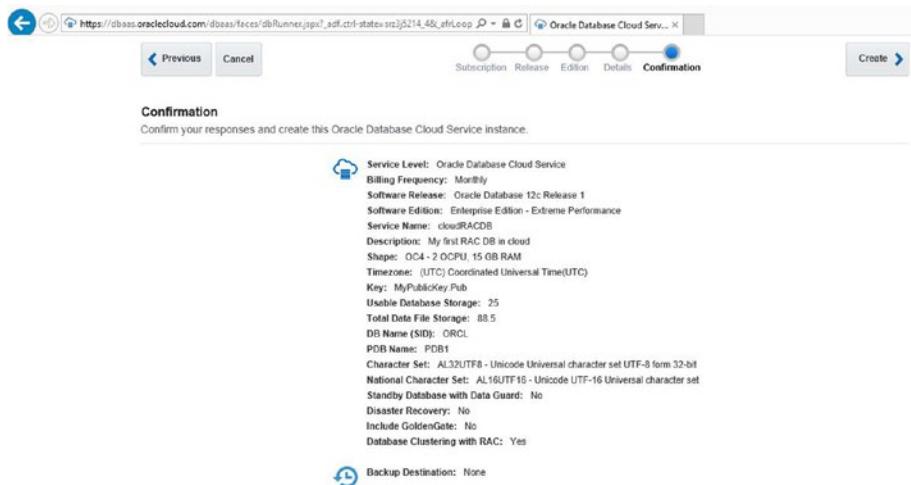


Figure 4-8. RAC provisioning in Oracle Cloud: confirmation

Step 9: After you submit the service creation request, you will be redirected to the dashboard (Figure 4-9) and you can refresh this page to check the status of your new RAC DB.

The screenshot shows the Oracle Database Cloud My Services dashboard. At the top, there are summary statistics: 1 Service, 4 OCPUs, 30 GB Memory, 256 GB Storage, and 2 Public IPs. Below this, under the 'Services' section, a search bar shows 'cloudRACDB'. A card for 'cloudRACDB' is displayed, indicating it was submitted on Sep 21, 2016, at 9:00:08 AM UTC. The status is 'In Progress'. The configuration includes 4 OCPUs, 30 GB Memory, and 256 GB Storage. Edition: Enterprise Edition - Extreme Performance. A link to 'Service create and delete history' is also present.

Figure 4-9. RAC provisioning in Oracle Cloud: build progress status

With the given configuration options, it took around 70 minutes to complete the RAC DB setup. In this time, Oracle cloud has set up the GRID infrastructure, ASM, RAC software, and RAC database. You can check this by connecting to your newly created RAC database.

To manage this RAC database, Oracle has provided the `raccl` utility. Be sure to note the public IP of these nodes (Figure 4-10) for connection using PuTTY.

The screenshot shows the Oracle Database Cloud Service dashboard for the 'cloudRACDB' service. The 'Overview' section indicates 2 Nodes. The 'Nodes' section lists two nodes: 'cloudracdb1' (Public IP: 129.152.151.234, SID: ORCL1, PDB Name: PDB1) and 'cloudracdb2' (Public IP: 129.152.151.70, SID: ORCL2, PDB Name: PDB1). Both nodes have 2 OCPUs, 15 GB Memory, and 95 GB Storage. The 'Additional Information' section provides details like Identity Domain: adcloudba, Edition: Enterprise Edition - Extreme Performance, Service Level: Oracle Database Cloud Service, and Subscription Type: Monthly.

Figure 4-10. RAC provisioning in Oracle Cloud: configuration details of RAC DB

Connecting to Your RAC Database Servers Using PuTTY

After the previous steps, you are all set to start using your RAC database. Figure 4-11 shows how you can connect using PuTTY. Remember to give the private key file location under SSH > Auth.

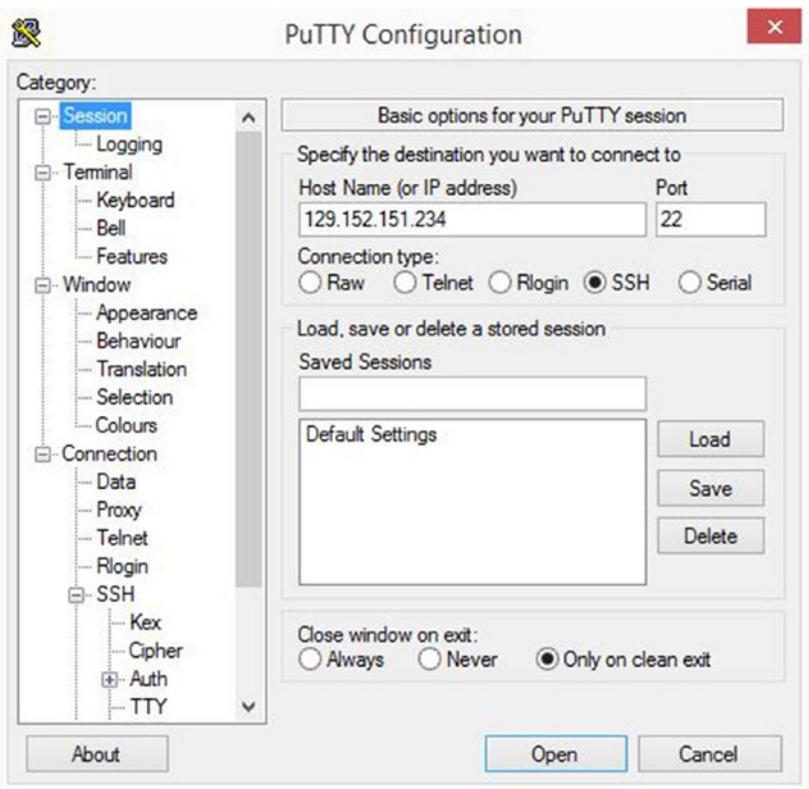


Figure 4-11. RAC provisioning in Oracle Cloud: connecting to RAC DB

After connecting, you can run the `raccli` commands (Figure 4-12) to see the RAC system details. The `raccli` utility is covered in detail in the “Manage and Monitor” chapter (Chapter 9).

```
[opc@cloudracdb1 ~]$ raccli describe system  
  
{  
  "serviceName" : "cloudracdb",  
  "nodes" : [ {  
    "nodeName" : "cloudracdb1",  
    "components" : [ {  
      "componentType" : "GridHome",  
      "componentName" : "OraGrid12102",  
      "componentVersion" : "12.1.0.2.160719"  
    }, {  
      "componentType" : "DatabaseHome",  
      "componentName" : "OraDB12102_home1",  
      "componentVersion" : "12.1.0.2.160719"  
    }, {  
      "componentType" : "RDK",  
      "componentName" : "RDK",  
      "componentVersion" : "16.4.3.0.0"  
    } ]  
  }, {  
    "nodeName" : "cloudracdb2",  
    "components" : [ {  
      "componentType" : "GridHome",  
      "componentName" : "OraGrid12102",  
      "componentVersion" : "12.1.0.2.160719"  
    }, {  
      "componentType" : "DatabaseHome",  
      "componentName" : "OraDB12102_home1",  
      "componentVersion" : "12.1.0.2.160719"  
    }, {  
      "componentType" : "RDK",  
      "componentName" : "RDK",  
      "componentVersion" : "16.4.3.0.0"  
    } ]  
  }]
```

Figure 4-12. RAC provisioning in Oracle Cloud: RAC configuration details

Connecting to Your RAC Database Using SQL Developer

To connect to the RAC database using SQL Developer, you need to first enable the Access rules for a scan listener; otherwise, you will get the error as shown in Figure 4-13.

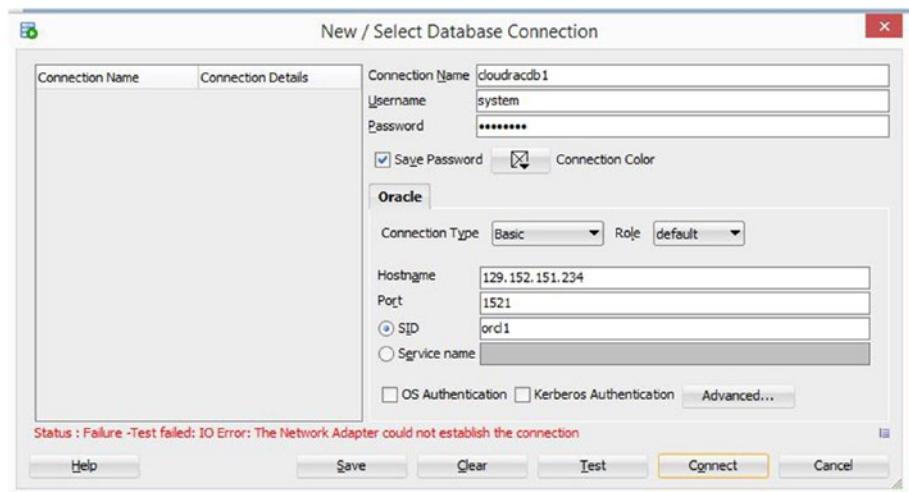


Figure 4-13. RAC provisioning in Oracle Cloud: RAC connection error

To fix this error, go to Access Rules and enable them. Click on the Action tab and select Access Rules (Figure 4-14).

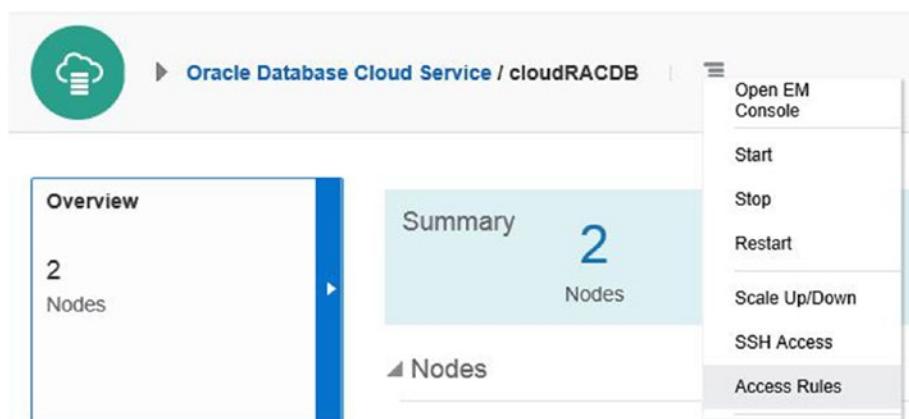


Figure 4-14. RAC provisioning in Oracle Cloud: the access rules

Now for the ora_p2_scan_listener rule, click on Action and select Enable. See Figure 4-15 for details.

Access Rules						Create Rule
You can use access rules to control network access to service components. On this page, you can manage your access rules.						
Results per page: 10						8 result(s) as of Sep 22, 2016 4:14:32 AM UTC
Status	Rule Name	Source	Destination	Ports	Description	Rule Type
✗	ora_p2_db_listener	PUBLIC-INTERNET	DB	1522	Permit access for Oracle listener	DEFAULT
✗	ora_p2_scan_listener	PUBLIC-INTERNET	DB	1521	Permit access for SCAN listener	DEFAULT (Enable)
✗	ora_p2_monitor_12c	PUBLIC-INTERNET	DB	5500	Permit access to EM Express	Disable
✗	ora_p2_monitor_11g	PUBLIC-INTERNET	DB	1158	Permit access for Database Console	Delete
✗	ora_p2_ons	PUBLIC-INTERNET	DB	6200	Permit access for RAC ONS	DEFAULT
✗	ora_p2_ssh	PUBLIC-INTERNET	DB	22	Permit ssh access to nodes	DEFAULT
✗	ora_trusted_hosts_scan_listener	127.0.0.1/32	DB	1521	DO NOT MODIFY: A securle to allow specific IPs to connect to this db	SYSTEM
✗	ora_trusted_hosts_db_listener	127.0.0.1/32	DB	1522	DO NOT MODIFY: A securle to allow specific IPs to connect to this db	SYSTEM

Figure 4-15. RAC provisioning in Oracle Cloud: enable the access rules

After this change, the DB connection will go through and you can see that both instances are running fine (Figure 4-16).

INST_ID	INSTANCE_NUMBER	INSTANCE_NAME	HOST_NAME	VERSION	STARTUP_TIME	STATUS	PARALLEL	THREAD#	ALO_GRANTED	LOG_SWITCH_WAIT
1	1	1 ora01	cloudraadb1	12.1.0.2.0	21-SEP-16	OPEN	YES	1 STARTED	(null)	
2	2	2 ora02	cloudraadb2	12.1.0.2.0	21-SEP-16	OPEN	YES	2 STARTED	(null)	

Figure 4-16. RAC provisioning in Oracle Cloud: output from SQL Developer

If you want to connect to a second DB instance that is running on another node, you can give the inputs shown in Figure 4-17. You need to give the public IP address of Node 2 and change ORACLE SID to Oracle instance ID of the second node. This is shown in Figure 4-17.

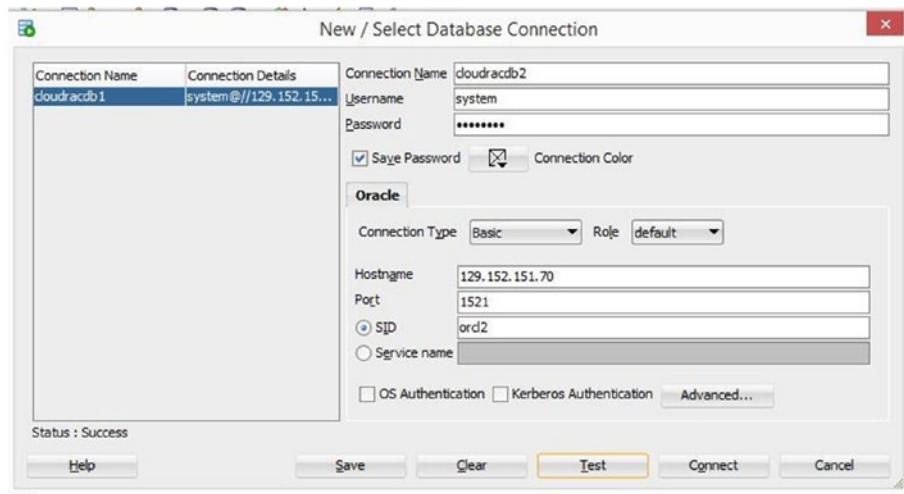


Figure 4-17. RAC provisioning in Oracle Cloud: SQL Developer connection to the second instance

A query output (Figure 4-18) from v\$instance shows that you are on the second node.

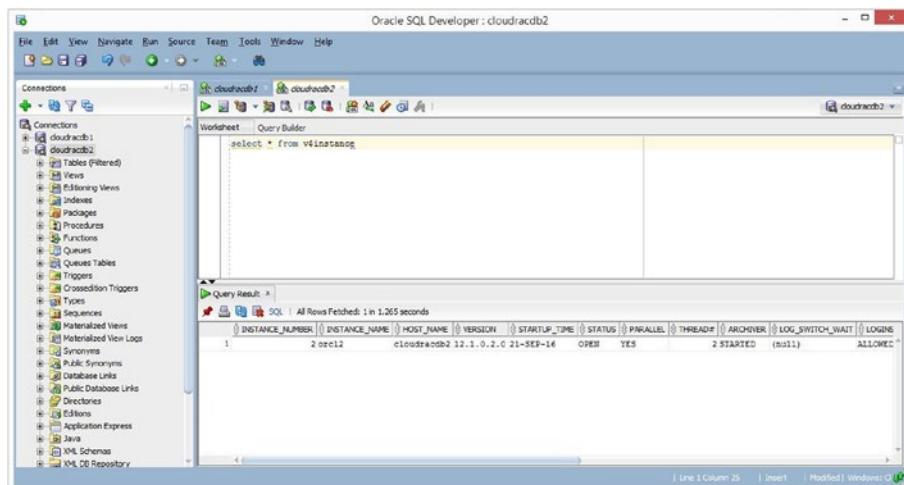


Figure 4-18. RAC provisioning in Oracle Cloud: query output from the second instance

RAC HA Testing

Now let's see how RAC high availability works in Oracle Cloud. For simulating this, you can shut down Node 1 and see if you can still connect to your database from Node 2. Figure 4-19 shows that only instance two is running and shutting down Node 1 didn't cause any issue in DB availability.

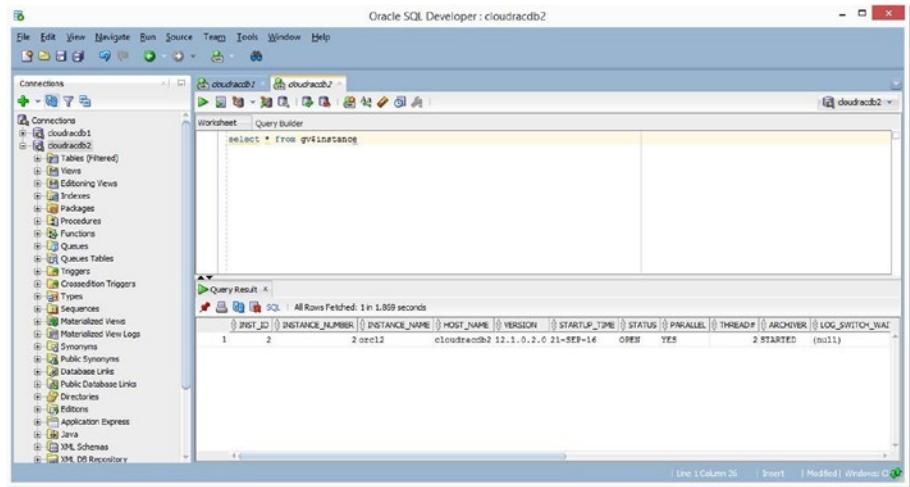


Figure 4-19. RAC provisioning in Oracle Cloud: RAC HA testing

HA Options in AWS

In AWS, you have Oracle RAC (in IaaS) and Multi-AZ deployment (in Amazon RDS) options to make your database highly available.

This section includes Oracle RAC provisioning overview. You will also learn how to set up Oracle in Multi-AZ deployment.

Overview of Oracle RAC Provisioning in AWS

As of the writing this book, ORACLE RAC is not officially supported in Amazon RDS. It is mentioned in <https://aws.amazon.com/rds/faqs/>.

However, Oracle RAC can be set up in IaaS (using an EC2 instance). RAC provisioning in AWS is a very complex process with several manual steps. Oracle doesn't certify and support this. You can refer to the "Oracle Real Application Clusters (RAC) Support on Third Party Clouds" article to get more details on Oracle's stand on this. See <http://www.oracle.com/technetwork/database/options/clustering/overview/rac-cloud-support-2843861.pdf>.

Oracle RAC setup steps are documented in the AWS article, “Deploying Scalable Oracle RAC on Amazon EC2”. You can search for this article in <https://aws.amazon.com/articles/>. This article is revised from time to time, so it is a good idea to look for the latest version of this article. At the end of this article, there are FAQs that cover key questions related to this kind of setup. You might want to start by reading this section first.

For this RAC setup, there is one free, preconfigured compute node (EC2 AMI) available in the AWS marketplace. You can search for it, as shown in Figure 4-20. This AMI is for Oracle 12c RAC and it uses Linux 7.1 OS. If you want to build RAC with an older Oracle version, you have to build all the components on your own, which are pre-built in this AMI.

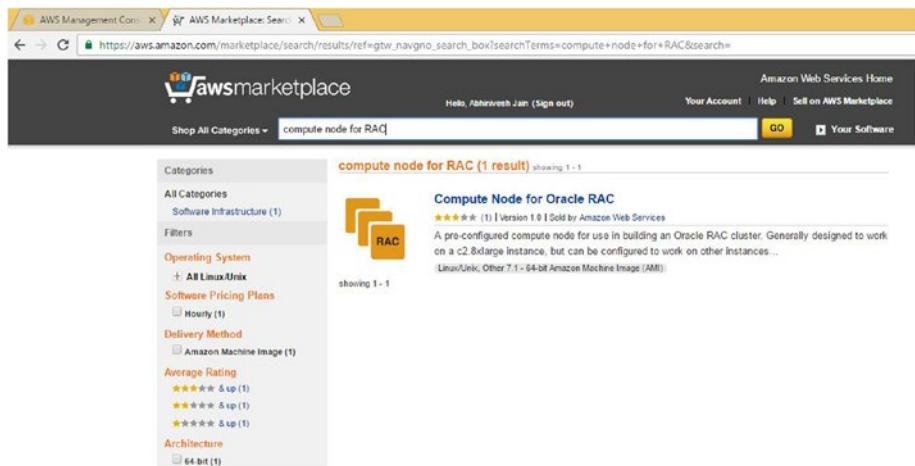


Figure 4-20. RAC node AMI in the AWS marketplace

Note If you are planning to do this deployment using a free tier account, remember that there are many chargeable components, like Route53, used here. Besides this, you need Oracle Enterprise Edition license.

Due to its complexity, questionable usefulness, and lack of Oracle support, this setup option is not covered in this book. It is not recommended to go for this option.

Overview of the Multi-AZ Deployment Option

AWS has a feature called Multiple Availability Deployment for Oracle RDS. In this deployment option, you can deploy your primary database instance in one availability zone and standby database instance in another availability zone, all in the same region. These DB instances utilize synchronous replication. Automatic failover occurs during planned maintenance, primary DB failure, or availability zone failure situations. This is a snapshot-based replication and not related to DataGuard.

There might be performance issues in writing and committing latency due to synchronous replication, hence it is recommended you use provisioned IOPS and higher DB classes.

Multi-AZ Deployment Method

Implementation of the Multi-AZ deployment is a very simple, one-click operation. You can specify this during the instance provisioning or can do this change afterward. This can also be done using APIs.

Figure 4-21 shows the option for selecting Multi-AZ deployment during instance provisioning.

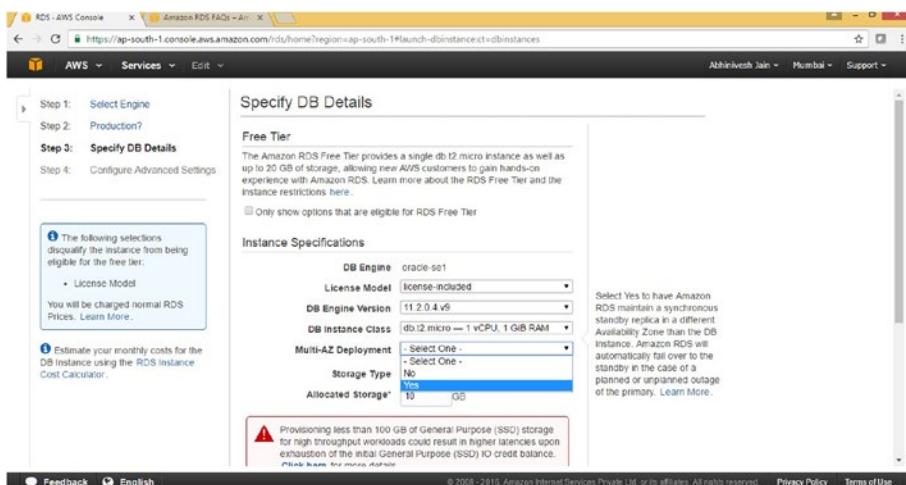


Figure 4-21. Multi-AZ deployment option

After selecting the Multi-AZ deployment, you provide a database name, a username, and a password, as shown in Figure 4-22.

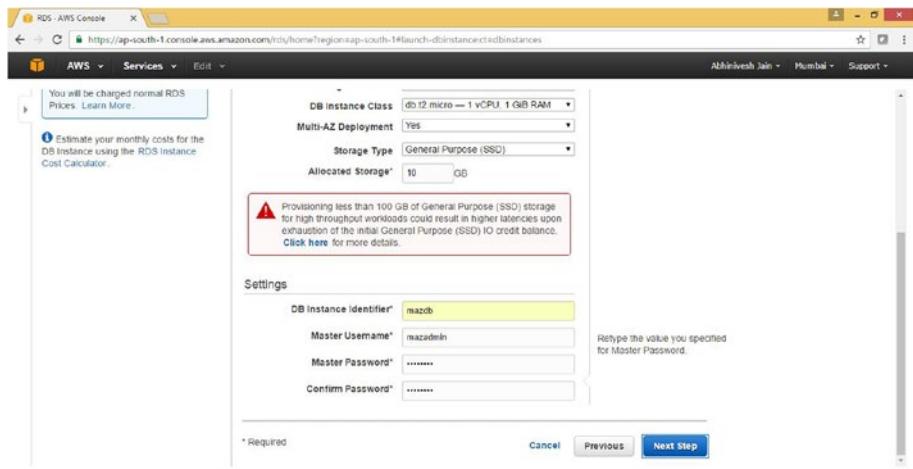


Figure 4-22. Multi-AZ deployment, step 1

Note The Multi-AZ deployment option is not available with the free tier option, hence it will be charged as per the RDS pricing.

Leave all the advanced settings as default and click on Launch DB Instance (Figure 4-23). You can refer to Chapter 3, “Database Provisioning,” to get more details on these advanced settings.

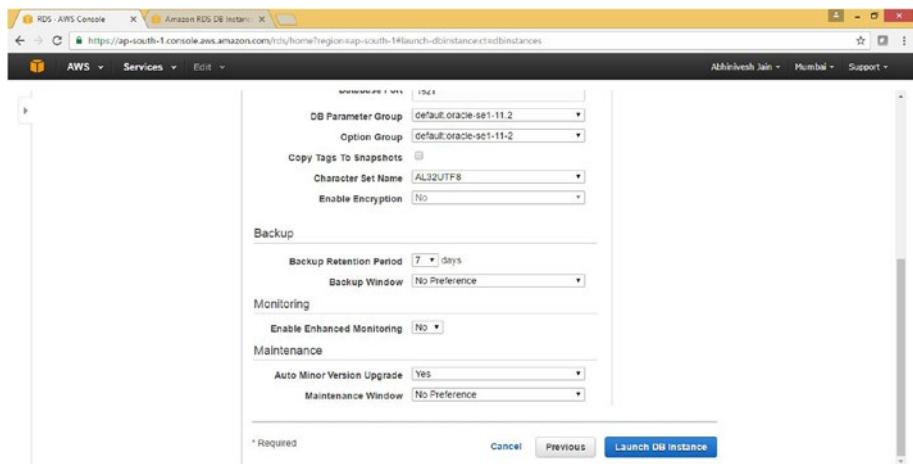


Figure 4-23. Multi-AZ deployment, STEP 2

CHAPTER 4 ■ HIGH AVAILABILITY OPTIONS

After a successful request submission, you will see a message similar to the one in Figure 4-24.

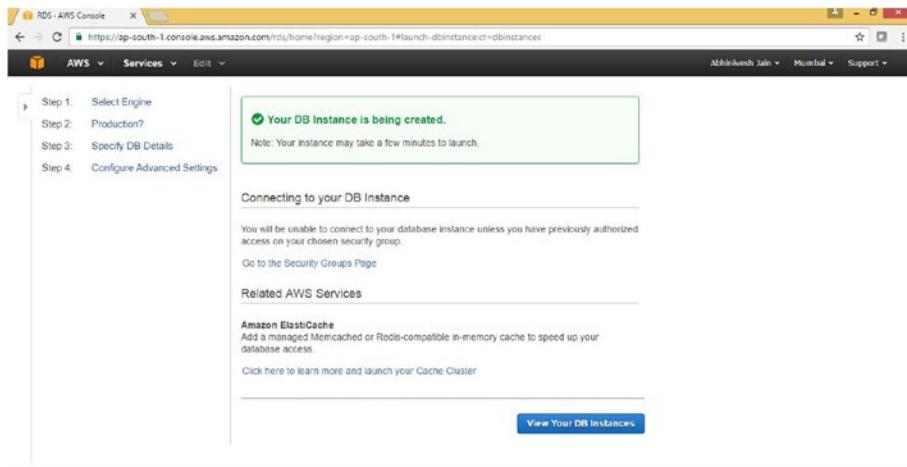


Figure 4-24. Multi-AZ deployment, step 3

After a few minutes, you can see the DB instances provisioned in Multi-AZ deployment. It is running in the ap-south-1b availability zone. See Figure 4-25, which shows the completed Multi-AZ deployment.

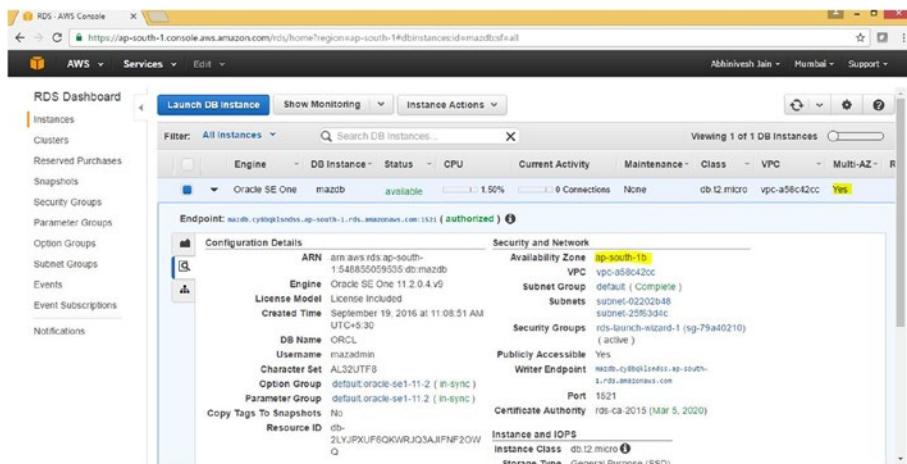


Figure 4-25. Completed multi-AZ deployment

Failover in Multi-AZ Deployment

Amazon RDS does automatic failover in case of any failure of the primary DB. It also changes the DNS record of the DB instance and points it to a standby instance. Hence, you don't need any changes at the application end to connect to the standby instance.

Automatic failover occurs in the following conditions:

- Availability zone outage
- DB instance failure
- OS patching of DB server
- DB server type change

You have the option of performing a manual failover by doing the DB reboot using the failover option. To perform a manual failover, click on Instance Actions ► Reboot, as shown in Figure 4-26.

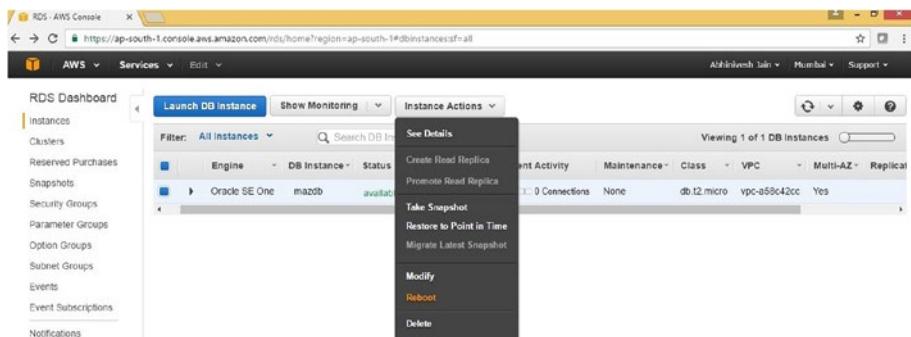


Figure 4-26. Multi-AZ deployment, failover step 1

Now select the Reboot with Failover option and click on Reboot (Figure 4-27).

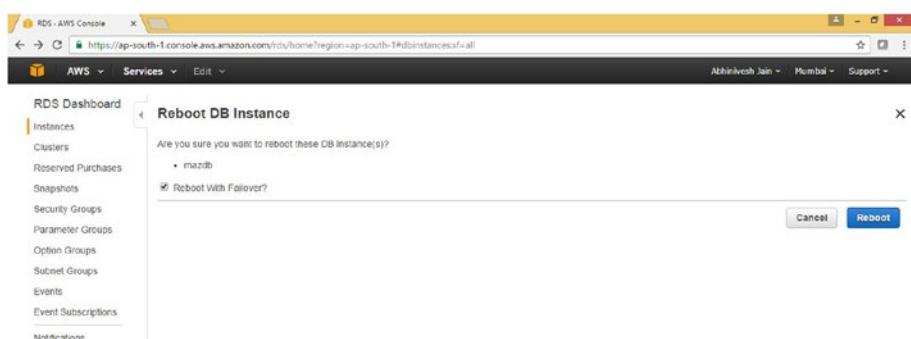


Figure 4-27. Multi-AZ deployment, failover step 2

It takes 1-2 minutes for failover and after that, your DB server is available on another availability zone. As you can see from Figure 4-28, the DB was previously running in the ap-south-1b availability zone. After failover, it is running in the ap-south-1a availability zone (Figure 4-28).

The screenshot shows the AWS RDS console with the URL <https://ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#dbInstances?all>. The main dashboard lists various resources like Instances, Clusters, and Parameter Groups. On the right, a detailed view of a database instance is shown.

Configuration Details:

- ARN:** arn:aws:rds:ap-south-1:154885059355:db:mazdb
- Engine:** Oracle SE One 11.2.0.4.v5
- License Model:** License Included
- Created Time:** September 19, 2016 at 11:08:51 AM UTC+9:30
- DB Name:** ORCL
- Username:** mazadmin
- Character Set:** AL32UTF8
- Option Group:** default.oracle-se1-11.2 (in-sync)
- Parameter Group:** default.oracle-se1-11.2 (in-sync)
- Resource ID:** db-2LYJFUXFSQKVRJQ3AJFNF2OWQ

Security and Network:

- Availability Zone:** ap-south-1a (highlighted)
- VPC:** vpc-a58c42cc
- Subnet Group:** default (Complete)
- Subnets:** subnet-02202d48, subnet-259334c
- Security Groups:** rds-launch-wizard-1 (sg-79a40210) (active)
- Publicly Accessible:** Yes
- Writer Endpoint:** mazdb.cy0ok1sed31.ap-south-1.rds.amazonaws.com
- Port:** 1521
- Certificate Authority:** rds-ca-2015 (Mar 5, 2020)

Instance and IOPS:

- Instance Class:** db.t2.micro (highlighted)
- Storage Type:** General Purpose (SSD)

Figure 4-28. Multi-AZ deployment: failover completion

Failover time depends on the database activity such as in-progress large transaction and other conditions. You can find out about automatic failovers using the “alarms and recent events” notification or by using RDS event notifications/APIs.

Summary

This chapter explained the need of high availability (HA), defined terms like RPO and RTO, and discussed the currently available HA options for your database in Oracle Cloud and AWS. You went through the step-by-step method for setting up Oracle RAC in Oracle Cloud, connecting to Oracle RAC database server using PuTTY, and connecting Oracle RAC instances using SQL Developer. After you were connected, you learned how to test the RAC high availability feature.

While exploring HA options available in AWS, you learned the basics of Multi-AZ deployment in AWS and learned the step-by-step process for setting up DB in Multi-AZ deployment. In the end, you learned about database failover testing in Multi-AZ deployment.

In the next chapter, you go through the various database disaster recovery options.

CHAPTER 5



Disaster Recovery Options

In this chapter, you first learn why Disaster Recovery (DR) is required for cloud-based databases. You will learn about Oracle DataGuard setup, along with DataGuard switchover, failover, and reinstating failed primary operations in Oracle Cloud. In the end, you learn how to perform some of the DataGuard-related operations using dbaascli and a brief overview of DR options available in AWS.

Need for Disaster Recovery with Cloud-Based DBs

Cloud service providers have built-in high availability, so the obvious question that comes to mind is, why do we need to consider Oracle database disaster recovery for a cloud database?

Although the infrastructure provided by Oracle Cloud, AWS, and other cloud provider is highly available, it doesn't protect you from all kind of failures. For example, an availability zone failure in AWS will cause your DB to go down if your database is running in only one availability zone, and a region-level failure (which is a very unlikely event) will cause your DB to go down even if you used multi-AZ deployment.

To protect from such failures and to keep your application available, you need to think about Disaster Recovery (DR) options. Choosing the right DR option is a similar exercise as to what you do for your on-premise database, but DR options available in the cloud are not as broad as the on-premise options. In the next sections, you go through all the available DR options. Let's first look at the database DR overview.

Database Disaster Recovery (DR) Overview

The cloud is entirely built on commodity hardware, hence failures are likely there. These failures could be as big as a site failure, where the entire datacenter/availability zone/region is unavailable, or the failure could be related to the DB server compute node failure or storage failure. Based on the criticality of your database, you can choose to have DR options for your database. This kind of setup provides you protection from these disaster scenarios and many more failures, such as data corruptions.

The method of choosing the right DR option is out of the scope of this book, as the focus is to understand DR from a cloud perspective. However, the following are key considerations for choosing the right DR option:

- RTO and RPO requirements around unplanned outages and planned maintenance
- Total cost of ownership (TCO)
- Enterprise architecture and IT architecture strategy
- Return on investment (ROI)

DR Options in Oracle Cloud

DataGuard is an effective DR solution. Oracle Cloud can be used in two ways to provision your standby database:

- Primary on-premise, DR in Oracle Cloud
- Primary and DR both in Oracle Cloud

The first option, using Oracle cloud as a DR, has the obvious benefits related to provisioning and maintaining a DR site, but it has its own challenges and complexities related to network latency, administration, and performance. You can use the Active DataGuard feature to effectively utilize the primary and DR databases.

At a high level, the following steps are performed to do this kind of configuration:

1. Create Oracle instance in Oracle Cloud.
2. Perform network configuration to enable on-premise DB connectivity with Cloud DB.
3. Drop the database created as part of step 1.
4. Instantiate a standby database using primary DB backup.
5. Configure the standby database.

How to take a backup of on-premise databases to the cloud is discussed in Chapter 8, “Backup and Restore”. With the details provided there, creating a standby database can be performed as per these high-level steps.

In the second option, Oracle Cloud provides a fully automated setup of your primary and standby databases. Here, your primary and DR databases both reside in Oracle Cloud. The DataGuard provisioning steps are explained in the next section.

As part of the DataGuard configuration, a single physical standby database is set up in another independent compute node. Oracle’s DataGuard option is provided for the software releases and editions shown in Table 5-1.

Table 5-1. Software Releases and Editions

Software Release	Edition
Oracle Database 11g R2 (11.2.0.4)	Enterprise Edition, Enterprise Edition- High performance, Enterprise Edition-Extreme performance
Oracle Database 12c R1 (12.1.0.2)	Enterprise Edition, Enterprise Edition- High performance, Enterprise Edition-Extreme performance

Standby Database with DataGuard comes with two options:

- High availability
- Disaster recovery

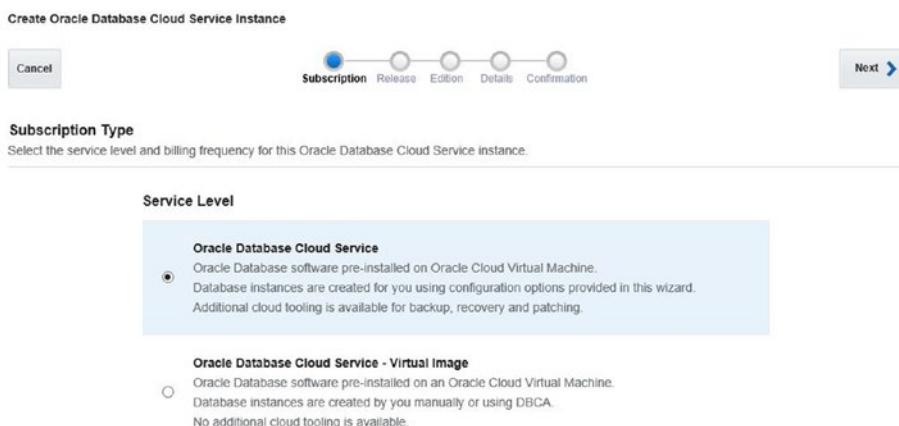
In high availability mode, you get DataGuard without disaster recovery, whereas in disaster recovery mode, you get DataGuard with disaster recovery. In other words, Data Guard with HA option ensures that your standby database is provisioned in different hardware in same data center whereas Data Guard with DR option ensures that your standby database is provisioned in different hardware in different data center.

Provisioning of DataGuard in Oracle Cloud

This section outlines the step to provision DataGuard setup in Oracle Cloud. This covers DataGuard setup for a standalone database, but you can choose to create DataGuard in RAC configuration to adopt a Maximum Availability Architecture (MAA).

Step 1: Log in to Oracle database Cloud Service wizard.

Step 2: Select Create Service and Choose Oracle Database Cloud Service in Service Level. This is shown in Figure 5-1.

**Figure 5-1.** DataGuard provisioning in Oracle Cloud: select the service level

Step 3: Select the required software release (any of the options 11.2.0.4 or 12.1.0.2 can be selected here), as shown in Figure 5-2.

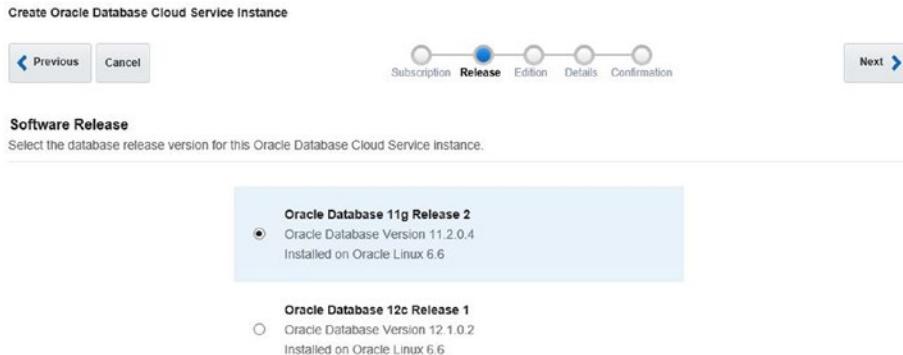


Figure 5-2. DataGuard provisioning in Oracle Cloud: select the release

Step 4: Select the required software edition (anything except Standard Edition), as shown in Figure 5-3.

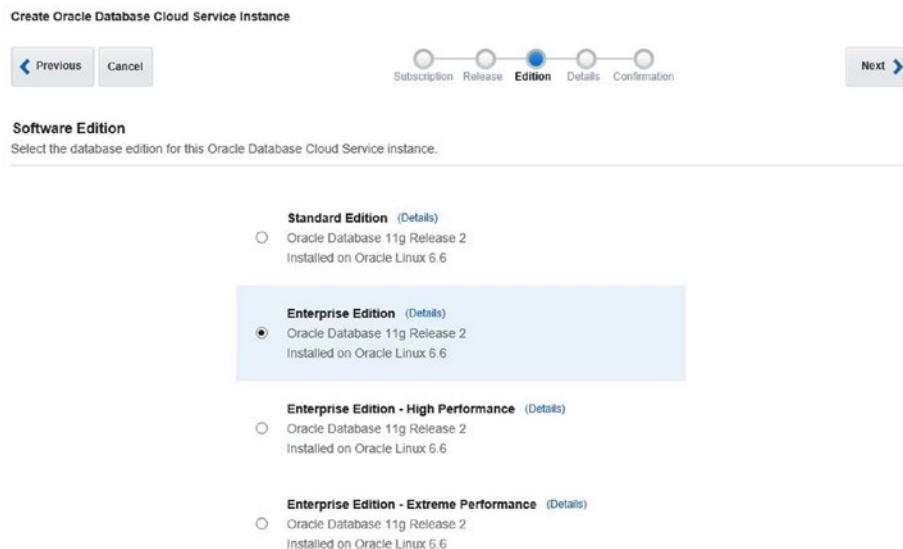


Figure 5-3. DataGuard provisioning in Oracle Cloud: select the edition

Step 5: Fill in the required details in the Service Details page and select Standby Database with DataGuard. Then select High Availability or Disaster Recovery, as shown in Figure 5-4.

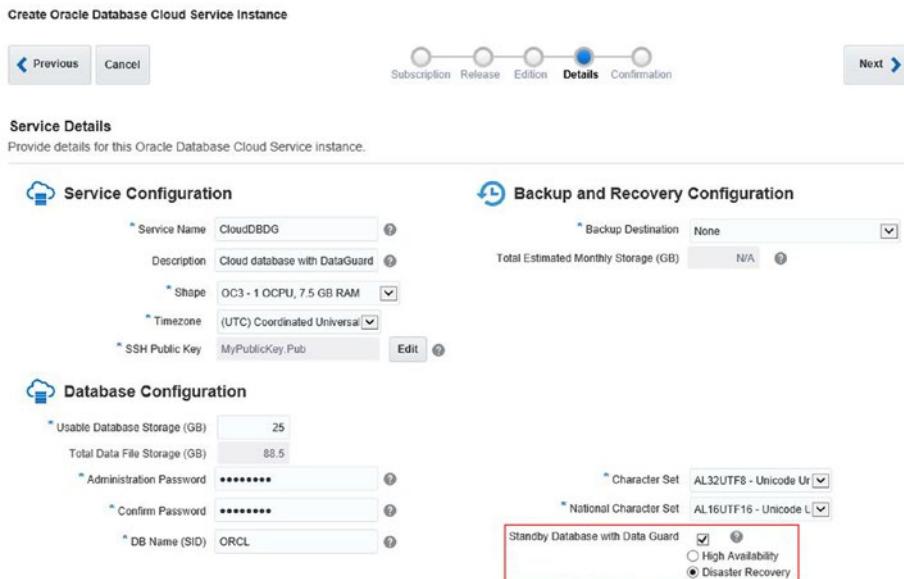


Figure 5-4. DataGuard provisioning in Oracle Cloud: select standby database with DataGuard

For the given selection, it took 50 minutes to set up Oracle DataGuard; however, your actual time taken will vary as per your selections. Now let's look at some of the key operations related to DataGuard in next section.

DataGuard Administration in Oracle Cloud

In this section, you will read about the methods for performing key DataGuard operations, including status check, switchover, failover, and reinstating failed primary databases. All of these operations can be done using the dbaascli utility or with Oracle database cloud service console.

Checking the DG Configuration Details

First let's look at what kind of DG configuration Oracle does. As you can see in Figure 5-5, Oracle created two hosts—CloudDBDG-dg01 and CloudDBDG-dg02—with exactly the same configuration (1 OCPU, 15GB memory, and 187GB storage). SID ORCL is running as the primary DB on the CloudDBDG-dg01 server and as standby on the CloudDBDG-dg02 server. Refer to Chapter 3 for more details on various provisioning options.

The screenshot shows the Oracle Database Cloud Service console for a CloudDBDG instance. The top navigation bar includes a cloud icon, the service name, and a gear icon. The main area has a light blue header with summary statistics: 2 Nodes, 2 OCPUs, 15 GB Memory, and 374 GB Storage. Below this is a table titled 'Nodes' showing two hosts:

	CloudDBDG-dg01	CloudDBDG-dg02
Public IP:	129.152.151.188	129.144.15.205
Database Role:	Primary	Standby
SQL *Net Port:	1521	1521
SID:	ORCL	ORCL
OCPUs:	1	1
Memory:	7.5 GB	7.5 GB
Storage:	187 GB	187 GB

On the left, there are sections for 'Overview' (2 Nodes), 'Administration' (1 Patch available), and 'Additional Information'. Under 'Additional Information', there are sections for 'Data Guard Metrics' and 'Activity'. The 'Data Guard Metrics' section includes metrics like Apply Lag, Transport Lag, and Standby Active Sessions.

Figure 5-5. DataGuard provisioning in Oracle Cloud: standby provisioning completion

Performing DB Switchover Using the Console

Let's look at performing DB switchover operation in cloud. DB switchover can be performed either by using the console or by using `dbaascli` commands.

For performing DB switchover using the console, click on the `switchover` command (Figure 5-6).

Overview

2 Nodes

Administration

1 Patches available

Summary

2 Nodes

Nodes

Node	Public IP	Database Role	Action
CloudDBDG-d1	129.152	Primary	Failover
CloudDBDG-d2	129.144	Standby	SSH Access

Figure 5-6. DataGuard in Oracle Cloud: DB switchover using the console

After clicking on Switchover, you will get confirmation. Click on OK to proceed, as shown in Figure 5-7.



Figure 5-7. DataGuard in Oracle Cloud: DB switchover confirmation

During the switchover (usually takes a few minutes), your DG setup will be shown in maintenance mode, as shown in Figure 5-8. During maintenance mode, application users won't be able to connect to the database.

The screenshot shows the Oracle Database Cloud Service dashboard. At the top, there are tabs for Services, Activity, and SSH Access, along with a Welcome message and REST APIs link. Below the header, a summary box displays resource details: 1 Service, 2 OCPUs, 15 GB Memory, 374 GB Storage, and 2 Public IPs. The main section is titled 'Services' and contains a search bar. A specific service entry for 'CloudDBDG' is shown, with its status highlighted in red as 'Maintenance'. Other details for this service include Version 11.2.0.4 and Enterprise Edition. To the right, deployment information is provided: Submitted On Sep 23, 2016 4:18:39 AM UTC, OCPUs 2, Memory 15 GB, and Storage 374 GB.

Figure 5-8. DataGuard in Oracle Cloud: DB switchover in progress

After completion of the switchover, you will see the roles reversed (Figure 5-9). Now dg01 is shown as the standby, whereas dg02 is shown as the primary. Application users can now connect to the DB as before. Application continuity features can be used to ensure application continuity and the least impact on the end users.

This screenshot shows the same Oracle Database Cloud Service interface after the switchover. The 'Nodes' section lists two database instances: 'CloudDBDG-dg01' and 'CloudDBDG-dg02'. 'CloudDBDG-dg01' is now listed as the Primary database, while 'CloudDBDG-dg02' is listed as the Standby database. Both nodes have identical configuration details: SQL *Net Port 1521, SID ORCL, 1 OCPU, 7.5 GB Memory, and 187 GB Storage. The timestamp indicates the data is current as of Sep 23, 2016 6:01:00 AM UTC.

Figure 5-9. DataGuard in Oracle Cloud: role reversal after DB switchover

Performing DB Switchover Using dbaascli

You can perform the switchover using the dbaascli commands. Let's look at this using an example. In this switchover test, we will first create a table and insert some records in it. After the switchover, we will see whether this table and its records are available.

Connect to Primary DB and create a sample table with 1 record.

```
[oracle@CloudDBDG-dg01 ~]$ sqlplus / as sysdba
SQL*Plus: Release 11.2.0.4.0 Production on Fri Sep 23 05:50:49 2016
Copyright (c) 1982, 2013, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
SQL> create table table_before_switchover(id number, details varchar2(50));
Table created.
SQL> insert into table_before_switchover values (1,'data before switchover');
1 row created.
SQL> commit;
Commit complete.

SQL> select * from table_before_switchover;

ID DETAILS
-----
1 data before switchover
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
[oracle@CloudDBDG-dg01 ~]$
```

Now let's check the status of DataGuard. As you can see, approximate role transition time is 34 seconds and data lag is only one second. This means the RTO is around 34 seconds and the RPO is 1 seconds. You can change the PROTECTION_MODE to reduce the RPO to zero.

```
[oracle@CloudDBDG-dg01 ~]$ dbaascli dataguard status --details yes
```

```
DBaaS CLI version 1.0.0
Executing command dataguard status --details yes
SUCCESS: Dataguard is up and running
{
  "instances": [
    {
      "DATABASE_TYPE": "PRIMARY",
      "PROTECTION_MODE": "MAXIMUM PERFORMANCE",
      "PROTECTION_LEVEL": "MAXIMUM PERFORMANCE",
      "ADDITIONAL_MESSAGES": "",
      "DATABASE_NAME": "ORCL_01",
      "PENDING_APPLY_LOG_CNT": 0,
      "ACTIVE_SESSIONS": "3",
      "HOST_NAME": "CloudDBDG-dg01",
      "OPEN_MODE": "READ WRITE"
    },
  ]
}
```

```
{
  "PROTECTION_LEVEL" : "MAXIMUM PERFORMANCE",
  "ADDITIONAL_MESSAGES" : "",
  "APPLY_LAG" : "00 days 00 hrs 00 min 04 sec",
  "DATABASE_NAME" : "ORCL_02",
  "APPROXIMATE_ROLE_TRANSITION_TIME" : "00 days 00 hrs 00 min 04 sec + 30 sec",
  "ACTIVE_SESSIONS" : "2",
  "HOST_NAME" : "CloudDBDG-dg02",
  "TRANSPORT_LAG" : "00 days 00 hrs 00 min 00 sec last computed 0 days 0 hrs 0 min 1 sec
before",
  "PROTECTION_MODE" : "MAXIMUM PERFORMANCE",
  "DATABASE_TYPE" : "PHYSICAL STANDBY",
  "OPEN_MODE" : "MOUNTED"
}
]
}
```

[oracle@CloudDBDG-dg01 ~]\$

After checking the status, we are all set for DB switchover. This is a single command operation. After running the `switchover` command, you need to restart ORDS using the root user.

ORDS stands for Oracle REST Data Service and is used to access the services over the Internet.

```
[oracle@CloudDBDG-dg01 ~]$ dbaascli dataguard switchover

DBAAS CLI version 1.0.0
Executing command dataguard switchover
Warning: Permanently added 'clouddb-dg01-nat,129.152.151.188' (RSA) to the list of known hosts.
Warning: Permanently added 'clouddb-dg02-nat,129.144.15.205' (RSA) to the list of known hosts.
Performing switchover NOW, please wait...
New primary database "ORCL_02" is opening...
Operation requires startup of instance "ORCL" on database "ORCL_01"
Starting instance "ORCL"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "ORCL_02"
SUCCESS : Switchover to Standby operation completed successfully

[root@CloudDBDG-dg01 opc]# /etc/init.d/ords restart
INFO: Stopping Oracle REST Data Services...
INFO: Oracle REST Data Services stopped
INFO: Starting Oracle REST Data Services...
INFO: Oracle REST Data Services started with PID 5860
[root@CloudDBDG-dg01 opc]# exit
```

After the switchover, you can log in to the new primary DB hosted on the second node and check the data from the table created before the switchover.

```
[oracle@CloudDBDG-dg02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Fri Sep 23 06:02:20 2016
Copyright (c) 1982, 2013, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from v$instance;

INSTANCE_NUMBER INSTANCE_NAME
-----
HOST_NAME
-----
VERSION      STARTUP_T STATUS    PAR   THREAD# ARCHIVE LOG_SWITCH_WAIT
-----        SHU DATABASE_STATUS INSTANCE_ROLE   ACTIVE_ST BLO
-----
LOGINS      SHU ALLOWED      ACTIVE PRIMARY_INSTANCE NORMAL NO

1 ORCL
CloudDBDG-dg02.compute-aclouddb.oraclecloud.internal
11.2.0.4.0      23-SEP-16 OPEN      NO      1 STARTED
ALLOWED      NO ACTIVE      PRIMARY_INSTANCE NORMAL NO

SQL> select * from table_before_switchover;

ID DETAILS
-----
1 data before switchover

SQL>
```

This concludes the successful switchover operation.

Performing DB Failover Using the Console

The DB failover operation can be performed using the console or with dbaascli. Let's first perform it using the console. For performing DB failover using the console, click on Failover option, as shown in Figure 5-10.

The screenshot shows the Oracle Database Cloud Service / CloudDBDG interface. On the left, there are three cards: 'Overview' (2 Nodes), 'Administration' (1 Patches available), and 'Summary' (2 Nodes). The 'Summary' card is active, showing two database instances: 'CloudDBDG-d1' (Public IP: 129.152.1.11, Database Role: Primary) and 'CloudDBDG-d2' (Public IP: 129.144.1.11, Database Role: Standby). A context menu is open over the 'CloudDBDG-d1' instance, listing options like Open DBaaS Monitor Console, Open Application Express Console, Open EM Console, Start, Stop, Restart, Switchover, and Failover. The 'Failover' option is highlighted with a red circle.

Figure 5-10. DataGuard in Oracle Cloud: DB failover using the console

After clicking on Failover, you will be asked for a confirmation similar to the one shown in Figure 5-11.



Figure 5-11. DataGuard in Oracle Cloud: DB failover confirmation

Click on OK to proceed. It will take few minutes to complete the failover. Afterwards, the status of your primary database will be *Reinstate*, whereas your standby database will become *Primary*.

You can also check the status of the DataGuard configuration from `dbaascli`. It will show the following output.

```
[oracle@CloudDBDG-dg01 ~]$ dbaascli dataguard status
DBAAS CLI version 1.0.0
Executing command dataguard status
SUCCESS : Dataguard is up and running

DETAILS:
Configuration - fsc

Protection Mode: MaxPerformance
Databases:
ORCL_02 - Primary database
ORCL_01 - Physical standby database (disabled)
ORA-16661: the standby database needs to be reinstated

Properties:
FastStartFailoverThreshold      = '30'
OperationTimeout                = '120'
FastStartFailoverLagLimit        = '30'
CommunicationTimeout           = '180'
ObserverReconnect               = '0'
FastStartFailoverAutoReinstate = 'TRUE'
FastStartFailoverPmyShutdown    = 'TRUE'
BystandersFollowRoleChange     = 'ALL'
ObserverOverride                = 'FALSE'
ExternalDestination1            = ""
ExternalDestination2            = ""
PrimaryLostWriteAction          = 'CONTINUE'

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS
```

Reinstating Failed Primary Database

You need to reinstate the old primary database after the failover, in order to make your old primary DB convert to the standby role. You can initiate this process by clicking on Reinstate, as shown in Figure 5-12.

The screenshot shows the Oracle Cloud interface for managing a DataGuard configuration. At the top, it displays system statistics: 2 Nodes, 2 OCPUs, 15 GB Memory, and 374 GB Storage. Below this, the 'Nodes' section lists two database nodes:

- CloudDBDG-dg01**: Public IP 129.152.151.188, Database Role: **Reinstate** (highlighted with a red circle).
- CloudDBDG-dg02**: Public IP 129.144.15.205, Database Role: Primary.

For each node, there are detailed resource specifications and management actions:

	SQL *Net Port	SID	OCPUs	Memory	Storage	Action
CloudDBDG-dg01	1521	ORCL	1	Start		
CloudDBDG-dg01	1521	ORCL	1	Stop		
CloudDBDG-dg01	1521	ORCL	1	Restart		
CloudDBDG-dg02	1521	ORCL	1	Switchover		
CloudDBDG-dg02	1521	ORCL	1	Failover		
CloudDBDG-dg02	1521	ORCL	1	Reinstate (highlighted with a red circle)		
				Scale Up/Down		

At the bottom, there is an 'Additional Information' section.

Figure 5-12. DataGuard in Oracle Cloud: reinstating failed primary

After The reinstate operation is completed, you will see that your primary database role has changed to Standby. This is shown in Figure 5-13.

The screenshot shows the Oracle Cloud interface after the reinstate operation for dg01 has been completed. The 'Nodes' section now displays:

- CloudDBDG-dg01**: Public IP 129.152.151.188, Database Role: **Standby** (highlighted with a red circle).
- CloudDBDG-dg02**: Public IP 129.144.15.205, Database Role: Primary.

The detailed resource specifications and management actions remain the same as in Figure 5-12, reflecting the restored configuration.

Figure 5-13. DataGuard in Oracle Cloud: reinstate completion

With this, you have successfully restored your original DataGuard configuration.

DR Options in AWS

There are two DR options available in AWS:

- Multi-AZ deployment
- DataGuard configuration

Multi-AZ deployment option is applicable only for Amazon RDS and it is already explained in Chapter 4, Database High Availability. Since this option provisions your database into two availability zones, it can be used as a DR option for handling availability zone level failure.

DataGuard configuration works in EC2 kind of setup and protects you from availability zone or region-level failures.

Oracle DataGuard can be configured in the following ways:

- Primary DB in on-premise and DR in AWS
- Primary and standby DB in the same AWS region
- Primary and standby DB in different AWS regions

The DataGuard configuration steps are similar to on-premise DB, hence those are not covered in this book.

Summary

This chapter covered the importance of disaster recovery and also went through the various options available in Oracle and AWS Cloud. You learned how to set up DataGuard in Oracle Cloud, how to do switchovers and failovers, and how to instantiate failed primary DBs using a GUI interface and using the command-line interface.

In the next chapter, you learn about the cloud security model, including security configuration in Oracle Cloud, security configuration in AWS, and DB security best practices.

CHAPTER 6



DB Security

In this chapter, you first learn about the cloud security model that explains the difference between cloud provider's responsibilities and yours. After that, you will read about the various security configurations required in Oracle Cloud and in AWS. This chapter covers SQL*Net encryption, transparent data encryption, and the security best practices for Oracle database running on Amazon RDS and EC2.

Need for Database Security with Cloud-Based DB

Security is number one priority for any organization moving to the cloud. In fact, this is the first priority for cloud service providers as well. With on-premise databases, you are only concerned about security around the database, but with cloud DBs, you need to think beyond the DB. Since your DB is being accessed over the Internet, you have to take care of the security of all aspects of it, such as security during transit of data and security at rest.

In the following sections, you will go through all the available security options. Let's first look at the basic cloud security model.

Cloud Security Model

The cloud security model is a shared security model where you and the cloud provider share the responsibility of various components (Figure 6-1).

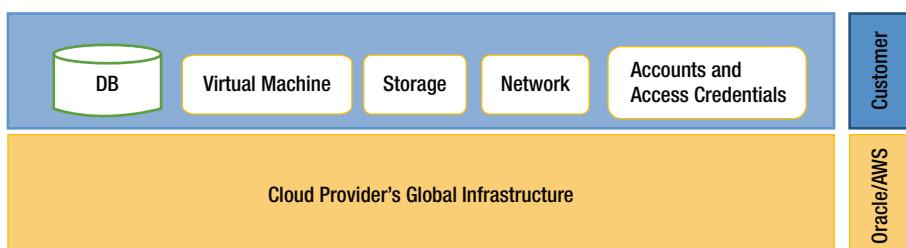


Figure 6-1. Cloud security model

As you can see, securing the datacenters and infrastructure is the cloud provider's responsibility, whereas security of the database, storage, network, and accounts is the customer's responsibility. In other words, security "of" the cloud is the cloud provider's responsibility, whereas security "in" the cloud is your responsibility. Amazon and Oracle take care of following, which is what the cloud is made of:

- Physical security of datacenter premises
- Access control mechanism
- Firewall and network security monitoring
- Hypervisor security
- OS-level patching and upgrade for managed services offering (such as Amazon RDS and Oracle Cloud) where you don't get OS-level access

As a DBA, your security responsibilities vary according to the DB hosting model. If you are using an IaaS kind of service, such as Oracle database running on Amazon EC2 instance, then you are responsible for securing your EC2 instance, your EBS/S3 storage, your network like VPC, and your database.

If you are running a database in a managed service kind of offering, such as Amazon RDS or Oracle Schema as a Service then your security related responsibilities are limited to accounts and credential management. In this case, you don't get OS-level access. This is explained in more detail in the next sections.

Security Configurations in Oracle Cloud

In Oracle Cloud, the following offerings are presented:

- Database Cloud Service
- Exadata Express Cloud Service
- Exadata Cloud Service
- Database Schema Service

Your security configuration requirements depend on the type of service that you are using. For example, in the Database Schema service, you have to do minimal security configurations.

Let's see what is required for the database cloud service security configuration. In a database cloud service, you need to focus on the security tasks listed in Table 6-1. This is in addition to securing the database cloud subscription account (securing the AWS root account, etc.).

Table 6-1. Security Task List at Different Layers

Layer	Tasks
VM	<ul style="list-style-type: none"> Securing your VM by defining network security policies
OS	<ul style="list-style-type: none"> Securing your OS by patching and OS-level hardening
Network	<ul style="list-style-type: none"> Securing your network by implementation of security groups, Network Access Control Lists (NACL)
Database	<ul style="list-style-type: none"> Securing data that is stored inside the database by implementing TDE Securing your backups (RMAN/data pump) by encryption Securing your database using least privilege mechanism Securing your access to the database by database access control Securing data transfer by implementing SQL*Net encryption

Oracle Cloud has two options for database security:

- Transparent Data Encryption (TDE)
- SQL*Net Encryption

Transparent Data Encryption (TDE)

Oracle Cloud automatically creates all new table spaces as encrypted (with the AES128 algorithm), whereas for on-premise databases, you have to explicitly mention ENCRYPT in the `create tablespace` clause.

Let's see this difference with an example:

```
SQL>select NAME, OPEN_MODE from v$pdbs;

PDB1  READ WRITE

SQL>alter session set container=PDB1;

Session altered.

SQL>Select tablespace_name, encrypted from dba_tablespaces;

SYSTEM NO
SYSAUX NO
TEMP  NO
USERS NO

SQL>Create tablespace newtbs datafile '/u02/app/oracle/oradata/CDB01/PDB1/newtbs.dbf' Size 10M
logging extent management local segment space management auto;
```

Tablespace NEWTBS created.

```
SQL>Select tablespace_name, encrypted from dba_tablespaces;
```

SYSTEM	NO
SYSAUX	NO
TEMP	NO
USERS	NO
NEWTBS	YES

You can control this tablespace encryption using the ENCRYPT_NEW_TABLESPACES initialization parameter.

There are three possible values:

- ALWAYS
- CLOUD_ONLY
- DDL

For Oracle database in the cloud, this parameter is set to CLOUD_ONLY.

SQL*Net Encryption

SQL*Net encryption is automatically enabled by default for your Oracle database running in Oracle database cloud service. You can check this configuration by looking at your `sqlnet.ora` file.

If you want to disable this configuration, you can simply rename the `sqlnet.ora` file. However, it is not recommended.

Security Configurations in AWS

Security in AWS can be categorized into three levels.

- Security at the account level
- Security for Oracle running on Amazon RDS
- Security for Oracle running on Amazon EC2

Security at the Account Level

Account-level security means that security controls are in place at the AWS resource level. It is done by using a free web service called Identity and Access Management (IAM). You can access IAM using the AWS Management console, the AWS command line, SDKs, and by IAM APIs.

IAM can be accessed using the AWS console by clicking on **Identity and Access Management**, which is present under the Security and Identity option (Figure 6-2).

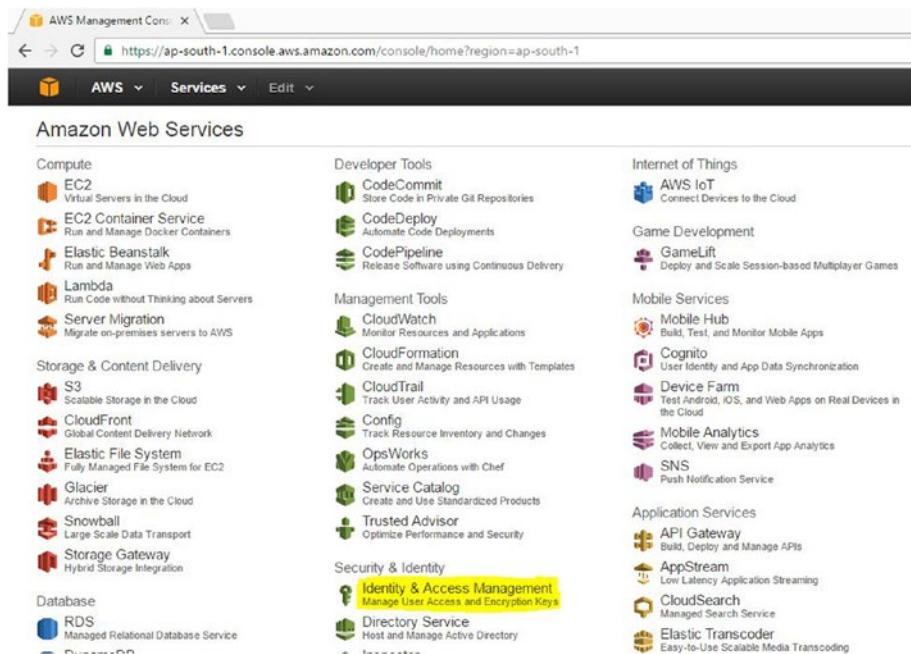


Figure 6-2. IAM access using the AWS console

You will see an IAM dashboard similar to the one in Figure 6-3.

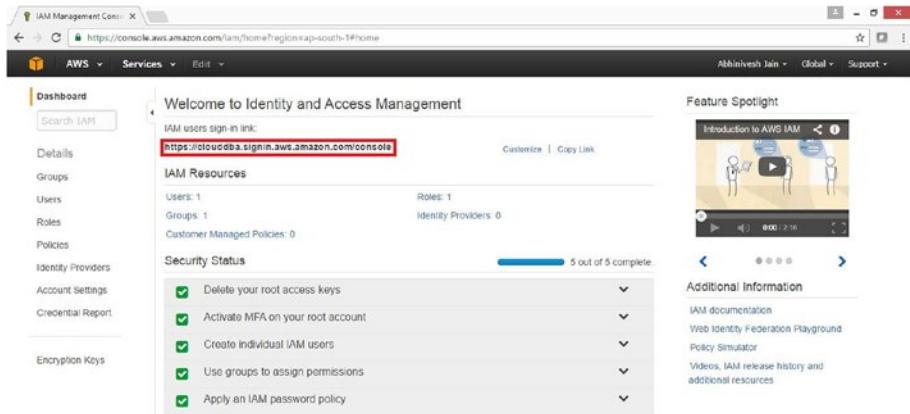


Figure 6-3. The IAM dashboard

In Figure 6-3, you can see the IAM users' sign-in link, which is customized to make it readable. This link can be used to access the IAM dashboard and perform all IAM related tasks. This way, you don't need to give the AWS console access to the IAM management team.

IAM allows you to create groups, users, and roles, define password policies, generate credential reports, and use the AWS Key Management Service (KMS). Remember, these users, roles, and password policies are not related to Oracle users, roles, or password policies. These are at the AWS account level rather than at Oracle database level. AWS account level security configurations are part of the cloud admin's responsibility. However, in smaller deployments, these could be part of the DBA responsibilities, hence these are covered in this chapter.

Security for Oracle Running on Amazon RDS

In Amazon RDS, you first define the *master* username. This user defines all the other users, objects, and permissions (Figure 6-4).

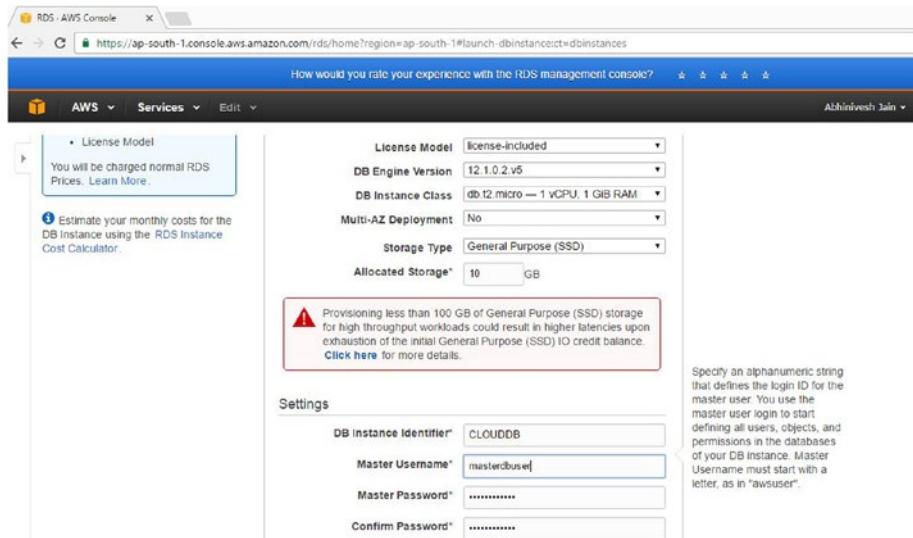


Figure 6-4. Defining the master user in Amazon RDS

The master account gets DBA privileges with some limitations. Other administrative accounts, such as SYS and SYSTEM, are locked and cannot be used. You use the master account for any administrative tasks.

The RDS instance also allows you to further enhance its security by configuring advanced settings, as shown in Figure 6-5.

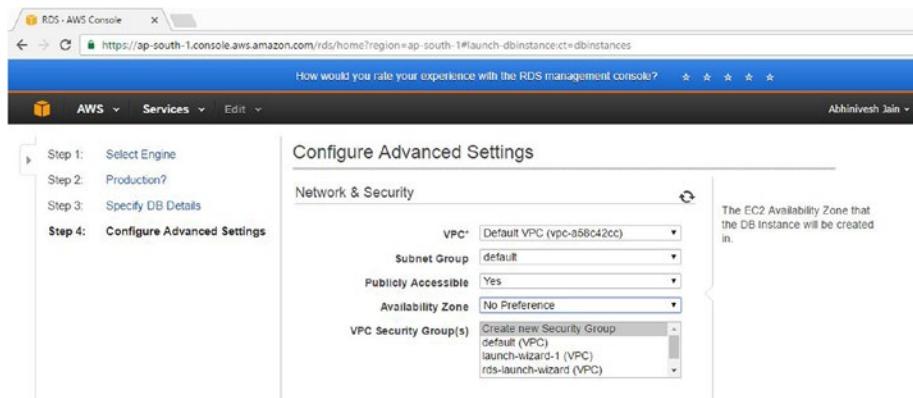


Figure 6-5. Advanced settings in Amazon RDS

Configuring VPC, subnet, and VPC security group is usually the cloud system administrator's responsibility; however, as a DBA, you must know how to do it.

You can also choose **No** for the publicly accessible setting. If you select No, no EC2 instance or other devices outside the VPC will be able to access your database.

It is recommended that you enable encryption while creating the RDS instance. This can be done by selecting the Enable Encryption option, as shown in Figure 6-6.

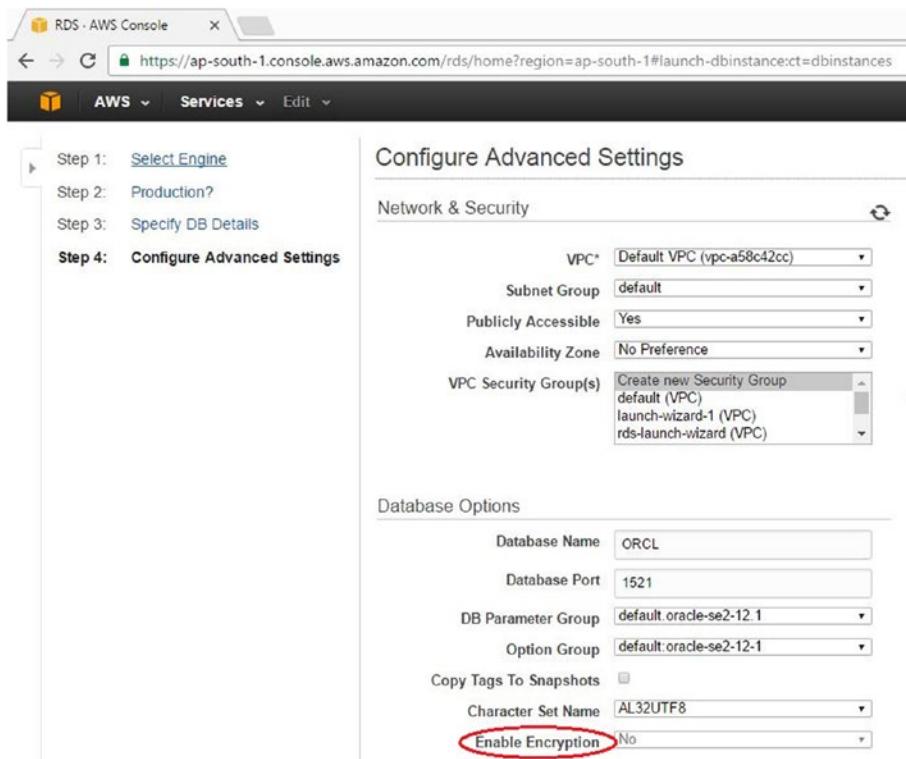


Figure 6-6. Enable encryption in Amazon RDS

Encryption is not available for all DB instance classes. It is not available for **t2.micro** (free tier), hence it is disabled. In the T2 category, this option is only available for **t2.large** instance types. Using different DB instance classes will result in higher costs, hence the appropriate instance class should be selected based on your requirements.

Security for Oracle Running on Amazon EC2

Security in Amazon EC2 is handled at multiple levels. Hypervisor level security and instance isolation is Amazon's responsibility. Instance isolation ensures that instances running on the same physical machines are isolated. You do not get direct access to raw disks, rather it is accessed by virtual disks so that your data is not exposed to others. Memory scrubbing, no elevated access to CPU, and AWS firewall are some other measures that AWS takes care of to ensure proper security.

Let's see what you need to do as the DBA to secure your Oracle database running in EC2.

Using an Encrypted EBS Volume

As a DBA, you need to ensure that you use an encrypted filesystem for keeping your database related files. This should be done while launching the EC2 instance (Figure 6-7). Just be sure to click on the Encrypted checkbox. This should be done for all EBS volumes in which you are planning to keep your database-related files.

The root volume is by default not encrypted, but it can be encrypted using third-party software, such as BitLocker.

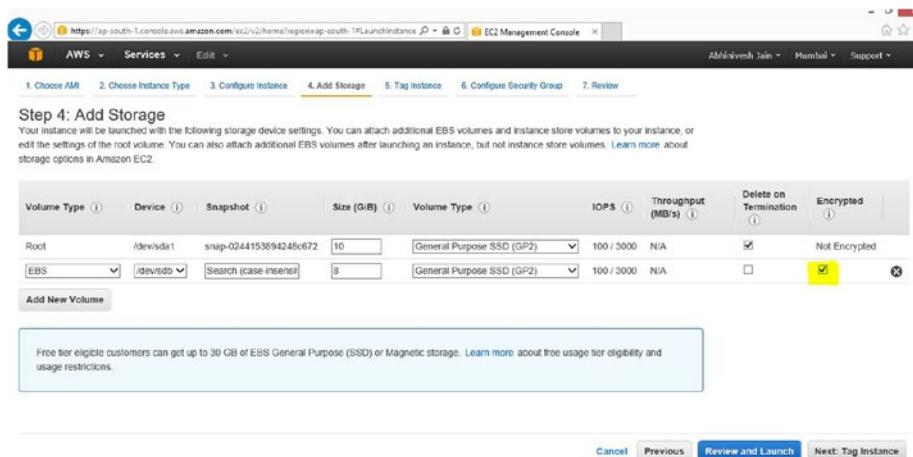


Figure 6-7. Enable encryption in the Amazon EC2 EBS volume

Using Correct Source IP

Allowing SSH access with the source as *Anywhere* is never recommended, as this will open your database server to everyone in the world. Figure 6-8 shows this kind of configuration.

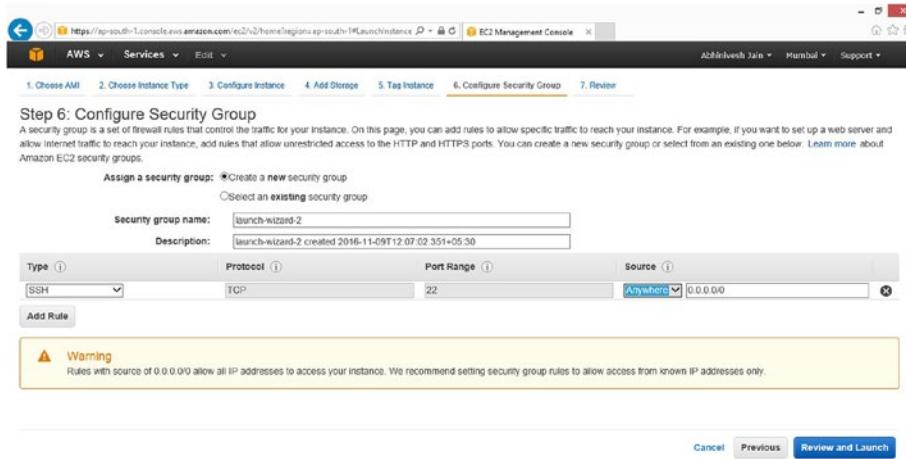


Figure 6-8. SSH port opening in Amazon EC2

Rather than opening Anywhere access, for trial purposes, you can select the My IP or Custom IP range. For production use or for any of customer database, this is not recommended and instead you should provide a specific IP range here.

Using VPC and Private Subnets

You should put your database in a custom VPC and inside a private subnet in that VPC. It should be accessed via some jump server (also known as a *bastion host*).

These settings can be done as part of the Configure Instance Details page when you create the EC2 instance (Figure 6-9).

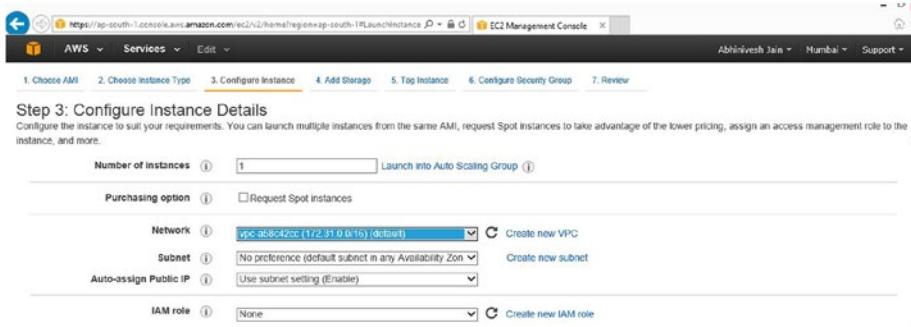


Figure 6-9. Amazon EC2 in VPC and private subnets

Using Dedicated Hosts

Based on the requirements and as an additional security measure, you can opt for dedicated hosts for your EC2 instance. This is controlled using the Tenancy option shown in Figure 6-10.

The default setting is Shared, whereby your EC2 instance runs on a shared hardware instance.

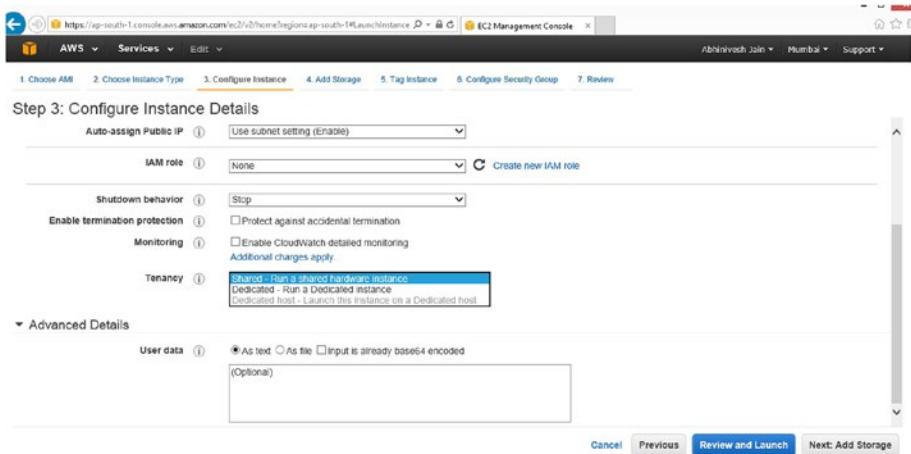


Figure 6-10. Amazon EC2 tenancy options

Security Best Practices

The following list explains some security best practices for Oracle database running on Amazon RDS:

- Run Oracle DB instance in a custom Amazon virtual private cloud other than default one.
- Use security groups to control IP addresses that are allowed to connect to your database.
- Use IAM and assign permissions to users who are allowed to manage your RDS resources.
- Use SSL to encrypt connections to the databases.
- Use RDS encryption. This will help secure your RDS instances and snapshots at rest.
- Use Oracle-specific security features that you would use with your on-premise database.
- Use Transparent Data Encryption (TDE) and network encryption.

The following list explains some of the security best practices for Oracle database running on an Amazon EC2 server:

- Keep your EC2 instance-based DB server up-to-date with OS-level security patches.
- Use Secure Shell (SSH) to log in to your Linux-based DB server.
- Disable password only access to your guests and use MFA (multifactor authentication).
- Disable remote root login.
- Use sudo to do privilege escalation.
- Do not share key pairs with AWS or other customers.
- Use the AWS firewall (security groups).
- Use the Windows firewall and secure VPNs.

Summary

This chapter explained the importance of database security for databases running in the cloud. You went through the cloud security model to understand your roles and responsibilities. You learned about the security configuration options in AWS and Oracle Cloud. In the end, you learned about some of the best practices related to database security. Covering all aspects of cloud security would require writing an entire book itself and is beyond the scope of this book, hence the focus of this chapter is on DB security only.

In the next chapter, you will learn about database migration to the cloud.

CHAPTER 7



DB Migration to the Cloud

In this chapter, you will go through the most important task of cloud DBA, which is database migration to the cloud. There are various ways you can do DB migration to Oracle or Amazon Cloud, including Export/Import, using SQL Developer, and RMAN based restore. Apart from these, some not so common methods are also available, such as SAN replication based restore and VM image level backup restore.

This chapter provides details on these Oracle database migration methods, along with recommendations and best practices. It also covers the challenges, limitations, and comparison between various migration methods for migrating Oracle database to the cloud.

DB Migration Key Considerations

Deciding the right migration method is a very challenging and complex task and you have to consider several factors before arriving at right migration method. One migration method might not be suitable for your entire DB landscape migration to the cloud, hence you'll need to identify a mix of migration methods to handle different needs.

Let's look at the factors/considerations for migrating a database to the cloud:

- *Database size:* This is the single most important factor along with available downtime.
- *Database version and edition:* Cloud providers do not support all editions or older DB versions like Oracle 10g and below, hence comparing the source DB version and edition with the allowed/available one is important.
- *ARCHIVELOG mode:* Migration methods such as RMAN hot backup, DataGuard-based migration, and many other methods require databases to be running in ARCHIVELOG mode. This is why this information should be captured before deciding the migration method.
- *Migration level:* Migration can be at different levels, such as DB server, DB schema, or even the table level.

- *OS endianness (Little endian/ big endian)*: Cloud providers do not support all OSs, hence chances of having to do a cross-platform migration is very high. For cross-platform migration, it is very important to know the OS endianness so that the appropriate migration method can be selected.
- *32-bit/64-bit*: 32-bit servers are not very common, but if you have very old legacy systems running on 32-bit, then you need to consider moving them as-is or converting them to 64-bit systems.
- *Physical/virtual*: The cloud is mostly a virtual environment (except where physical hardware is provided by a cloud provider, like the Exadata cloud service, where Exadata is physically hosted in your datacenter). If you are running your Oracle database on a physical server, you will have to convert it into virtual server (P2V migration). Performance, licensing, supportability, certification, and many other factors come into the picture while doing such P2V migration.
- *Law of the land*: Certain regulatory requirements might require you to keep your database in your own datacenter or in your country. You should consider such requirements while planning for migrating to the cloud.
- *Data security requirements*: Certain databases might have SOX, HIPAA, PCI, or any other regulatory requirement for data security, and this will result in certain migration considerations for data at transit or data at rest.
- *Application nature (mission-critical/performance sensitive/big data)*: The application nature defines the downtime availability, hence it affects the migration method used.
- *Available downtime and available time for migration*: Some database migrations might have zero or very little downtime. This is an important factor to remember while determining the migration method.
- *Network bandwidth between on-premise and the cloud*: Network bandwidth plays an important role with large database migrations.
- *Application compatibility*: While doing database migration, you need to consider whether the application is compatible with the proposed cloud deployment model.
- *Cloud provider (Oracle Cloud, AWS, or Microsoft Azure)*: Different cloud vendors have different migration methods. Amazon AWS has a database migration service called DMS, which is not available in Oracle Cloud.

- *Tools availability (OGG, OEM12c cloud control)*: Tools availability in your DB landscape will also play a critical role in deciding your tool-based migration method.
- *Cloud deployment model (IaaS, PaaS, DBaaS)*: The migration method depends on the database deployment model, for example, you can't do a RMAN backup restore in Oracle Schema as a service deployment.
- *Cloud type (public/private)*: Migration methods for a private cloud in your own datacenter will be different than for a public cloud.

In the next sections, you will go through various migration methods. Let's first look at the migration lifecycle.

Migration Lifecycle

Migration is multistep process; the different stages of the migration lifecycle are listed here:

- Analyze
- Identify
- Prepare
- Migrate
- Validate

These stages (Figure 7-1) are best carried out as a pilot or dry run before planning for actual migration. The process can be iterative and may need multiple runs, as you may encounter issues that need fixing.

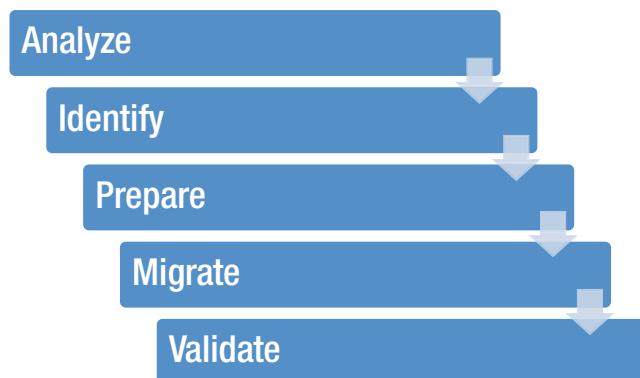


Figure 7-1. Cloud migration lifecycle

In the Analyze phase, you go through the customer requirements and key considerations. These inputs help in the next phase, called Identify, where you identify the migration method.

The Prepare phase is all about getting ready for migration by having things in place. For example, you prepare the target environment for DB migration.

In the Migrate phase, you do the actual migration and in the Validate phase, you do data integrity validation along with application testing and validation.

This is very high-level view so that you can understand where actually the migration part fits in this larger scheme of things. The next section discusses the migration phase.

Migration Approach

At a high level, all DB migration method consists of three phases, as shown in Figure 7-2.



Figure 7-2. *The cloud migration approach*

Let's take a look at these phases.

Backup

Backup means backup of DB server, database, schema, table, etc. You will go for a DB server level backup when you are moving to an IaaS or PaaS kind of offering. This is also called VM or server-level image. You can take database-level backups as physical or logical. Physical backups consist of RMAN or user-managed backups like hot or cold backups, whereas logical backups consist of database export dumps.

Transfer

The Transfer phase refers to moving the backup to the cloud. The exact transfer method depends on the size and the network bandwidth. If the database is running into the terabytes (TB) and you don't have sufficient bandwidth to copy it quickly, then special migration methods like hard disk shipping or Amazon Snowball are used. These are explained in the migration method section. With some migration methods, you may not need to actually transfer any files, e.g. while using a DB link or the materialized views-based migration method. In such cases, data transfer is done using buffers, but there is no file movement as such.

You might also need to apply acceleration methods (such as multipart upload, Tsunami UDP, S3 transfer acceleration, and AWS Direct Connect) to get the optimal migration time.

Restore

The Restore phase refers to restoring the DB server image or restoring the database backups. You don't see this phase in a few migration methods, such as with DB links or the MView refresh-based migration method.

Migration Options

There are various options for DB migration to the cloud and the method you choose depends on several factors that are explained in the previous section. Your migration strategy should be based on your organization's priorities, the end state architecture, and the available budget.

Let's look at the various migration options for migrating Oracle database to the cloud.

Migration Using Tools

The latest Oracle and third-party database management tools support migration to the cloud.

SQL Developer

At present, Oracle's native GUI tool, SQL Developer, has built-in capabilities to connect to the cloud database (Figure 7-3) and can also be used to migrate data from on-premise databases to the cloud.

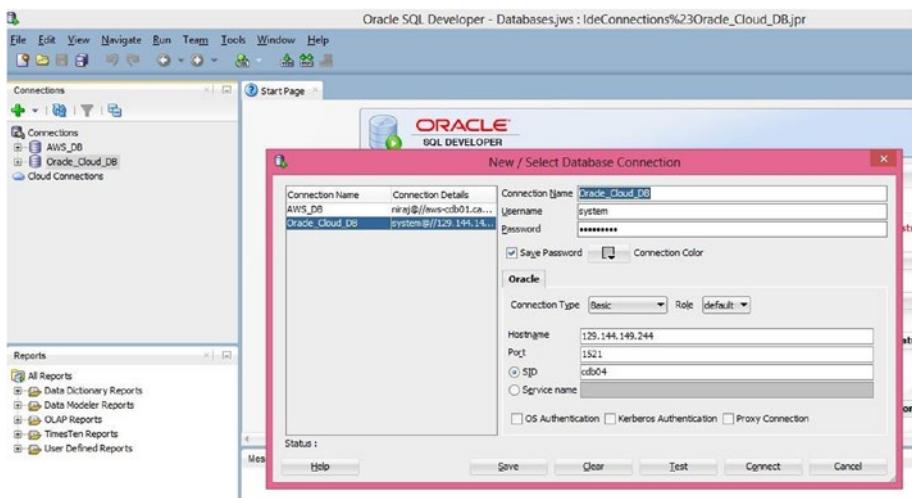


Figure 7-3. Cross-cloud migration using SQL Developer: login

After establishing a connection to the local and the cloud database using SQL Developer, choose Tools ▶ Database Copy and select both databases. There are options to copy selective objects, schemas, or table spaces, as shown in Figure 7-4. This method is very easy and good for smaller databases.



Figure 7-4. Cross-cloud migration using SQL Developer: the copy wizard

Migration using SQL Developer is covered in detail at a later stage in this chapter.

Oracle Enterprise Manager (OEM)

Oracle Enterprise Manager (OEM) is another tool-based option for migrating data to the cloud. OEM can also be used to move data from the local database to the cloud using the Export/Import method or even be used to set up DataGuard in the cloud.

Traditional Migration Methods

The traditional migration method that you use to move your Oracle database across database servers can be used to migrate to the cloud as well.

Import/Export

Logical data backup and data movement across Oracle databases has been achieved using Import/Export and data pump for a very long time now. The same method can be extended to move data from a local DB to the cloud. As a prerequisite, you should run a successful tnsping test between the two DBs. An authenticated DB link pointing to the cloud DB from local DB can be used. The remote filesystem is accessed using such a DB link, using DBMS_FILE_TRANSFER.PUT_FILE to copy the dump file to the cloud before starting the import. Figure 7-5 shows the high-level steps for migrating to the cloud using data pump.

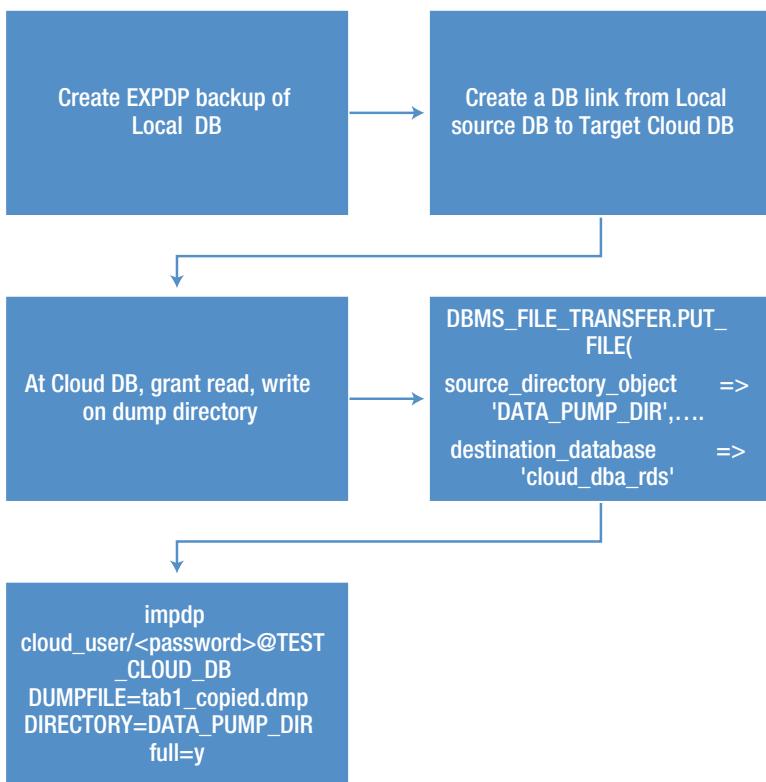


Figure 7-5. Cloud migration using data pump

There is no access provided to the filesystem of AWS RDS, but the following command can help you verify the dump file upload. Migration using data pump is covered in detail at a later stage in this chapter.

On Cloud DB

```
SQL> select * from table(RDSADMIN.RDS_FILE_UTIL.LISTDIR('DATA_PUMP_DIR'));
```

RMAN Backup Restore

Similar to Import/Export, DB migration can be done using RMAN backups. Taking RMAN DB backup to the cloud filesystem and restoring it on the cloud is similar to the local restore process. Moving the local database backup to the cloud is covered in more detail in Chapter 8, “Backup and Restore”.

Shipping the Hard Drive

For large databases, it is not practical to move data over the network, considering the restore time and network bandwidth related expenses for a one-time data transfer. In such cases, the traditional method that was commonly used for datacenter migration comes handy—physically shipping the hard drive. This method can be used along with multiple restore methods, like cold restore, Import/Export, etc.

Oracle also offers Oracle Cloud storage appliance-based import option as part of its data transfer services. This helps with secure, bulk data movement. You can send up to 400TB in one go.

Lesser Known Migration Methods

In addition to the traditional methods, some lesser-known methods are:

- Create a DataGuard in the cloud
- Perform on-premise backups to the cloud and restore the cloud backup to create the DB in the cloud
- Perform a DB server VM backup and restore the machine backup to the cloud, then restore it as new cloud VM
- Use Oracle Golden Gate-based DB replication to the cloud
- For AWS, Data Migration Services (DMS) is also a very effective method

Comparison of Migration Methods

Table 7-1 shows a quick comparison of the various migration methods, which can help you decide which migration method to choose, given your scenario.

Table 7-1. *Migration Methods Comparison*

Migration Method	Table Level Migration	Schema Level Migration	Database Level	DB Server Level
SQL Developer	Yes	Yes	Yes	No
Data Pump	Yes	Yes	Yes	No
DataGuard	No	No	Yes	No
RMAN Backup Restore	No	No	Yes	No
VM Image	No	No	No	Yes
Oracle Golden Gate Based DB Replication	Yes	Yes	Yes	No

Detailed Migration Steps for Data Migration from the Local Database to Amazon RDS

The initial chapters already covered how to create a database in AWS RDS, hence the steps given here are based on the assumption that you have already created an empty database in AWS RDS. It is also assumed that the connectivity to the database is tested successfully using SQL Developer and other connectivity options.

You can start data migration using the SQL Developer data copy option. The local DB referred to in this section means database hosted in your datacenter.

Let's first create a new user called `local_machine_user` and, after giving it proper privileges, create and populate the table.

```
select instance_name,status,host_name from v$instance;

INSTANCE_NAME      STATUS        HOST_NAME
-----
local_db           OPEN          LOCALHOST

SQL> create user local_machine_user identified by xxxxxxxxx;

User created.

SQL> grant dba to local_machine_user;
Grant succeeded.

SQL> conn local_machine_user/xxxxxxxxxx
Connected.

SQL> create table testing as select * from dba_objects;
Table created.

SQL> insert into testing select * from testing;
/

SQL> commit;

Commit complete.

SQL> select count(*) from testing;

COUNT(*)
-----
128,670

SQL> select sum(bytes)/1024/1024 from user_segments where segment_name='TESTING';

SUM(BYTES)/1024/1024
-----
12.635
```

Let's now copy this 12MB table to the AWS Oracle database.

First, connect to the local DB using SQL Developer, as shown in Figure 7-6.

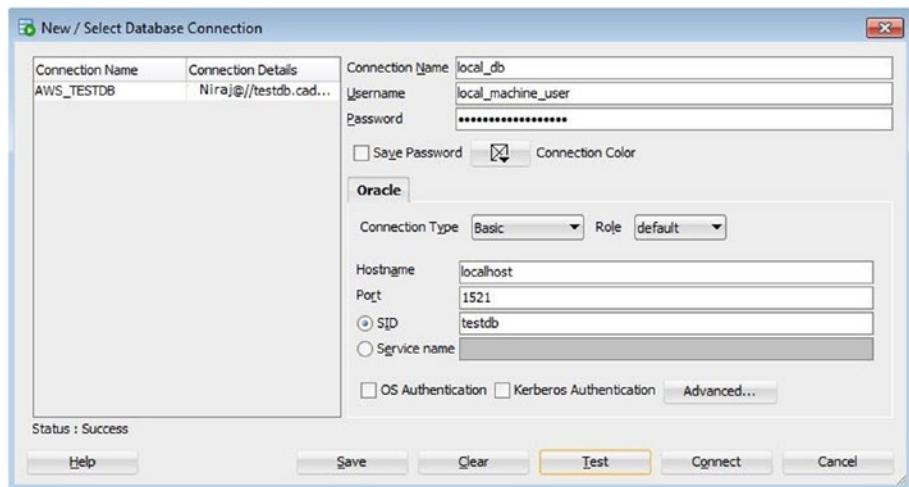


Figure 7-6. Connecting to the local DB using SQL Developer

Now choose Tools > Database Copy and select both databases, as shown in Figure 7-7. Ensure that the correct source and destination databases are selected. Making wrong or inverted selections at this stage can lead to painful recovery needs at a later stage.

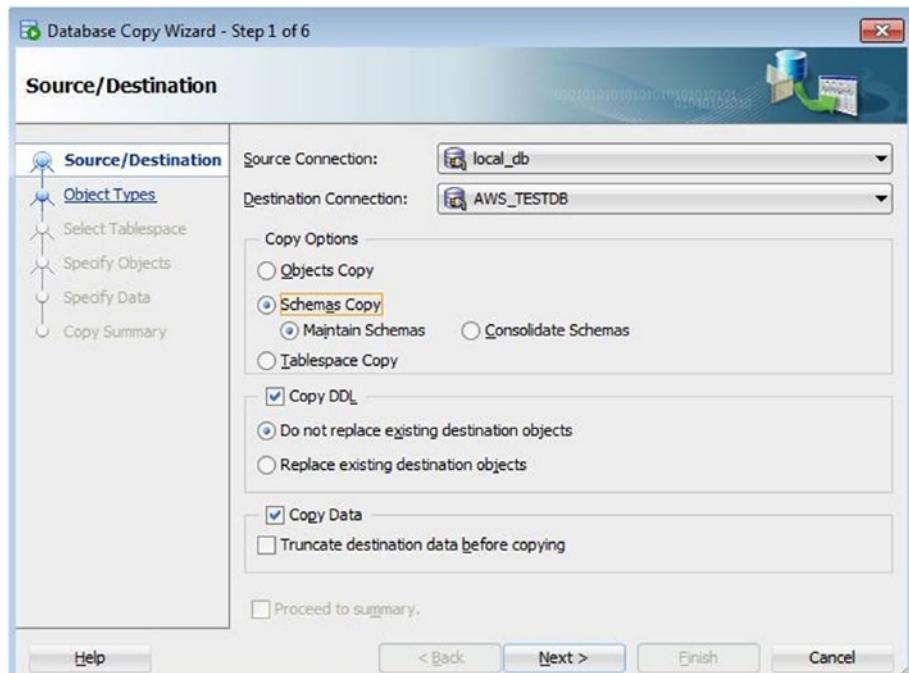


Figure 7-7. Database copy wizard: select source/destination

There are various options here, for example, you can choose a set of objects or a schema. You can even select multiple schemas to be merged or even a particular tablespace, as shown in Figure 7-8.

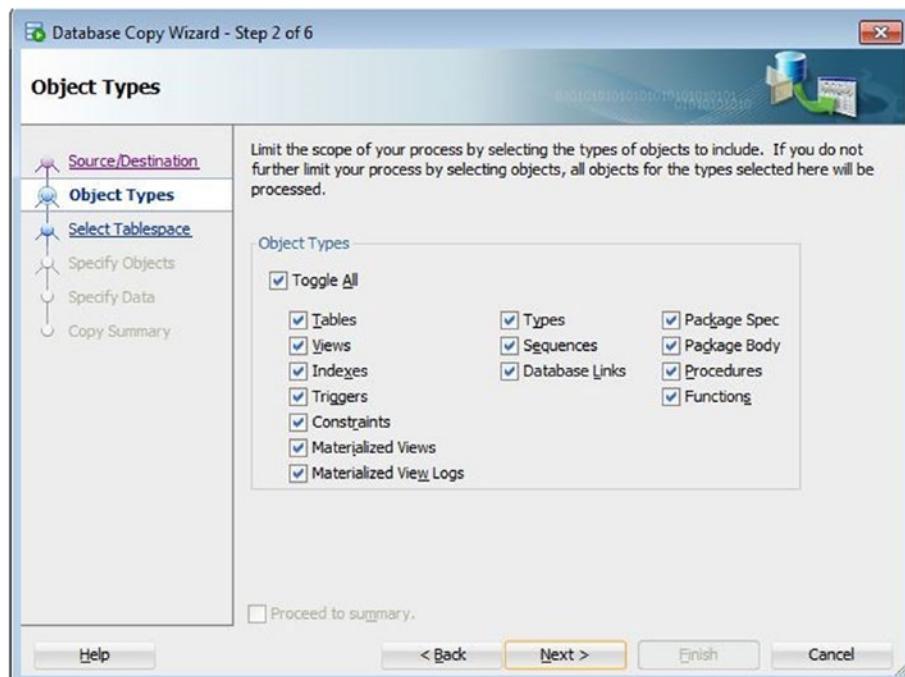


Figure 7-8. Database copy wizard: select object types

In this case, you will choose the schema copy option. You can select all or specific object types, as per your requirements.

The next step is to select the appropriate schemas (Figure 7-9).

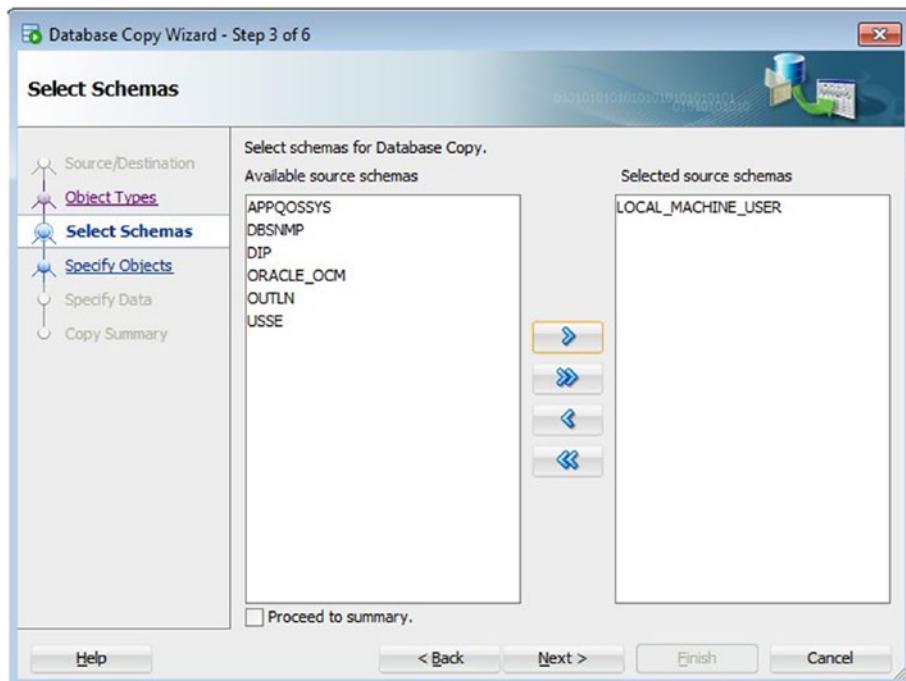


Figure 7-9. Database copy wizard: select schemas

If it fails with an error that states that the schema does not exist on the target machine, then create the required schema in AWS database and try again. The same schema must exist at the target and it will not get created automatically.

Next is a summary of the data copy selections chosen so far, as shown in Figure 7-10.

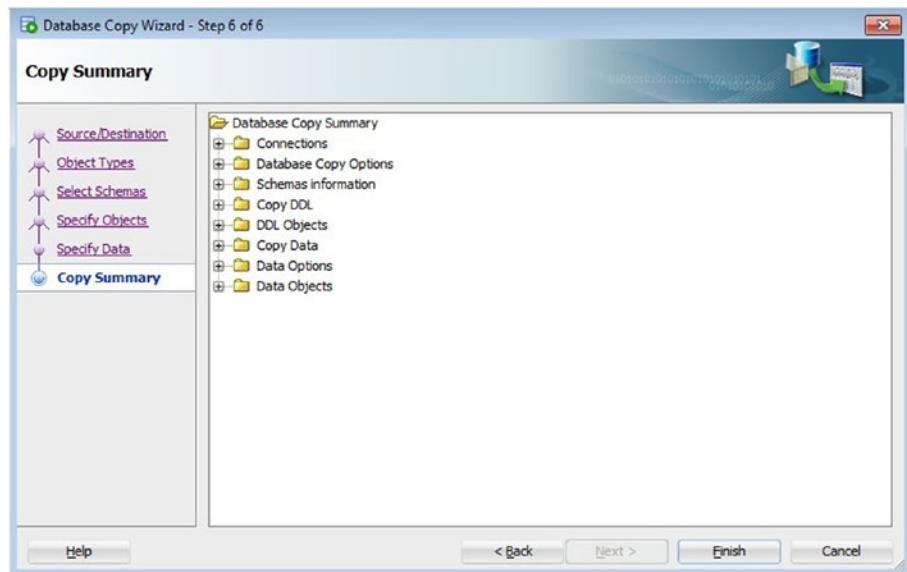


Figure 7-10. Database copy wizard: summary screen

After clicking Finish, a data copy job starts and gives you the option to let it run in the foreground or in the background (Figure 7-11).

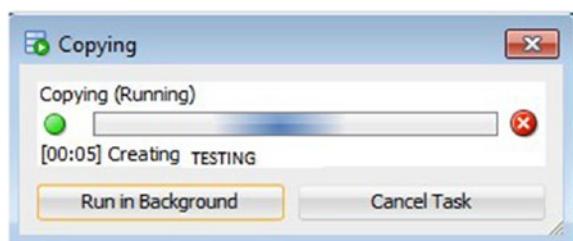


Figure 7-11. Database copy

You can see other copy-related stats from the log file that's generated. It is shown here:

```
--- START -----
Moving Data for object TESTING

Insert 128,670 rows into TESTING in 90,421 milliseconds
--- END -----
--- END -----
--- START -----
set define off;

set define off;

--- END -----
```

The record count and actual data should be verified at the cloud DB. The target DB data should match the local DB data for the data migration to be successful. In case of any discrepancies, specific errors need to be fixed by rerunning a partial or full process, depending on the situation.

The data copy time increases with the size of the table and the amount of data to be copied and hence this method is good for smaller tables.

Detailed Migration Steps for Data Migration Using Oracle Data Pump Utility

Data pump based migration can be used for bigger tables and schemas.

Once the dump file is created, you can create a DB link from the local source DB to the target cloud DB. This DB link will be used to place the data on the cloud DB's filesystem.

```
tnsping testdb.xxxx.yyyyyy.rds.amazonaws.com:1521/TESTDB

TNS Ping Utility for 32-bit Windows: Version 11.2.0.1.0 - Production on 02-AUG-2016 19:17:11

Copyright (c) 1997, 2010, Oracle. All rights reserved.

Used parameter files:
D:\app\Niraj\product\11.2.0\dbhome_1\network\admin\sqlnet.ora

Used EZCONNECT adapter to resolve the alias

Attempting to contact
(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=TESTDB))(ADDRESS=(PROTOCOL=TCP)(HOST=11.22.3
3.199)(PORT=1521)))

OK (50 msec)
```

Now create a DB link using this connection string.

```
create database link Cloud_DBA_rds connect to CLOUD_USER identified by secret_pwd using
'(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=TESTDB))(ADDRESS=(PROTOCOL=TCP)(HOST=11.22.3
3.199)(PORT=1521))';
```

You should test the connectivity using the DB link, which was created in the previous step.

On the local DB:

```
SQL> select table_name from user_tables@cloud_dba_rds;
No rows selected.
```

On the cloud DB:

```
SQL> grant read,write on directory data_pump_dir to cloud_user;
Grant succeeded.
```

You are now ready to transfer the export dump file to the remote system. This is done using the DBMS_FILE_TRANSFER package.

```
BEGIN
DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object    => 'DATA_PUMP_DIR',
    source_file_name          => 'test_local.dmp',
    destination_directory_object => 'DATA_PUMP_DIR',
    destination_file_name     => 'tab1_copied.dmp',
    destination_database      => 'cloud_dba_rds'
);
END;
/
PL/SQL procedure successfully completed.
```

As you do not have access to the filesystem of AWS RDS, you cannot verify the availability of the dump file directly. However, there are some built-in packages available to check it. It is shown here:

On the cloud DB:

```
SQL> select * from dba_directories where DIRECTORY_NAME='DATA_PUMP_DIR';
OWNER          DIRECTORY_NAME   DIRECTORY_PATH
-----  -----
SYS            DATA_PUMP_DIR           /rdsdbbin/datapump

SQL> select * from table(RDSADMIN.RDS_FILE_UTIL.LISTDIR('DATA_PUMP_DIR'));

filename        type        filezise       mtime
-----  -----
datapump/      directory      4096      0      02-OCT-16
tab1_copied.dmp  file        3211264      0      04-OCT-16
```

On the local DB, create a TNS entry of the remote DB in the local system so that you can import the data in the remote DB using this entry. After creating a TNS entry, you can do a tnsping test and then a connection test, as shown here:

```
TEST_CLOUD_DB =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 11.22.33.199) (PORT = 1521))
  (CONNECT_DATA =
    (SID = TESTDB)
  )
)
```

tnsping TEST_CLOUD_DB

TNS Ping Utility for 32-bit Windows: Version 11.2.0.1.0 - Production on 02-OCT-2016 20:38:52

Copyright (c) 1997, 2010, Oracle. All rights reserved.

Used parameter files:

D:\app\Niraj\product\11.2.0\dbhome_1\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias

Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = 11.22.33.199) (PORT = 1521)) (CONNECT_DATA = (SID = TESTDB)))

OK (30 msec)

sqlplus cloud_user/xxxxx@TEST_CLOUD_DB

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter password:

Connected to:

Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production

*With the Partitioning, Oracle Label Security, OLAP, Data Mining
and Real Application Testing options*

19:23:05 02-Oct-16 CLOUD_USER @TEST_CLOUD_DB>

Now import the data in the cloud DB with a user who has the required privileges.

On the local DB:

```
C:\windows\system32>impdp cloud_user/<password>@TEST_CLOUD_DB
DUMPFILE=tab1_copied.dmp DIRECTORY=DATA_PUMP_DIR full=y

Import: Release 11.2.0.1.0 - Production on Tue Aug 2 20:43:25 2016

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining
and Real Application Testing options

Master table "CLOUD_SER"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "CLOUD_USER"."SYS_IMPORT_FULL_01": cloud_user/********@TEST_COLUD_DB
DUMPFILE=tab1_copied.dmp DIRECTORY=DATA_PUMP_DIR full=y
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/TABLESPACE_QUOTA
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
.. imported "USER1"."TAB1"           2.905 MB  71692 rows
Job "CLOUD_USER"."SYS_IMPORT_FULL_01" successfully completed at 08:17:36
```

Verify the data in the cloud DB using SQL Developer. While copying the dump file may take more time, the import is expected to be instantaneous since it is done locally in the cloud. In case of data mismatch, specific table/objects may need to be imported again. The exact steps for fixing the partially successful migration depend on the specific issues encountered.

Sample Comparison Between SQL Developer and Data Pump Migration

Table 7-2 shows the performance comparison between SQL Developer and data pump-based migration methods.

Table 7-2. Comparison Between SQL Developer and Data Pump Performance

Migration Method	Object Size (MB)	Object Type	No. of Records	Time Taken
SQL Developer Database Copy	12.6	Table	128,670	1.6 minutes
SQL Developer Database Copy	120	Table	716,484	21 minutes
SQL Developer Database Copy	640	Table	3,715,500	124 minutes
Data Pump	2.91	Table	71,692	0.4 minutes
Data Pump	508	Schema with three objects	1,852,320	14 minutes

As you can see, the data pump-based migration method is faster than the SQL Developer-based database copy option. These results depend on many factors, such as server configuration, network speed, etc., hence it is recommended you perform a proof of concept (PoC) in your environment to arrive at a proper conclusion. This PoC is usually done as part of the pilot phase of migration.

Migration Using RMAN Backup Restore

This migration method is good for scenarios when you want to migrate your entire database from on-premises to the cloud. This migration method can be executed in two ways.

Option 1

1. Take a RMAN backup on the local server.
2. Transfer the RMAN backup from the local server to cloud storage.
3. Restore from the RMAN backup.

Option 2

1. Take a RMAN backup on the cloud.
2. Restore from the RMAN backup.

RMAN backup restore steps are the same ones used with on-premise DBs. Refer to Chapter 8, “Backup and Restore” for the detailed steps.

Migration By Cloning a PDB into the Cloud

In Oracle database 12c, Oracle introduced a multitenant architecture and this made PDB-level migration a very easy task. In this section, you will look at the steps that are required to clone an on-premise PDB and plug it into a cloud-based CDB.

The objective is to migrate a LocalPDB (Oracle EE 12.1.0.2, DB Size ~1GB) running on Windows to Oracle Public Cloud.

Tools used:

- SQL Developer
- WinSCP
- PuTTY
- PuTTYKeyGen
- EM Express (provided as part of Oracle Cloud)

The high-level steps are as follows:

1. Multitenant setup on-premise (local).
2. Create Oracle 12c database in Oracle Public Cloud.
3. Clone LocalPDB (optional).
4. Open the cloned LocalPDB in READWRITE mode (optional).
5. Close LocalPDB.
6. Convert LocalPDB into READONLY mode.
7. Unplug LocalPDB.
8. Copy the .XML and data files to the cloud DB server.
9. Use EM Express to plug LocalPDB into the cloud.
10. Open LocalPDB in the cloud.

Let's look at these steps in detail.

Multitenant Setup On-Premise (Local)

Since this example is about moving a PDB to the cloud, you must have a multitenant DB in your source DC. If you are doing this migration for POC purposes, this step is applicable. As part of this, you need to install Oracle 12c Enterprise Edition and then select Create and Configure a Database, as shown in Figure 7-12.

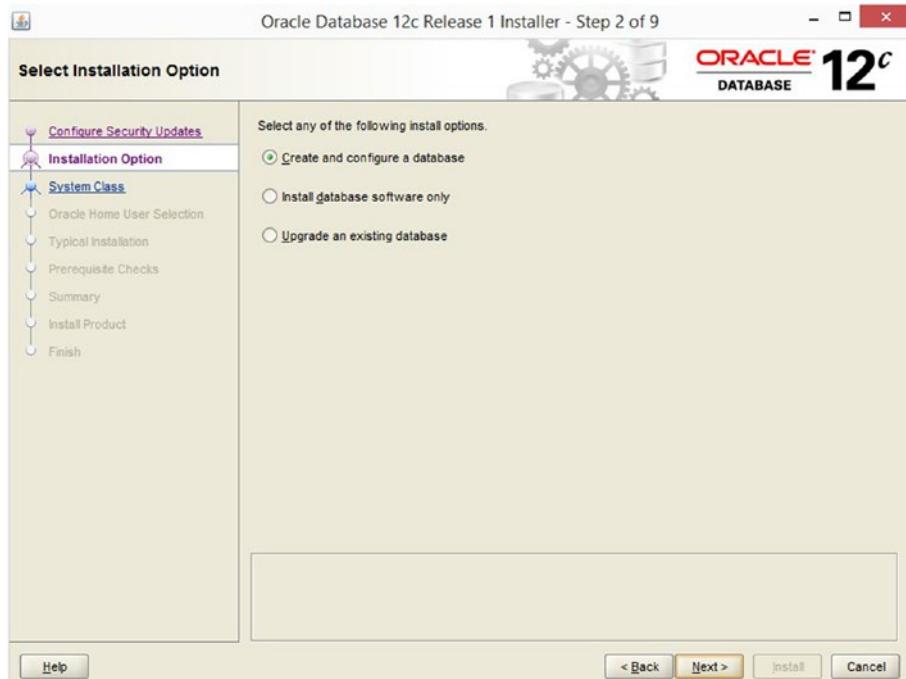


Figure 7-12. Select the installation option

Continue with the database creation steps and select the configuration as per your requirements. Selections made for a typical installation are shown in Figure 7-13.

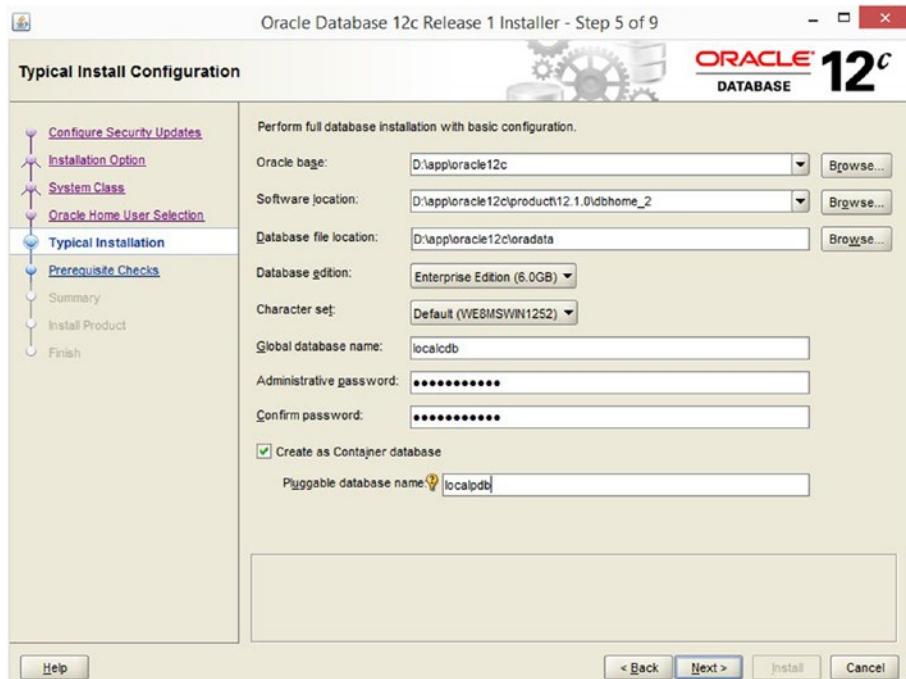


Figure 7-13. Create a container database

Create Oracle 12c PDB Database in Oracle Public Cloud

This setup is explained in Chapter 3, “Database Provisioning”. Refer to the “Database Provisioning in Oracle Cloud” section.

Clone LocalPDB (Optional)

This step is required if you want to create a copy of your local PDB database and move the copy instead of moving the original PDB to the cloud. If you are moving your original PDB to the cloud, then this step and the next aren’t required.

Cloning the PDB can be done using SQL Developer. Right-click on the pluggable database for which you need to create a clone. This is shown in Figure 7-14.

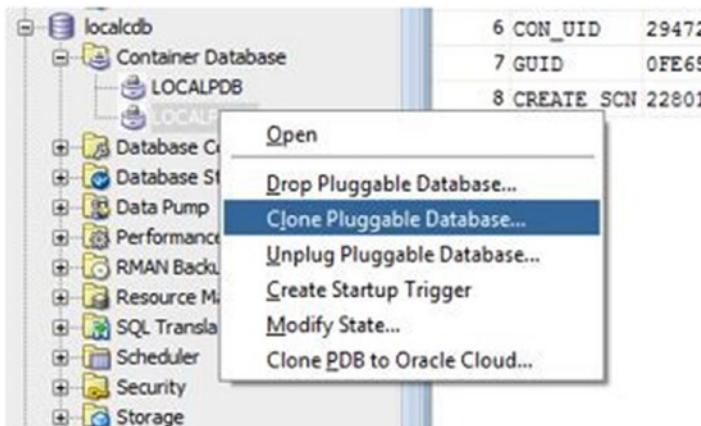


Figure 7-14. Select *Clone Pluggable Database*

Give the new PDB a name and select Custom Names from the File Name Conversions option. This is shown in Figure 7-15, where the old PDB name was LocalPDB and cloned PDB name is LocalPDB2.

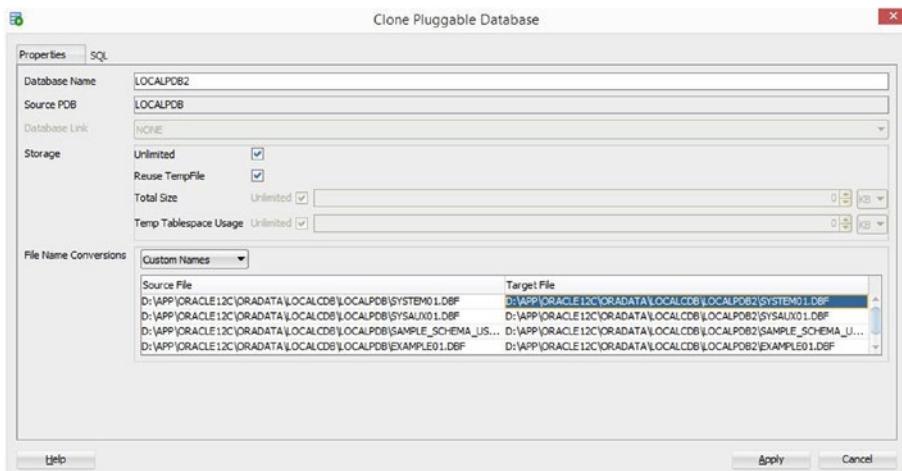


Figure 7-15. *Clone the pluggable database*

Open the Cloned LocalPDB in READWRITE Mode (Optional)

This step ensures that the cloned PDB (LocalPDB2) is working as expected. Right-click on LocalPDB2 and select Modify State. Select the State option as READ WRITE and click on Apply, as shown in Figure 7-16.

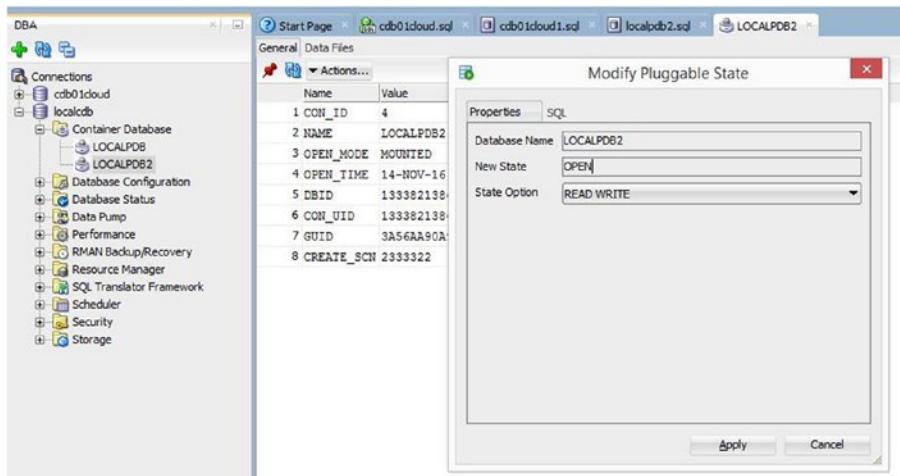


Figure 7-16. Open the cloned database

Confirm that the changes are applied and that the OPEN_MODE for this PDB is now set to READ WRITE (Figure 7-17).

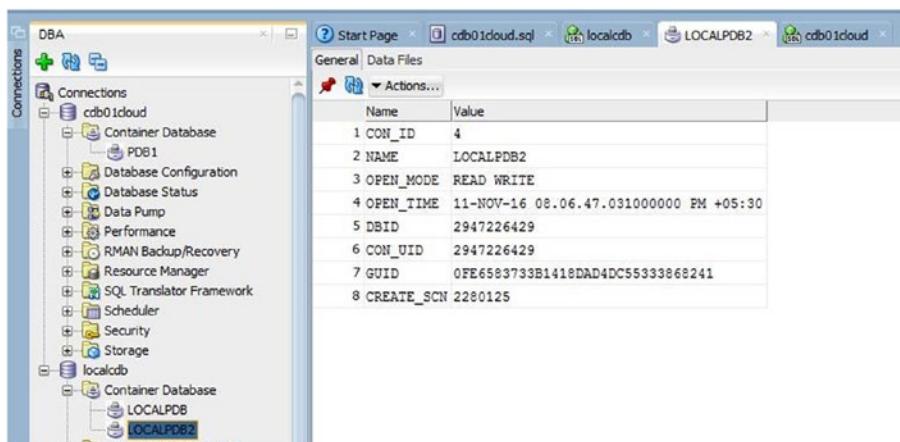


Figure 7-17. Cloned database status

Close LocalPDB

Close LocalPDB (or LocalPDB2, whichever is your case) to get it ready for unplugging.

Right-click on LocalPDB2 and select Modify State. If the database is already open in READWRITE, you will see the CLOSE IMMEDIATE option, as shown in Figure 7-18. Click on Apply to close the PDB.

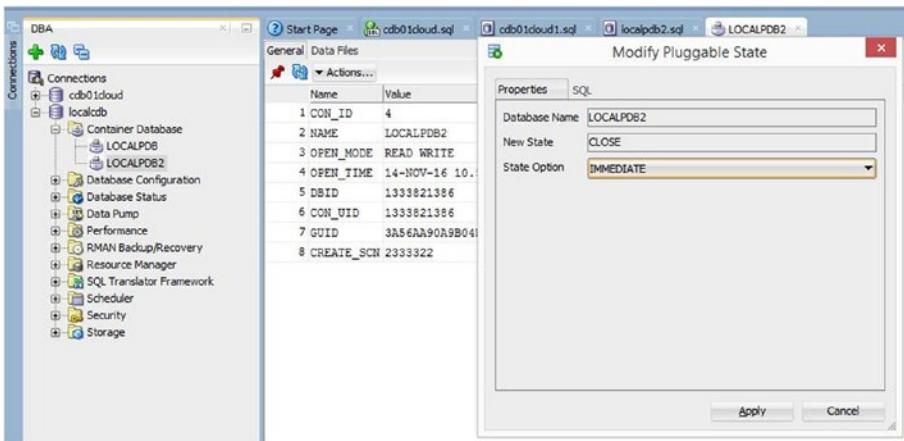


Figure 7-18. Close the database

Convert LocalPDB into READONLY Mode

Now convert LocalPDB2 to READONLY (Figure 7-19) mode so that it can be unplugged. This is required, as the data file headers need to be consistent for successful restoration of the PDB.

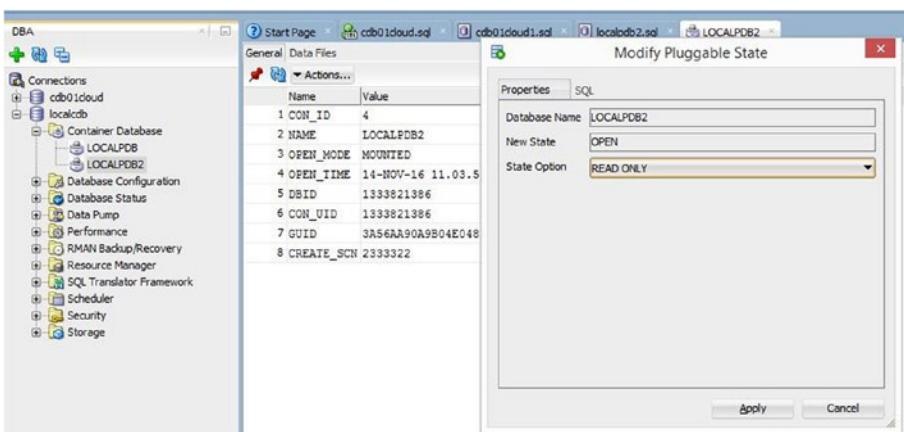


Figure 7-19. Change the database state to READONLY

Unplug LocalPDB

After converting the database into READONLY mode, you are all set to unplug it. To unplug it, right-click on LocalPDB2 and select Unplug Pluggable Database. Give the location of the .XML file and click on Apply, as shown in Figure 7-20.

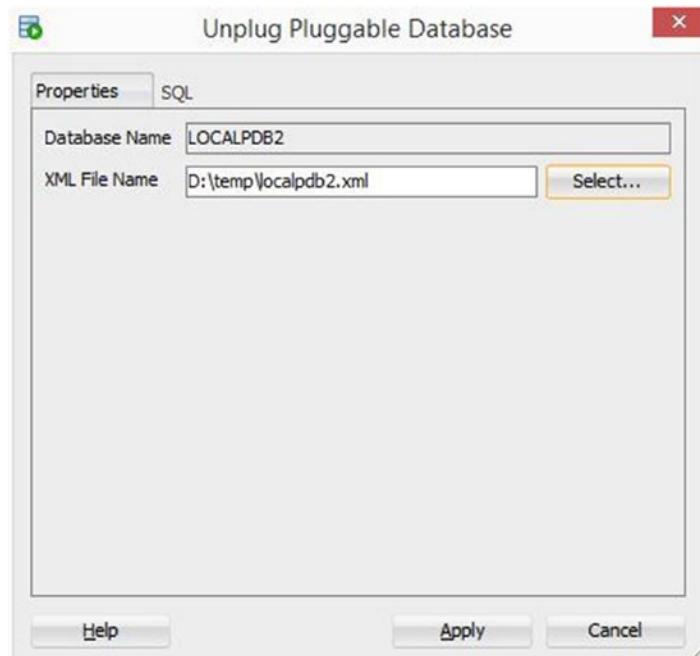


Figure 7-20. Unplug the pluggable database

Copy the .XML and Data Files to the Cloud DB Server

To plug this database into the cloud CDB, you need to copy the .XML and data files to the cloud DB server. For this, you can use either do scp from your UNIX server (if the local DB is running on UNIX) or you can use the WinSCP tool to transfer these files from a Windows server. Let's see how it is done in Windows.

Open WinSCP and enter the Cloud DB server's public IP. For the username, give opc and don't enter a password (Figure 7-21).

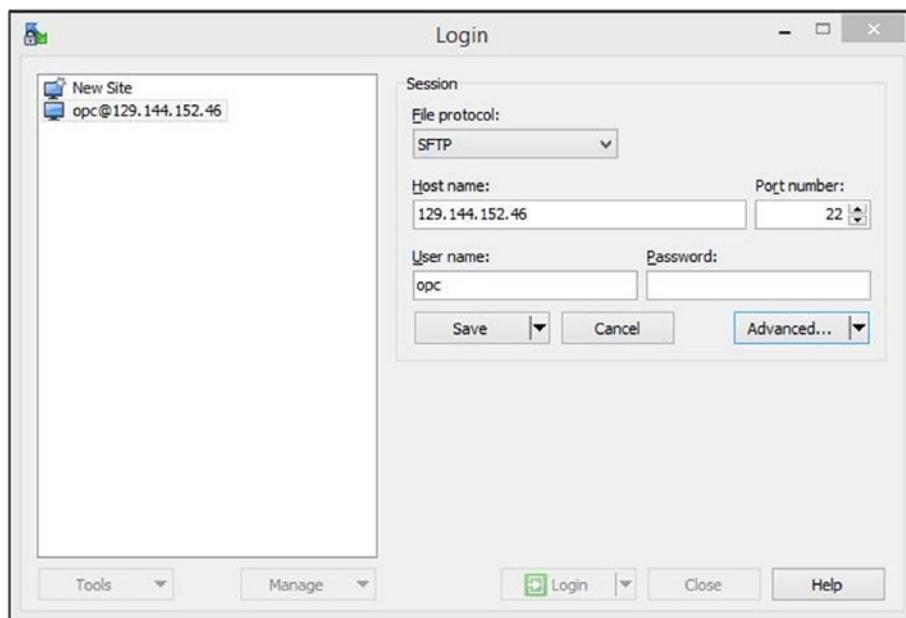


Figure 7-21. Connecting the Cloud DB server using WinSCP

Now click on Advanced and go to SSH ► Authentication. Enter your private key file location. This is shown in Figure 7-22. Without the public-private key pair, the connection will not be allowed for security reasons.

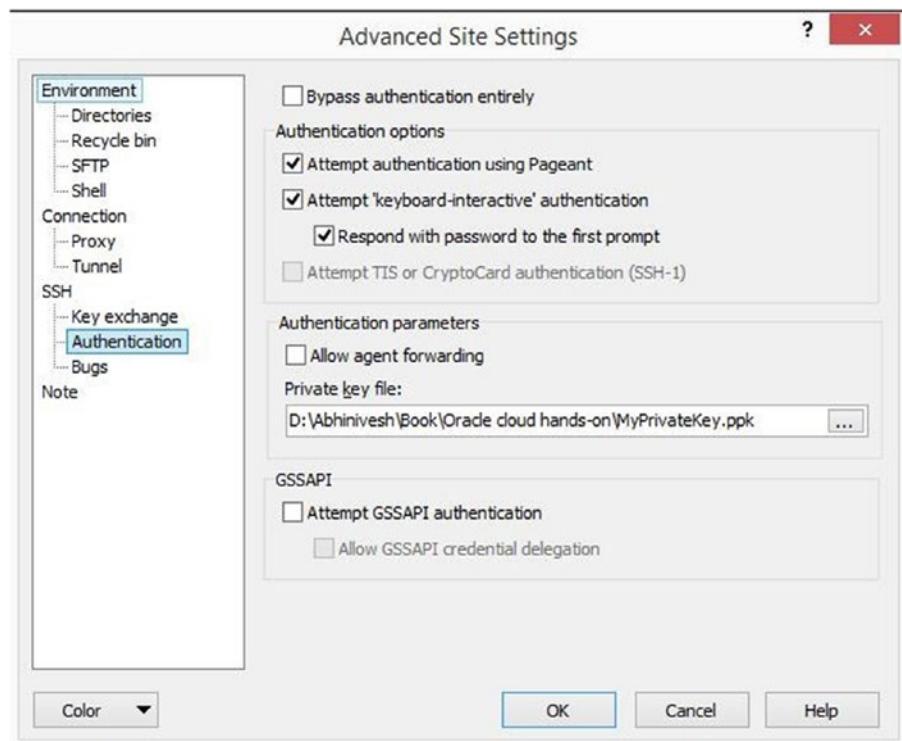


Figure 7-22. Specifying a private key file in WinSCP

Now you can see the local files on left side and the cloud directory on the right. With this, you are ready to drag and drop the files to Oracle Cloud DB server. Your screen will look like Figure 7-23.

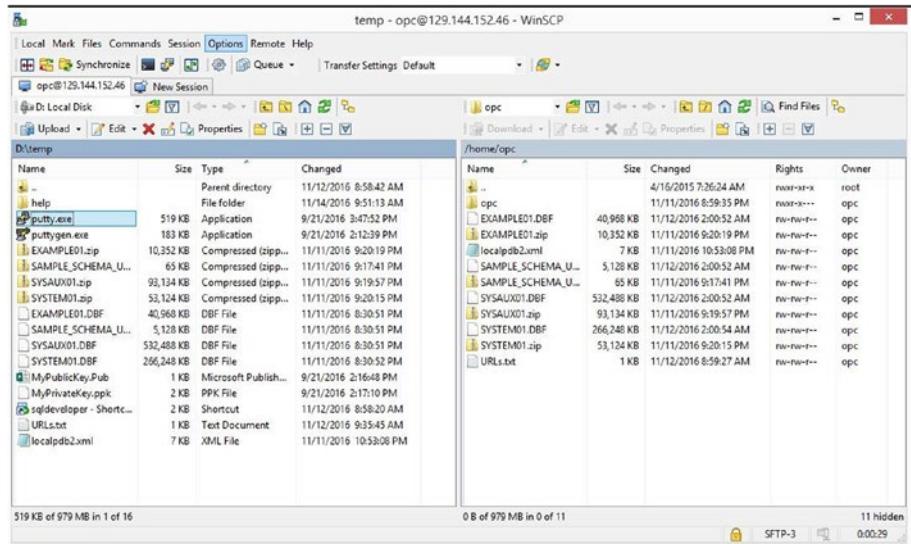


Figure 7-23. WinSCP file transfer screen

Since you are copying as the opc user, you need to move these files to Oracle user directories by logging in as the root. You will also need to change the ownership of the file and change the file permissions. These steps are shown in Figure 7-24.



The screenshot shows a terminal window titled "root@CDB02:~". The terminal displays the following command-line session:

```
login as: opc
Authenticating with public key "rsa-key-20160921"
[opc@CDB02 ~]$ sudo su -
[root@CDB02 ~]# ls /home/opc
EXAMPLE01.DBF      opc          SYSAUX01.DBF  SYSTEM01.zip
EXAMPLE01.zip      SAMPLE_SCHEMA_USERS01.DBF  SYSAUX01.zip  URLs.txt
localpdb2.xml       SAMPLE_SCHEMA_USERS01.zip  SYSTEM01.DBF
[root@CDB02 ~]# cp SYSTEM01.DBF /home/oracle/
cp: cannot stat `SYSTEM01.DBF': No such file or directory
[root@CDB02 ~]# cp /home/opc/SYSTEM01.DBF /home/oracle/SYSTEM01.DBF
[root@CDB02 ~]# chown oracle:oinstall /home/oracle/SYSTEM01.DBF
[root@CDB02 ~]# chmod 750 /home/oracle/SYSTEM01.DBF
[root@CDB02 ~]# ls -l /home/oracle/SYSTEM01.DBF
-rwxr-x--- 1 oracle oinstall 272637952 Nov 14 06:35 /home/oracle/SYSTEM01.DBF
[root@CDB02 ~]#
```

Figure 7-24. File copy and permissions

It is recommended you compress the files before transferring them so that the transfer time is reduced.

Using EM Express to Plug LocalPDB into the Cloud

To plug LocalPDB into the Cloud CDB, open EM Express, as shown in Figure 7-25.

The screenshot shows the Oracle Database Cloud Service EM Express interface. On the left, there's a navigation sidebar with sections for Overview (1 Node), Administration (0 Patches available, 0 Snapshots available), and Summary (1 Nodes). The main area displays the CDB02 instance details: Public IP: 129.144.152.46, SID: CDB02, PDB Name: PDB1, SQL*Net Port: 1521, OCPU: 1, Memory: 7.5 GB, and Storage: 150 GB. A timestamp at the bottom right indicates "As of Nov 14, 2016 6:46:41 AM UTC". A context menu is open over the storage information, listing options like "Open DBaaS Monitor Console", "Open Application Express Console", "Open EM Console", "SSH Access", "Access Rules", and "Replace Database using Backup".

Figure 7-25. Open the EM console

You will see a screen similar to Figure 7-26. Here you can see that CDB02 has one PDB available. This is the default PDB that was created during the provisioning. To plug the on-premise PDB into the CDB02, click on the CDB hyperlink just above the version information.

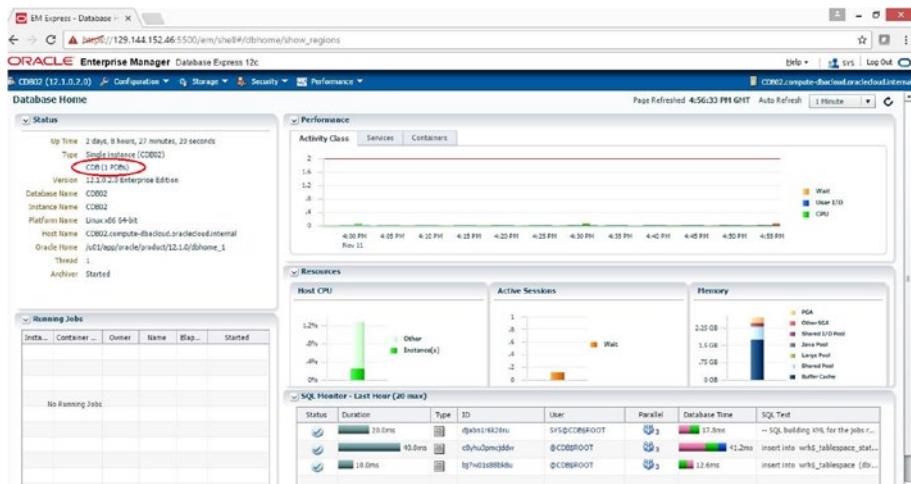


Figure 7-26. EM console: home page

This should bring up the page shown in Figure 7-27. It shows all the PDBs that are associated with this CDB. It also gives you an option to plug in the PDB.

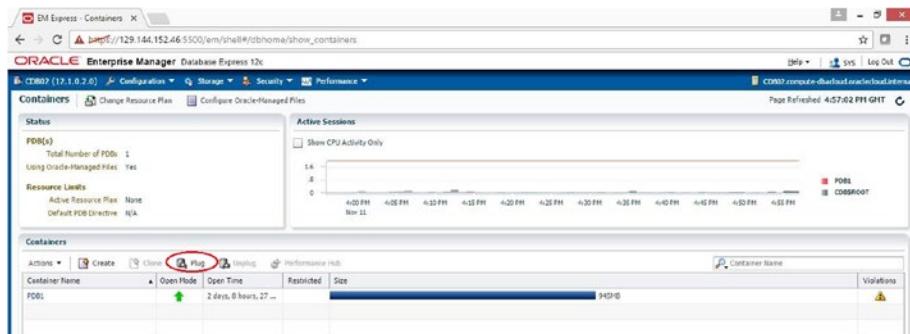


Figure 7-27. EM console: CDB home page

Now click on Plug and provide the metadata file and source file location. Uncheck the Reuse Source Data File Location from Metadata File option. Click on OK to start PDB plugging, as shown in Figure 7-28.

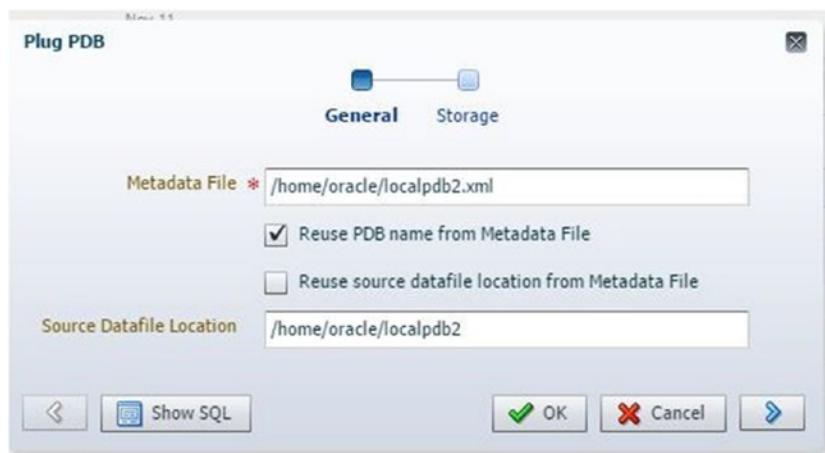


Figure 7-28. EM console: plug PDB

This will take few minutes and you will see a successful completion message when it's done.

Open LocalPDB in the Cloud

Now it's time to bring up the newly moved LocalPDB in the cloud. It is associated with the CDB Cloud, but is not yet ready to use. To open the newly plugged local PDB in the cloud, you need to select LocalPDB2 and click on Actions ► Open.

The newly plugged PDB is now open and ready to use, as shown in Figure 7-29.

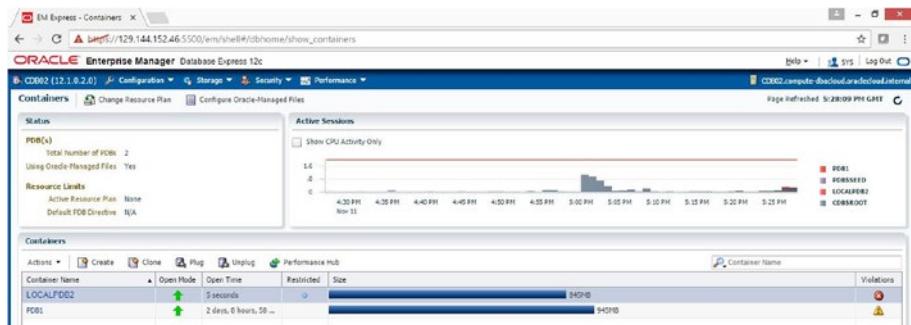


Figure 7-29. EM console: open PDB

Your LocalPDB2 is now up and running under Cloud CDB. With this, migration of a local PDB is successful and it is now available in the cloud.

Migration Using DMS

Amazon provides the Database Migration Service (DMS). You can select this service from the Database section in the AWS Console (Figure 7-30).

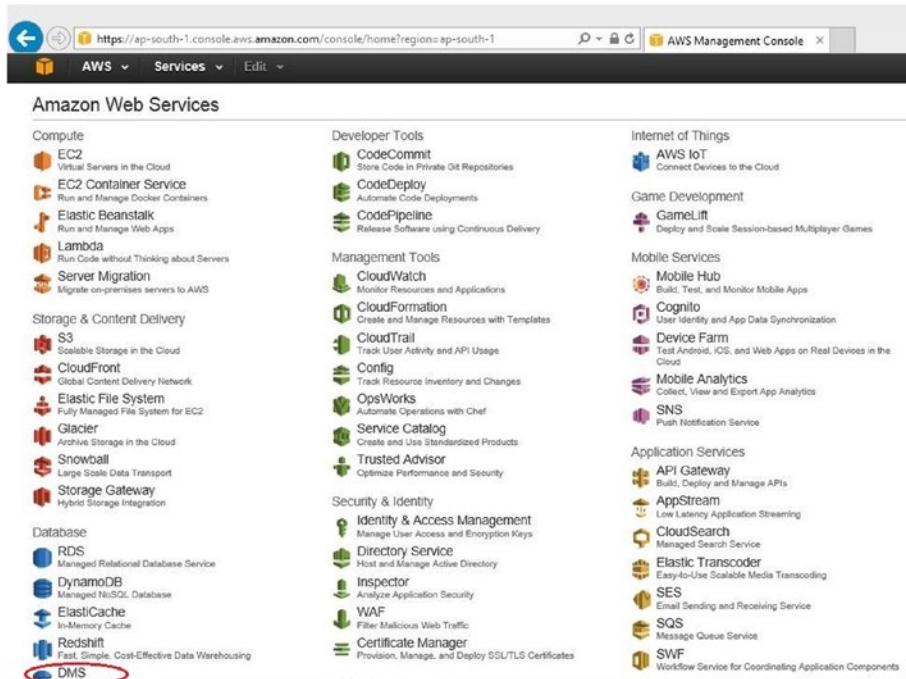


Figure 7-30. DMS in the AWS Console

After selecting DMS, you will see the screen shown in Figure 7-31. In the bottom-right corner of this screen, you will see an option for downloading the AWS Schema conversion tool. This tool is locally installed and can be used for database migration, not only within Oracle database but also with cross-RDBMS migration (for example, migration from MySQL to Oracle).

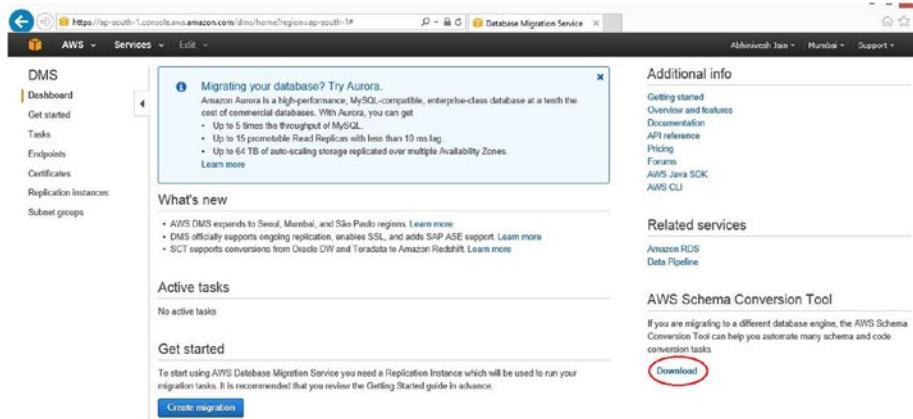


Figure 7-31. AWS schema conversion tool download

You can download it for your OS, as shown in Figure 7-32.

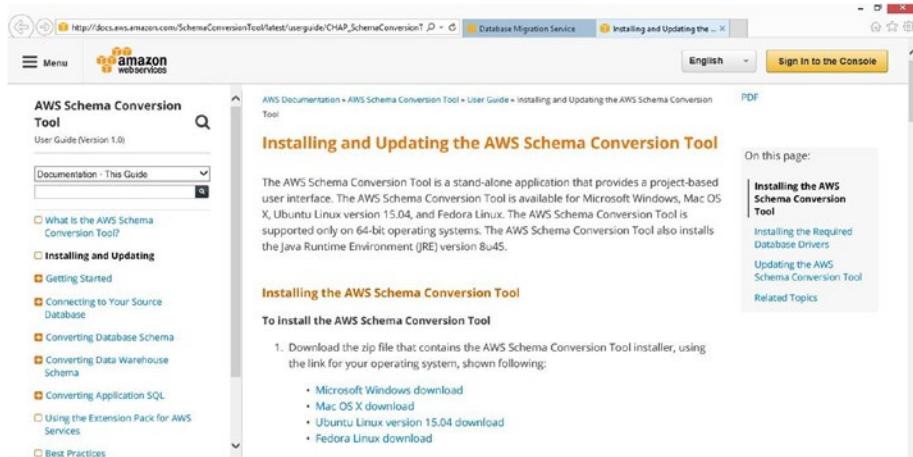


Figure 7-32. AWS schema conversion tool installation

The schema conversion tool is very small (116MB), so the installation and configuration process is quick. After installation, you can invoke the tool to perform schema conversion analysis, as shown in Figure 7-33.

To do this, choose File ► Create New Project.

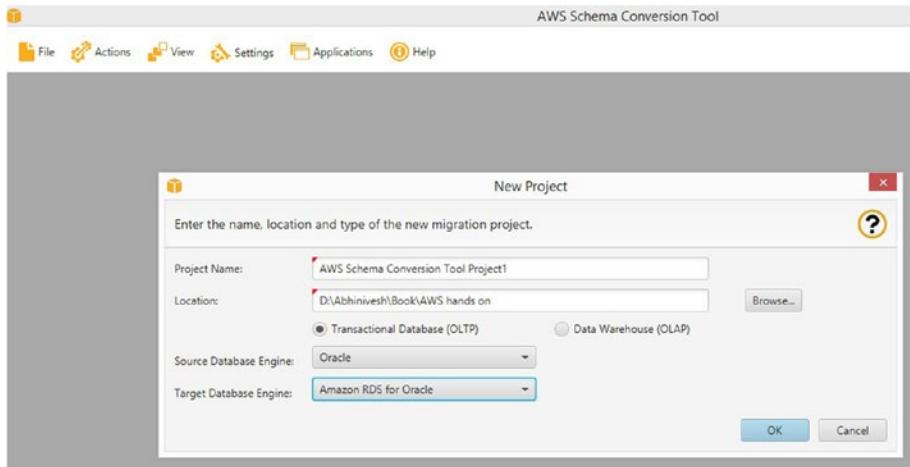


Figure 7-33. AWS schema conversion tool: new project creation

This tool also requires Oracle JDBC driver for execution. You can download this driver from OTN. The download location of various drivers is given on the AWS documentation page, as shown in Figure 7-34.

Database Engine	Download
Amazon Aurora	https://www.mysql.com/products/connector/
Amazon Redshift	http://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html
Microsoft SQL Server	https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774
MySQL	https://www.mysql.com/products/connector/
Oracle	http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html
PostgreSQL	https://jdbc.postgresql.org/download.html
Teradata	https://downloads.teradata.com/download/connectivity/jdbc-driver

Figure 7-34. AWS schema conversion tool: database driver download

After download, you need to specify the driver location in the global settings shown in Figure 7-35.

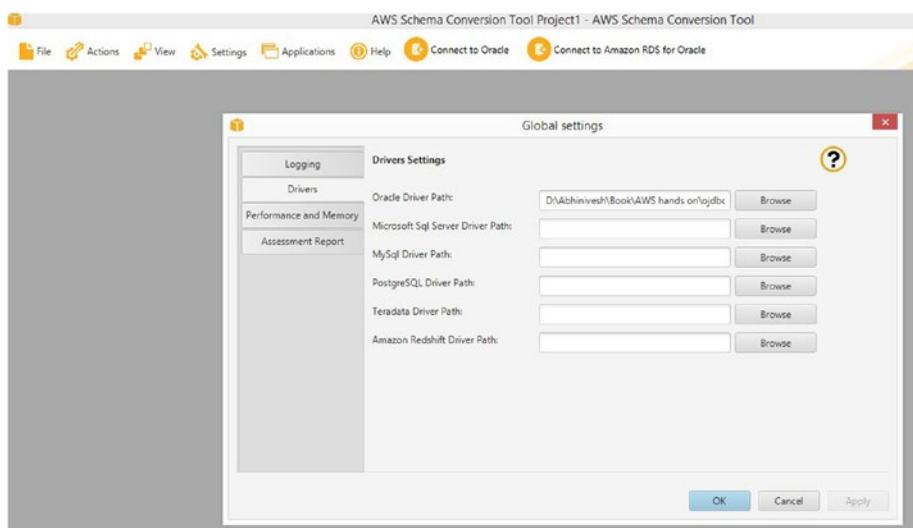


Figure 7-35. AWS schema conversion tool: drivers setting

Now you are all set to run the analysis.

Migration Using AWS Snowball

Cloud adoption and migration is a phased process. After testing is completed and you are happy with running smaller databases in the cloud, it is time to move your larger databases to the cloud.

When porting your data over a network is a tough and time-consuming task due to data size and bandwidth issues, Snowball from AWS comes to the rescue. It is a travel bag-like device that secures and weatherproofs your data during transport. Snowball's storage capacity and service availability are growing and it's expected to be available globally by the start of 2017. As of writing this book, the single largest Snowball is 50/80TB (and it's growing) and AWS Snowball is available in 9 out of 16 regions.

You can order Snowball from AWS by creating a data transfer job in the AWS Management Console and connecting multiple Snowball devices to load your large databases running in Terabytes. The data is stored in a tamper-resistant case and is encrypted. After loading the data, you need to ship this device to AWS. You will be notified once your data is available. There are APIs available for integration with existing storage, for larger adoption of Snowball. Snowball also has S3 connectivity for moving data in and out of AWS without having to depend on network bandwidth.

Figure 7-36 shows the simple workflow involved in migrating a database to AWS using Snowball.

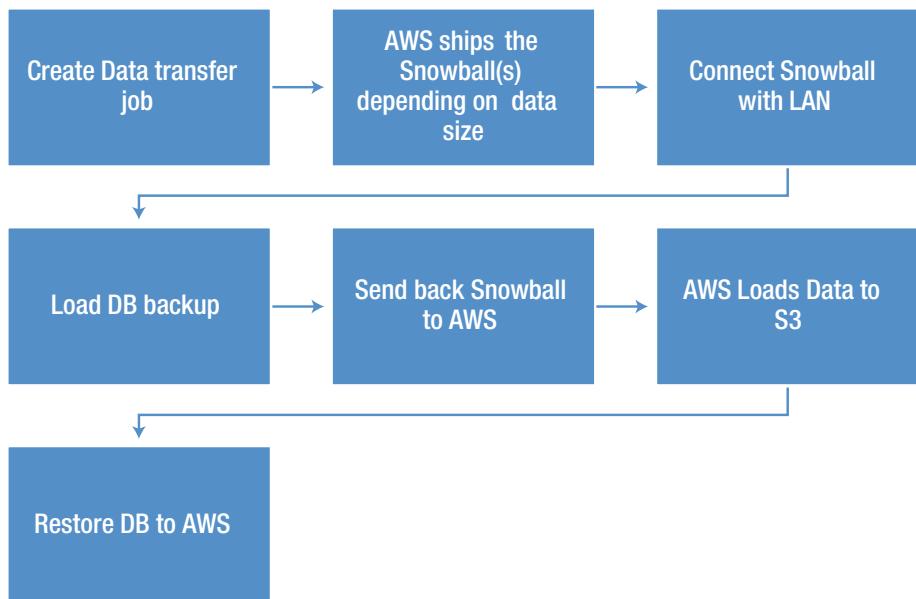


Figure 7-36. Database migration flowchart using Snowball

Here are some of the reasons why you should use Snowball:

- No network overloading
- Works where bandwidth is an issue
- Secure
- Fast
- Large capacity
- Cost effective

Once your data is made available using Snowball, you can restore your database like any other storage operation. Snowball can be used to get data back home as DR or to move away from AWS.

There are some limitations when working with Snowball, like data transfer time limit of 90 days, region-based service availability, and some encryption-related limitations.

Migration Best Practices

This section lists some of the best practices for Oracle database migration to the cloud.

- Choosing the correct cloud host configuration as per requirements is vital to avoiding future scaling and performance issues.
- Setting the DB configuration parameters correctly yields required performance for Managed Cloud DB services.
- Setting tight security settings to provide only the required access and enable only required ports is a must.
- Understanding the limitations of unsupported features like active DataGuard between two cloud DBs will help in planning migration.
- Identifying and segregating responsibilities between the cloud team and the client team is necessary for smoother functioning.
- Understanding the licensing management is very much required when migrating to the cloud, e.g., BYOL versus pay-as-you-go.
- Excluding DBs with strict performance requirements and different compliance and regulatory requirements.

Summary

Database migration is a complex and challenging process. It requires analysis and detailed planning to accomplish a successful migration. In this chapter you learned about key considerations and lifecycle stages for database migration to the cloud. You are now aware of more than 10 methods for migrating to the cloud. This chapter also covered the available migration tools and explained how to use them. You read about migration using cloning a PDB into the cloud and AWS-specific options like using DMS and Snowball.

A single migration method might not be suitable for migrating your entire DB landscape to the cloud, so you should identify a mix of migration methods to handle different needs.

In the next chapter, you will learn how the backup and recovery process works for databases in the cloud.

CHAPTER 8



Backup and Restore

This chapter covers the database backup and restore options in Oracle Cloud and AWS. The backup and restore processes are vital for business continuity and the DBA has the responsibility to ensure that the databases are recoverable as per the requirements. This chapter covers various options available for backing up your database to the cloud. It also provides step-by-step details on how to move a local backup to the cloud.

Overview of Database Backup and Restore from the Cloud Perspective

Successful database backup is the most important job of any DBA, hence it is extremely important for DBAs to know their options while using the cloud. As a DBA, you can use the cloud in following two ways for your backup and restore requirements:

- Use the cloud as a backup destination for an on-premise database
- Use the backup and restore options (within the cloud) for the cloud database

As a DBA, your responsibilities vary according to the DB hosting model. If you are using an IaaS kind of service, such as Oracle database running on an Amazon EC2 instance, then you are responsible for backing up your database or even the entire EC2 instance.

If you are running database in managed service kind of offering, such as Amazon RDS or Oracle Schema as a service then backup and restore is very easy task and it is internally managed by the cloud provider. In this case, your responsibility is limited to specifying the maintenance window and retention period of backup. This is explained in detail in the following sections.

In the following sections, you will go through all the available backup and restore options. Let's first look at these two categories.

Using the Cloud as a Backup Destination for an On-Premise Database

Backups are traditionally first taken at disk and then moved to tape and tapes are then moved to an offsite location for long-term storage and disaster prevention. In the new model, the on-premise DB backup goes directly to the cloud (Figure 8-1).

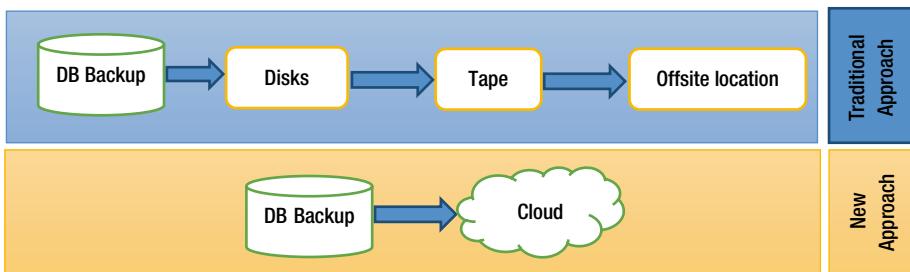


Figure 8-1. On-premise DB backup approaches

There are many benefits to such a cloud-based backup approach:

- No CAPEX
- No OPEX for tape maintenance-related activities
- Anywhere/anytime access of backups
- Faster restore time (compared to cases, where tape has to be recalled from an offsite location)
- On-demand capacity increase
- Secure
- Encrypted
- Compressed

Oracle provides a cloud backup module for both Oracle Cloud and Amazon AWS to integrate RMAN backup with cloud storage. Using this approach, you can run simple RMAN commands to save on-premise DB backups to the cloud.

Using Backup and Restore Options (Within the Cloud) for a Cloud Database

For a database hosted in the cloud, you can use the backup and restore options provided by cloud providers. Based on these two categories, all available methods can be classified as shown in Table 8-1.

Table 8-1. Cloud Backup Offering Comparison

Category/Cloud Provider	Oracle Cloud Offering	Amazon AWS Offering
Moving On-Premise Backup to the Cloud	<ul style="list-style-type: none"> • Oracle Database Cloud Backup Service 	<ul style="list-style-type: none"> • Amazon S3
Backup Restore within the Cloud	<ul style="list-style-type: none"> • DB Snapshots • DB Backups 	<ul style="list-style-type: none"> • DB Backup and Restore in Amazon RDS • DB Backup and Restore in Amazon EC2

Besides these options, you can also perform the backup operations using Oracle Enterprise Manager Cloud control 13c and third-party tools such as CloudBerry Lab's cloud backup solution.

Details of these six methods are discussed in the next two sections under Oracle cloud and AWS. To differentiate the methods according to the cloud service providers, the methods are grouped based on the cloud providers rather than on the two categories mentioned before.

DB Backup and Restore in Oracle Cloud

In Oracle Cloud, different options exist for different requirements:

- *Oracle Database Cloud Backup Service*: This method is for taking an on-premise DB backup to the cloud.
- *DB Snapshots*: This method is for taking a cloud database backup for cloning.
- *DB Backups*: This method is for an on-demand backup and recovery.

Let's look at these options in detail.

Oracle Database Cloud Backup Service

This is also known as Oracle Database Backup Service, or ODBS. It is a three-step process.

1. Subscribe to Oracle Cloud Backup service.

You can do so by going to the following link and selecting either Sign In or Try It.

https://cloud.oracle.com/database_backup

After receiving a confirmation mail, activate your service. You are ready to move to the next step.

2. Download and configure Oracle database cloud backup module.

To download Oracle Cloud Backup Module, visit the OTN link shown in Figure 8-2.

<http://www.oracle.com/technetwork/database/availability/oracle-cloud-backup-2162729.html>

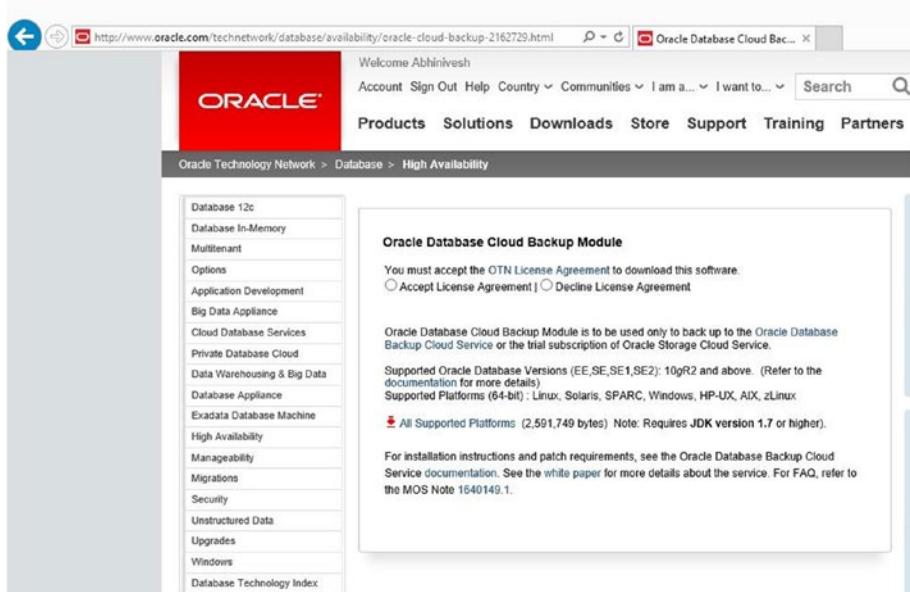


Figure 8-2. OTN page for Oracle Database Cloud Backup Module

After accepting the OTN agreement, it will ask you to sign in to your Oracle account. Then it will download the file. It is only 2.5MB, so the download is usually quick.

Unzip the installer file (`opc_installer.zip`). You will see two files in an uncompressed folder—`opc_install.jar` and `opc_readme.txt`. The README file has all the instructions to configure this installer; additionally you can run the **java -jar opc_installer.jar** command to see the usage instructions. The output looks similar to Figure 8-3.



```
D:\Abhinivesh\Book\Oracle cloud hands-on\opc_installer>java -jar opc_install.jar
Oracle Database Cloud Backup Module Install Tool, build 2016-10-07
No arguments supplied
Usage: java -jar opc_install.jar
    -host:          OPC hostname
    -opcId:         OPC Username
    -opcPass:       OPC Password
    -walletDir:     Directory to store wallet
    -configFile:   File name of config file
    -libDir:        Directory to store library
    -libPlatform:  Platform of library to download
    -proxyHost:    HTTP proxy host
    -proxyPort:    HTTP proxy port
    -proxyId:      HTTP proxy username, if needed
    -proxyPass:    HTTP proxy password, if needed
    -container:    OPC container to store backups

Oracle Database Cloud Backup Module Install Tool.

This utility installs the credentials and software
that are required to use the OPC Backup Service
library.

The following parameters can be specified:

-host:           Host name for the Oracle Public Cloud account.
                  It is recommended to use this parameter to specify
                  the host to which backups to be sent. Usage of
                  -serviceName and -identityDomain is deprecated.

-opcId:          The userid for the Oracle Public Cloud account.
                  This parameter is required.

-opcPass:        The password for the Oracle Public Cloud account.
                  This parameter is required.

Note that your Oracle Public Cloud userid and password
are never transmitted outside of the computer where
the install utility is running.

-walletDir:      The directory where the install tool will create an
                  Oracle Wallet containing your OPC userid and password
                  This parameter is required.

-configFile:    The name of the initialization parameter file that
                  will be created by the install tool. This parameter
                  file will be referenced during your RMAN jobs.
                  If this parameter is not specified then the
                  initialization parameter file will be created in a
                  system-dependent default location.

-libDir:         The directory where the install tool will download
                  the RMAN interface library. This parameter is optional.
                  If not specified, then the RMAN interface library will
                  will not be downloaded.
```

Figure 8-3. The *opc installer* usage

You can run it as shown in the following code. First set ORACLE_HOME and ORACLE_SID and then run the installer command. Remember to provide values according to your environment. The following example shows the command for Windows. For UNIX, take walletDir and libDir as follows:

```
walletDir $ORACLE_HOME/dbs/opc_wallet
libDir $ORACLE_HOME/lib
```

```
D:\Abhinivesh\Book\Oracle cloud hands-on\opc_installer>set
ORACLE_HOME=D:\app\oracle12c\product\12.1.0\dbhome_1
```

```
D:\Abhinivesh\Book\Oracle cloud hands-on\opc_installer>set ORACLE_SID=DB1
```

```
D:\Abhinivesh\Book\Oracle cloud hands-on\opc_installer>java -jar opc_install.jar
-serviceName Storage -identityDomain clouddb -opclId username@xxxxxx.com -opcPass
passwd -walletDir D:\app\oracle12c\product\12.1.0\dbhome_1\database\opc_wallet -
libDir D:\app\oracle12c\product\12.1.0\dbhome_1\bin
```

Oracle Database Cloud Backup Module Install Tool, build 2016-10-07

Oracle Database Cloud Backup Module credentials are valid.

Oracle Database Cloud Backup Module wallet created in directory D:\app\oracle12c
\product\12.1.0\dbhome_1\database\opc_wallet.

Oracle Database Cloud Backup Module initialization file D:\app\oracle12c\product
\12.1.0\dbhome_1\database\opcDB1.ora created.

Downloading Oracle Database Cloud Backup Module Software Library from file opc_w
indows64.zip.

Downloaded 20224088 bytes in 125 seconds. Transfer rate was 161792 bytes/second.

Download complete.

Once the Cloud Backup Module is successfully installed, you are good to move to the next step.

3. Use RMAN to take a backup using the SBT library present in backup module.

First you need to configure the RMAN backup channel to make the cloud the backup destination. Then you can run RMAN backups with the usual RMAN commands.

The RMAN configuration command is shown next. This example is for Windows OS.

For UNIX, you need to run a command similar to this one.

```
CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS='SBT_LIBRARY=/$ORACLE_HOME/lib/libopc.so,
SBT_PARMS=(OPC_PFILE=/$ORACLE_HOME/dbs/opct1.ora)';
```

Let's see how to do this configuration in Windows.

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMS='SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\oraopc.dll, ENV=(OPC_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\opcDB1.ora)';
```

old RMAN configuration parameters:

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS 'SBT_LIBRARY=libosbws.dll, SBT_PARMS=(OSB_WS_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbwsDB1.ora)';
```

new RMAN configuration parameters:

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS 'SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\oraopc.dll, ENV=(OPC_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\opcDB1.ora)';
```

new RMAN configuration parameters are successfully stored

```
RMAN> show all;
```

RMAN configuration parameters for database with db_unique_name DB1 are:

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
```

```
CONFIGURE BACKUP OPTIMIZATION OFF; # default
```

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
```

```
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
```

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
```

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; # default
```

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
```

```
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
```

```
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
```

```
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
```

```
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
```

```
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
```

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS 'SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\oraopc.dll, ENV=(OPC_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\opcDB1.ora)';
```

```

CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'D:\APP\ORACLE12C\PRODUCT\12.1.0\DBHOME_1\DATABASE\SNCFDB1.ORA'; # default

```

Once this step is done, you can run the RMAN commands as before. The following code shows the table space backup in the cloud:

```

RMAN> backup device type sbt tablespace users;

Starting backup at 23-NOV-16
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=251 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: Oracle Database Backup Service Library VER=3.16.8.5
channel ORA_SBT_TAPE_1: starting full datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00004 name=D:\APP\ORACLE12C\ORADATA\DB1\DB1\USERS01.D
BF
channel ORA_SBT_TAPE_1: starting piece 1 at 23-NOV-16
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_SBT_TAPE_1 channel at 11/23/2016 11
:40:17
ORA-27030: skgfwr: sbtwrite2 returned error
ORA-19511: non RMAN, but media manager or vendor specific failure, error text:
  KBHS-01602: backup piece 02rlknft_1_1 is not encrypted

## This error is due to the fact that your source tablespace is not encrypted. Cloud backup
doesn't allow backup without encryption.

SQL> SELECT tablespace_name, encrypted, status FROM dba_tablespaces where table
space_name='USERS';

TABLESPACE_NAME  ENC   STATUS
-----  -----
USERS          NO    ONLINE

```

DB Snapshots

Snapshot backups are used in Oracle Cloud for cloning the DB using the backup. You can go to the detail page of your CDB, as shown in Figure 8-4, and click on the Snapshots Available link.

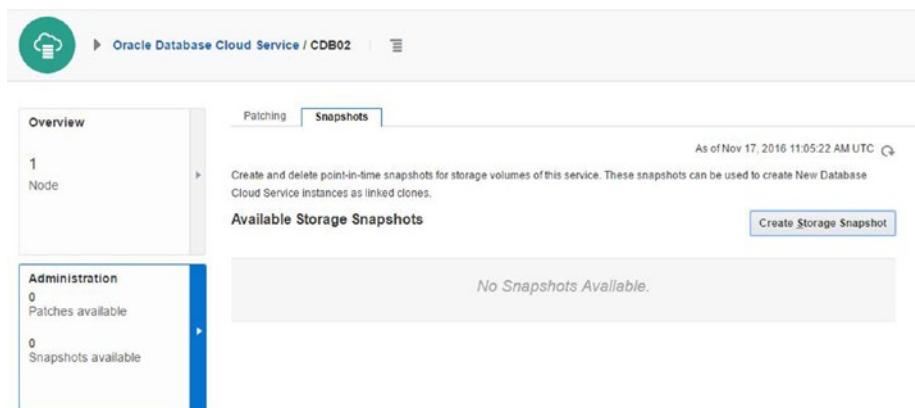


Figure 8-4. Checking for available snapshots

The Snapshots tab provides the details about the available snapshot backups for that database. If there are no snapshot backups available, you can create one by clicking Create Storage Snapshot button. It asks for the snapshot name and a description, as shown in Figure 8-5.

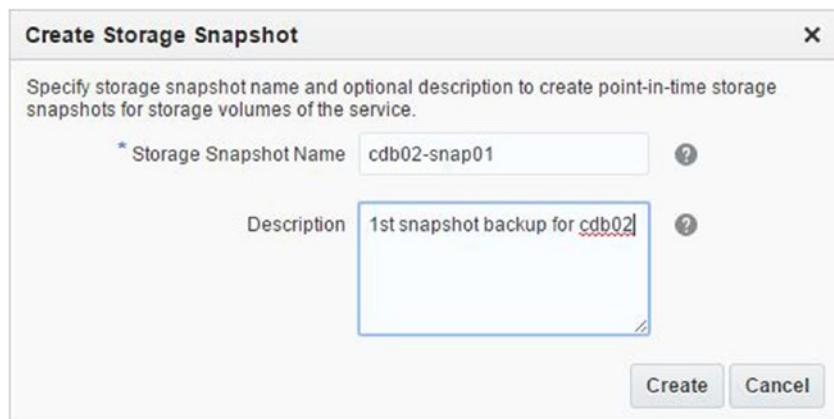


Figure 8-5. Create a snapshot

Once you provide the details, it asks for a confirmation, as shown in Figure 8-6. During the snapshot creation time, the database will be in backup mode and will not be available for normal operations.



Figure 8-6. Create snapshot: confirmation page

Upon submitting the snapshot job, the database icon will turn into Under Maintenance, as shown in Figure 8-7. The snapshot backup details will also be shown along with information if any of the clones are created using this snapshot backup.

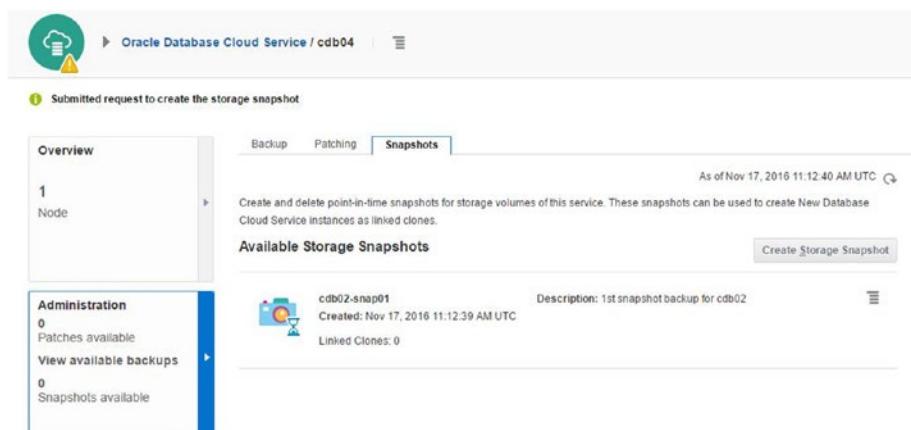


Figure 8-7. Create snapshot: in progress

Upon completion of the snapshot backup, you can use this to create clones. The linked clone details will be visible along with the snapshot details, as shown in Figure 8-8.

The screenshot shows the Oracle Database Cloud Service interface for a database named 'cdb04'. The 'Schemas' tab is selected. On the left, there's a sidebar with 'Overview' (1 Node), 'Administration' (0 Patches available, View available backups), and 'Storage' (1 Snapshot available). The main area has tabs for 'Backup', 'Patching', and 'Schemas' (which is selected). A message says 'As of Nov 18, 2016 5:10:25 AM UTC'. Below it, under 'Available Storage Snapshots', there's a list with one item: 'cdb02-snap01' (Created: Nov 17, 2016 11:12:39 AM UTC, Description: 1st snapshot backup for cdb02, Linked Clones: 0). There are 'Create Database Clone' and 'Delete' buttons next to it.

Figure 8-8. Create snapshot completion

The snapshot backup method is a very easy and simple way to back up your database; it can be used to clone existing DBs quickly. The snapshot backup consumes the storage that's equal to the size of the database.

Oracle Database Backup Service

You can set the backup for your database at the time of provisioning itself. This can be done using the storage provided under Oracle Database Backup Service. Note the REST endpoint link provided under the database backup service, as highlighted in the Figure 8-9.

The screenshot shows the 'Service Details' page for the 'Oracle Database Backup Service'. The top navigation bar has a cloud icon and the text 'Service Details: Oracle Database Backup Service'. The main content area has two sections: 'Overview' (selected) and 'Additional Information'. Under 'Overview', there are sections for 'Billing Metrics' (with a single row: 'Billing Metrics') and 'Documents' (with a single row: 'Documents'). Under 'Additional Information', there are several key details: Plan: Oracle Database Backup Service, Service Start Date: 8-Nov-2016, Service End Date: 8-Dec-2016, Subscription ID: 550120867, Service Instance ID: 550121812, Customer Account: Infosys Limited (IN), CSI Number: Not available, Data Center: US Commercial 2, Version: 16.1.0.0.0, Status: Active, and REST Endpoint: <https://storage.us2.oraclecloud.com>.

Figure 8-9. REST endpoint identification

During the database provisioning, the fourth step, called Details, gives you option to set the backup for the database being created, as shown in Figure 8-10. The backup destination can be a local VM-based backup, cloud storage backup, or none.



Figure 8-10. Backup and recovery configuration

If no backup is set during database creation, the only option that can be used for restoration is creating a snapshot-based backup, as discussed in the earlier section. For a backup to be set using cloud storage in a new container, provide the “Cloud Storage Container” path as follows:

```
https://<DB backup service REST Endpoint link> / v1/Storage-<domain name>/<new container name>
```

Replace the placeholders with the correct values as per your own settings. There can be other string formats as well and a relevant hint is provided while filling up the container details.

Storage access should be authenticated with the correct username and password. Also do not forget to check the Create Cloud Storage Container checkbox if there is no existing container available or you want to create a new container. The Confirmation page provides the selection details before actually creating the database along with backup settings, as shown in Figure 8-11.

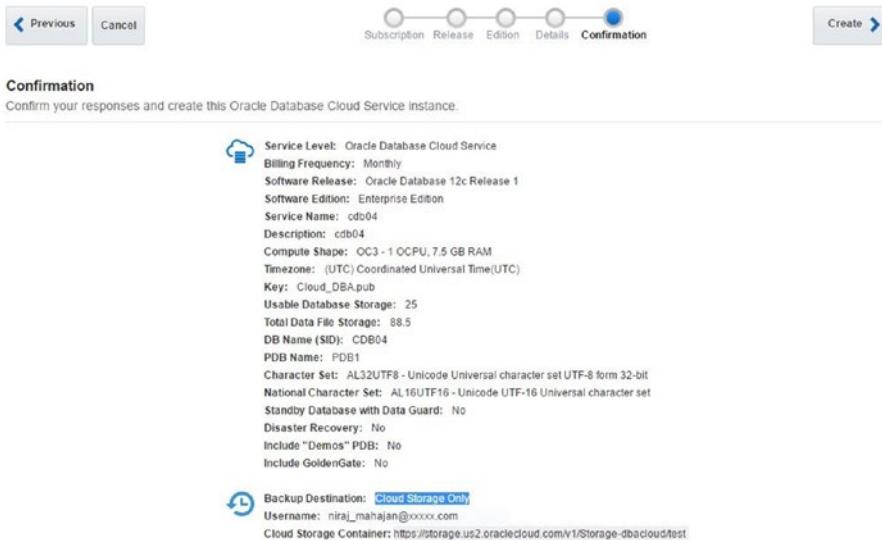


Figure 8-11. Backup and recovery configuration: confirmation page

Now, with the backup settings completed and the database creation over, it's time to take the backup and ensure that the database is recoverable.

To take a backup, click on View Available Backups, as shown in Figure 8-12. As you can see, it shows all the available backups for that database and will be blank if no backup is available.

Such on-demand backups can be taken by the `bkup_api` utility for standalone or DataGuard databases. For RAC databases, you can use the `raccli` utility.

Retention of backups, tagging backups, deleting backups, and other customization of current backups can be done using the `bkup_api` and `raccli` utilities.

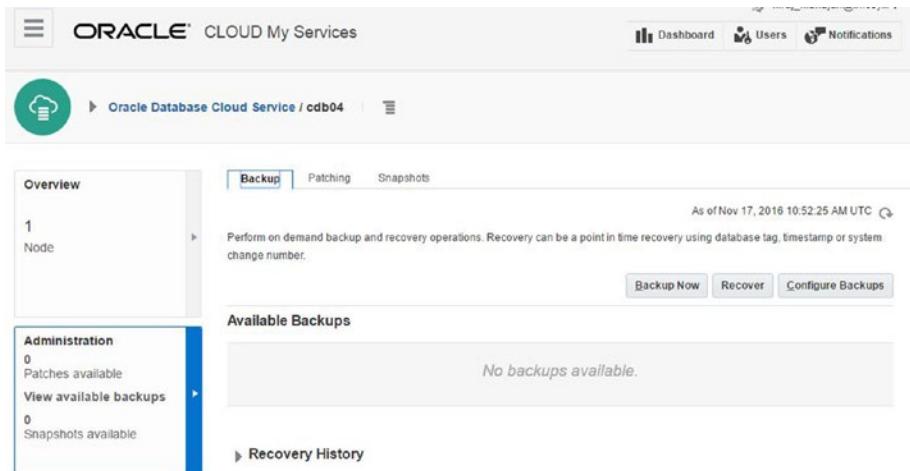


Figure 8-12. Available backups

To create a backup, click on the Backup Now button and then confirm, as shown in Figure 8-13.



Figure 8-13. On-demand backup

Post confirmation, the backup job is triggered and the details will be available upon completion, as shown in Figure 8-14.

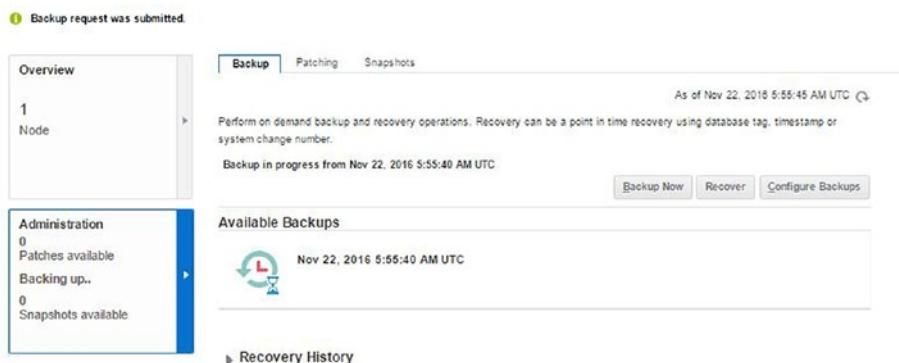


Figure 8-14. On-demand backup: in progress

If there is a change in the password for the storage access user credentials provided in the earlier section, a new password can be provided using the Configure Backups button, as shown in Figure 8-15. Note that the replication policy must be set under Oracle Storage Cloud Service before you can use the storage service for backup. You need to select the replication location to set a replication policy.

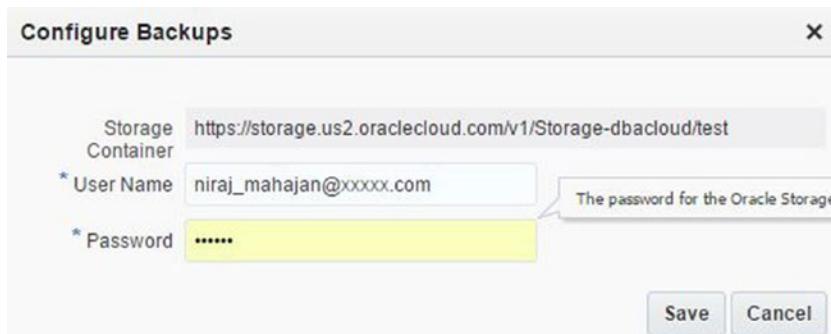


Figure 8-15. Change password for the backup user

Once the backup is complete, it is listed along with the tag and timestamp details, as shown in Figure 8-16.

Figure 8-16. Available backups list

This backup can now be used for database recovery. Databases can usually be restored up to very recently. Let's see how recovery works in Oracle Cloud database.

Database Recovery in Oracle Cloud

Database recovery in Oracle Cloud is a very simple operation and it is entirely automated. It is explained with a test case, where one table is created and one record is inserted into it. Afterward, two point-in-time recovery scenarios are explained to show the recovery of only a table and recovery of a table along with the table's data.

As the first step, connect to the database server using PuTTY and create a table to test recovery.

```
login as: opc
Authenticating with public key "rsa-key-20160621"

[opc@cdb04 ~]$ sudo -s

[root@cdb04 opc]# su - oracle

[oracle@cdb04 ~]$ sqlplus / as sysdba

SQL> ALTER SESSION SET CONTAINER = pdb1;
Session altered.

SQL> create user niraj identified by abcd1234 default tablespace users;
User created.

SQL> grant connect, dba to niraj;
Grant succeeded.

SQL> create table niraj.recovery_test (sr_no number, created_date default sysdate);
Table created.

SQL> insert into niraj.recovery_test (sr_no) values (1);
1 row created.

SQL> commit;
Commit complete.

SQL> select sr_no, to_CHAR(created,'YYYY-MM-DD-HH24MISS') create_date from
niraj.recovery_test;

SR_NO CREATE_DATE
----- -----
1    2016-11-22-060324
```

After waiting for five minutes or so, create a recovery scenario of an accidental Drop Table by dropping the newly created table. This wait time is introduced to ensure that the latest backups include your table data.

```
SQL> drop table niraj.recovery_test;  
Table dropped.  
  
SQL> select * from niraj.recovery_test;  
select * from niraj.recovery_test  
*  
ERROR at line 1:  
ORA-00942: table or view does not exist
```

In the next step, let's recover the table using the backup that was taken before the user and table were created. It will use the required archive log files automatically.

To start the recovery, click the Recover button, as shown in Figure 8-17.

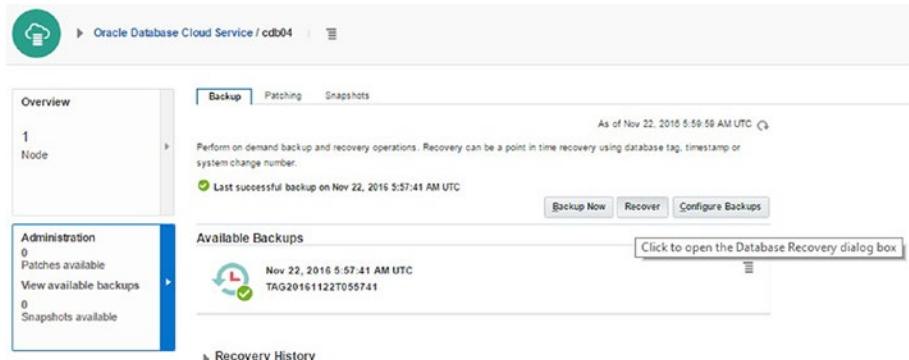


Figure 8-17. Initiate a recovery

Select a recovery date that is after the table creation but before the record insertions, as shown in Figure 8-18.

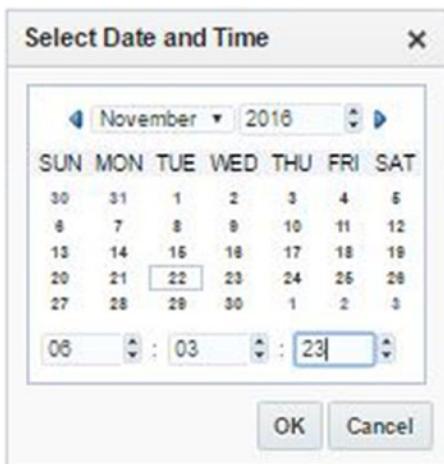


Figure 8-18. Select an appropriate recovery date

You can perform the recovery till the latest time or for a given SCN. There is also an option to perform a recovery based on a defined timestamp, as shown in Figure 8-19.

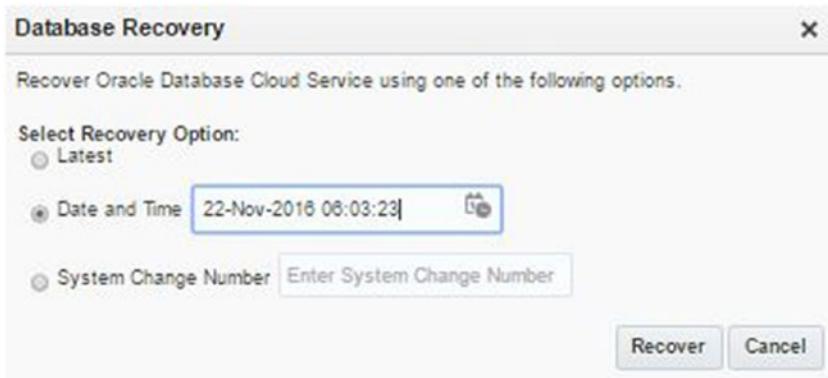


Figure 8-19. Select recovery time

Recovery will start. This will put the DB in maintenance mode and all connections will be lost, as shown in Figure 8-20.

The screenshot shows the Oracle Database Cloud Service interface for a database named 'cdb04'. The 'Backup' tab is selected in the top navigation bar. In the main area, there is an 'Overview' section showing 1 Node. Below it is an 'Administration' section indicating 0 Patches available, 0 Restoring, and 0 Snapshots available. The central panel displays a message: 'Perform on demand backup and recovery operations. Recovery can be a point in time recovery using database tag, timestamp or system change number.' It also shows a status message: 'Recovery in progress from Nov 22, 2016 6:12:41 AM UTC'. At the bottom of this panel are three buttons: 'Backup Now', 'Recover', and 'Configure Backups'. To the right, a timestamp 'As of Nov 22, 2016 6:12:45 AM UTC' is shown with a refresh icon. Below this is a 'Available Backups' section listing a backup from 'Nov 22, 2016 5:57:41 AM UTC' with tag 'TAG20161122T055741'. Further down is a 'Recovery History' section showing a recovery entry for 'Nov 22, 2016 6:12:41 AM UTC' with 'Recovery Timestamp:22-Nov-2016 06:03:23' and a status of 'Recovering...'.

Figure 8-20. Recovery in progress

Recovery is now completed, as per the given timestamp. See Figure 8-21 for more details.

This screenshot is identical to Figure 8-20, showing the Oracle Database Cloud Service interface for database 'cdb04'. The 'Backup' tab is selected. The 'Overview' section shows 1 Node. The 'Administration' section shows 0 Patches available, 0 Restoring, and 0 Snapshots available. The central panel displays a message about performing on-demand backups and recovery, and shows a recovery in progress from 'Nov 22, 2016 6:12:41 AM UTC'. It includes 'Backup Now', 'Recover', and 'Configure Backups' buttons, along with a timestamp 'As of Nov 22, 2016 6:18:00 AM UTC'. The 'Available Backups' and 'Recovery History' sections are also present, showing the same information as Figure 8-20, including the successful completion of the recovery process at 'Nov 22, 2016 6:16:52 AM UTC'.

Figure 8-21. Recovery completion

Now let's see if the table is restored.

```
SQL> ALTER SESSION SET CONTAINER = pdb1;  
Session altered.  
  
SQL> select sr_no, to_CHAR(created,'YYYY-MM-DD-HH24MISS') create_date from niraj.recovery_test;  
no rows selected
```

As expected, this process has restored the table but not the data.

Now let's give the recovery time to bring back the data as well, as shown in Figure 8-22.

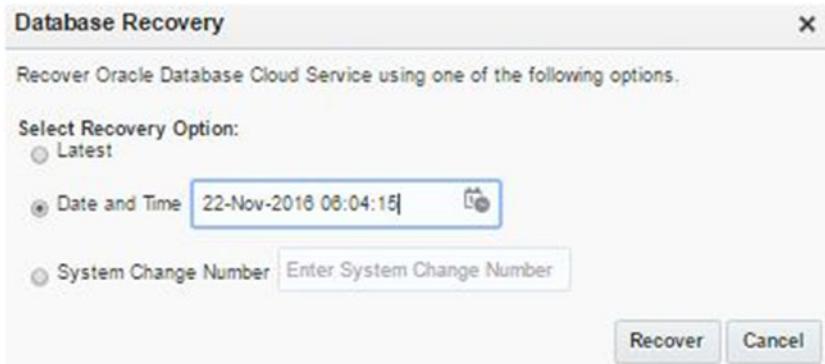


Figure 8-22. Recovery scenario 2: date and time

With the new recovery time provided, the restoration and recovery is in progress again, using the same backup, as shown in Figure 8-23.

The screenshot shows the Oracle Database Cloud Service interface for a database named 'cdb04'. The 'Backup' tab is selected. In the top right, it says 'As of Nov 22, 2016 6:26:05 AM UTC'. Below that, a message states 'Recovery in progress from Nov 22, 2016 6:26:00 AM UTC'. On the left, the 'Administration' section shows '1 Node', '0 Patches available', 'Restoring...', and '0 Snapshots available'. The 'Available Backups' section lists a single backup entry: 'Nov 22, 2016 5:57:41 AM UTC' with tag 'TAG20161122T055741'. The 'Recovery History' section shows two entries: 'Nov 22, 2016 6:26:00 AM UTC' (status 'Recovering...') and 'Nov 22, 2016 6:12:41 AM UTC' (status 'Recovering').

Figure 8-23. Recovery scenario 2: in progress

It takes around 10-15 minutes to complete the recovery. The recovery time depends on the DB size and transaction volume. Figure 8-24 confirms the completion of the recovery, which should bring back the data as well.

The screenshot shows the Oracle Database Cloud Service interface for a database named 'cdb04'. The 'Backup' tab is selected. In the top right, it says 'As of Nov 22, 2016 6:41:51 AM UTC'. Below that, a message says 'Last recovery completed successfully on Nov 22, 2016 6:31:19 AM UTC'. On the left, there's a sidebar with 'Administration' sections for patches and snapshots. The main area shows 'Available Backups' with one entry: 'Nov 22, 2016 5:57:41 AM UTC TAG20161122T055741'. Below that is a 'Recovery History' section with two entries, both with green checkmarks indicating success: 'Nov 22, 2016 6:26:00 AM UTC Recovery Timestamp:22-Nov-2016 06:04:15' and 'Nov 22, 2016 6:12:41 AM UTC Recovery Timestamp:22-Nov-2016 06:03:23'.

Figure 8-24. Recovery scenario 2: completion

Once the recovery is complete, let's verify the availability of the table and its data.

```
SQL> ALTER SESSION SET CONTAINER = pdb1;
Session altered.

SQL> select sr_no, to_CHAR(created,'YYYY-MM-DD-HH24MISS') create_date from niraj.recovery_test;
SR_NO    CREATE_DATE
-----  -----
1        2016-11-22-060324
```

This confirms that the data along with table can be restored to a given point in time.

DB Backup and Restore in AWS

The following options are present in AWS:

- On-premise backup to Amazon Cloud using Amazon S3
- DB backup and restore in Amazon RDS
- DB backup and restore in Amazon EC2

On-Premise Backup to Amazon Cloud Using Amazon S3

For taking your on-premise database backup to Amazon Cloud, you need to follow these steps:

1. Create a DB user with S3 access.
2. Download and configure Oracle Secure Backup Cloud Module for AWS.
3. Configure a RMAN backup channel.

Create a DB User with S3 Access

It is good practice to have a dedicated user for Oracle database backup. This is four-step process.

In the first step, you provide the username and select the access type, as shown in Figure 8-25.

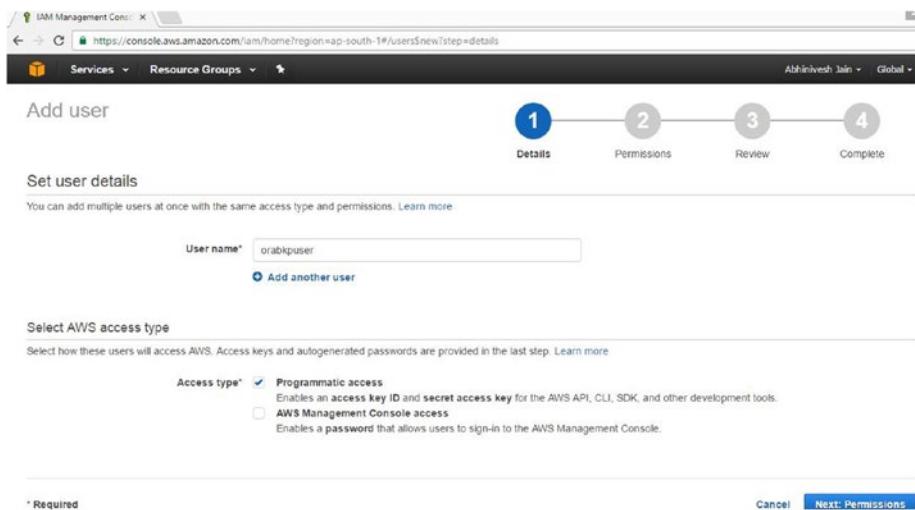


Figure 8-25. Backup user creation, step 1

In the second step, you assign 'AmazonS3FullAccess' permission to the backup user. This is shown in Figure 8-26.

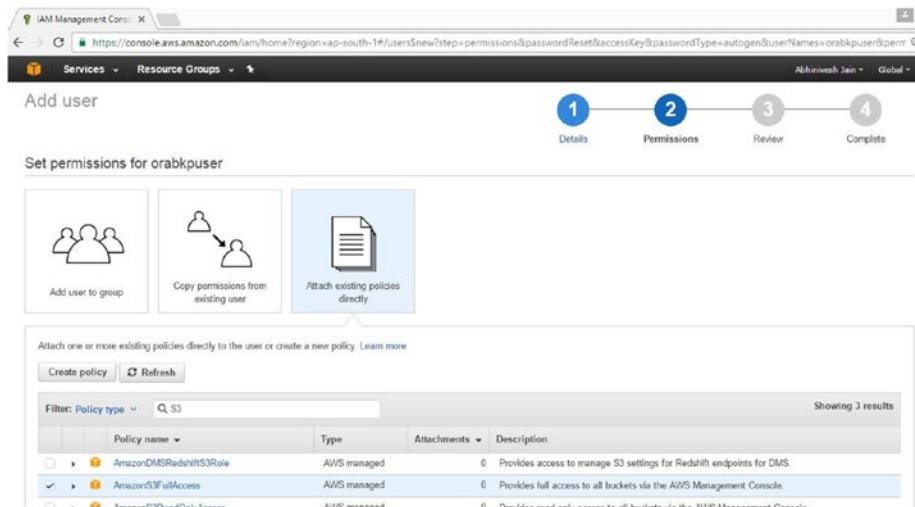


Figure 8-26. Backup user creation, step 2

In the third step, you review the details as shown in Figure 8-27.

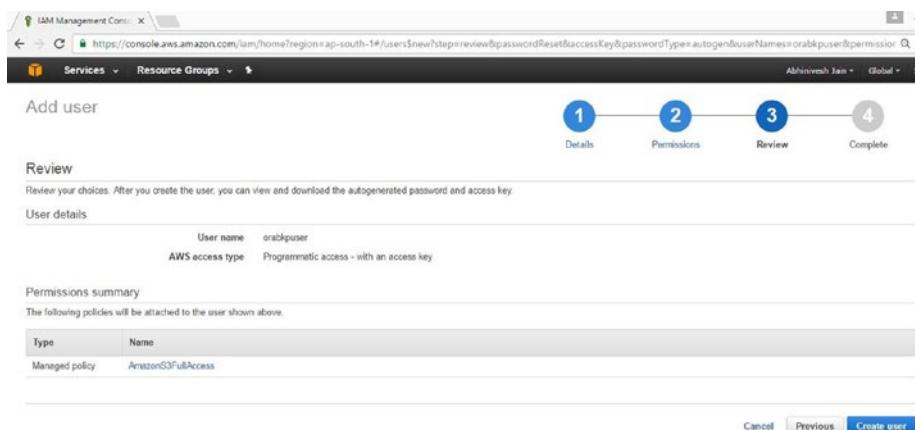


Figure 8-27. Backup user creation, step 3

In the fourth step, you click on Create User, as shown in Figure 8-28.

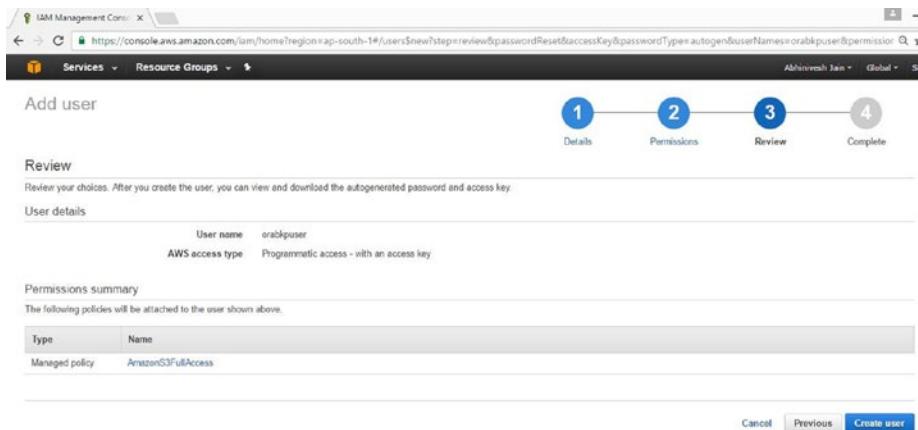


Figure 8-28. Backup user creation, step 4

After this step, you will get the successful completion message shown in Figure 8-29.

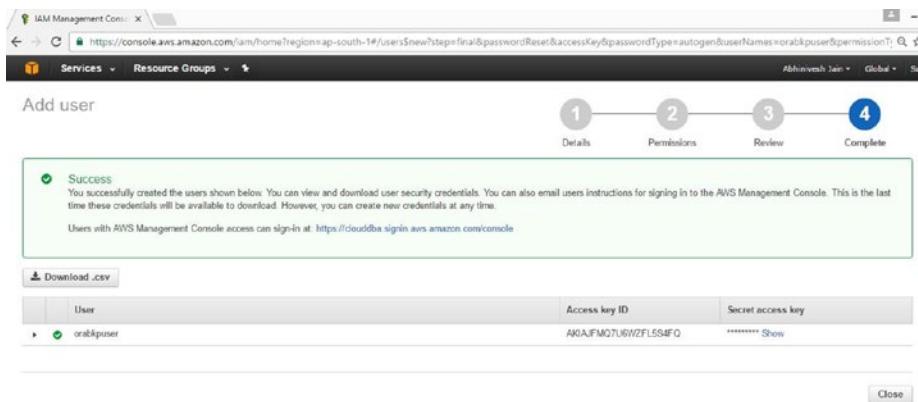


Figure 8-29. Backup user creation: completion

Download and Configure Oracle Secure Backup Cloud Module for AWS.

You can download the required module from this OTN link, as shown in Figure 8-30.

<http://www.oracle.com/technetwork/products/secure-backup/secure-backup-s3-484709.html>

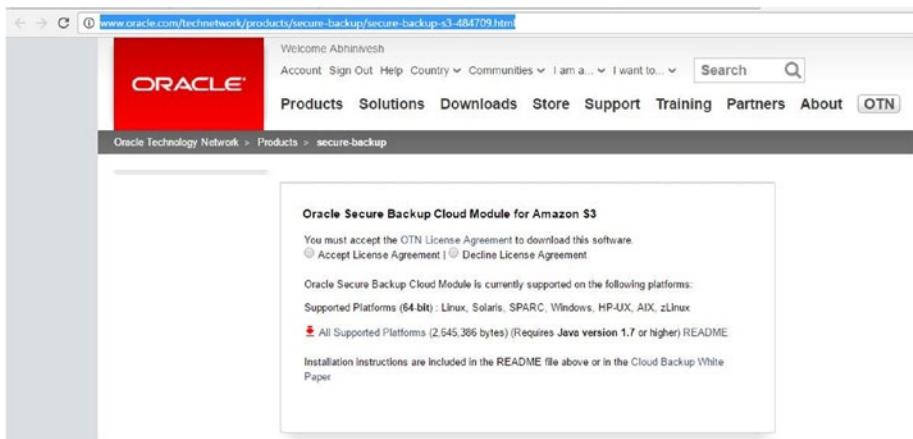


Figure 8-30. Oracle Secure Backup Cloud Module for Amazon S3 download link

Accept the OTN license agreement. After that, you will be prompted to enter the OTN credentials. The file is downloaded very quickly, as it is only 2.5MB.

Download and unzip the file before starting the configuration. Change these values as per your environment. For UNIX, the location of the wallet directory and Lib directory is shown here:

```
walletDir $ORACLE_HOME/dbs/osbws_wallet
libDir $ORACLE_HOME/lib
```

NOTE: Enter entire command in one single line. It is shown in different lines for readability purpose.

```
D:\Abhinivesh\Book\AWS hands on\osbws_installer>java -jar osbws_install.jar
-AWSID AKxxxxxxxxxxxxxxFQ
-AWSKey 06xxxxxxxxxx/eb
-otnUser Abhinivesh.jain@xxxxxx.com
-walletDir D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbws_wallet
-libDir D:\app\oracle12c\product\12.1.0\dbhome_1\bin
```

Oracle Secure Backup Web Service Install Tool, build 2016-10-07
AWS credentials are valid.

Oracle Secure Backup Web Service wallet created in directory D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbws_wallet.

Oracle Secure Backup Web Service initialization file D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbwsDB1.ora created.

Downloading Oracle Secure Backup Web Service Software Library from file osbws_windows64.zip.

Downloaded 20215967 bytes in 62 seconds. Transfer rate was 326063 bytes/second.
Download complete.

Configure a RMAN Backup Channel

In this step, you configure a RMAN backup channel to specify the cloud destination.

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt PARMs='SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\libosbws.dll, ENV=(OSB_WS_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbwsDB1.ora)';
```

old RMAN configuration parameters:

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMs 'SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\oraopc.dll, ENV=(OPC_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\opcDB1.ora)';
```

new RMAN configuration parameters:

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMs 'SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\libosbws.dll, ENV=(OSB_WS_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbwsDB1.ora)';
```

new RMAN configuration parameters are successfully stored

```
RMAN> show all;
```

RMAN configuration parameters for database with db_unique_name DB1 are:

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F';
# default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
```

```

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS 'SBT_LIBRARY=D:\app\oracle12c\product\12.1.0\dbhome_1\bin\libosbws.dll, ENV=(OSB_WS_PFILE=D:\app\oracle12c\product\12.1.0\dbhome_1\database\osbwsDB1.ora)';
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'D:\APP\ORACLE12C\PRODUCT\12.1.0\DBHOME_1
\DATABASE\SNCFDB1.ORA'; # default

```

Now you are ready to run backups using RMAN commands.

DB Backup and Restore in Amazon RDS

In RDS, database instances can be recovered as new instances only. The new instance can then be restored to point in time, as shown in Figure 8-31.

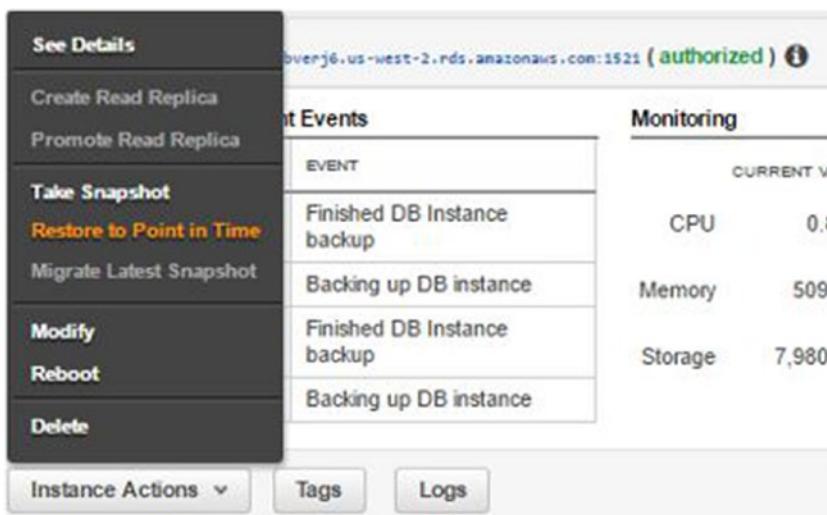


Figure 8-31. Amazon RDS recovery option

You can restore to the last restorable time or to a custom restore time. This is shown in Figure 8-32.

Launch DB Instance

You are creating a new DB Instance from a source DB Instance at a specified time. This new DB Instance will have the default DB Security Group and DB Parameter Groups.

Use Latest Restorable Time November 22, 2016 at 2:41:52 PM UTC+5:30

Use Custom Restore Time : : UTC+5:30

Figure 8-32. Amazon RDS recovery: restore time option

You also have the option of taking a manual snapshot backup at any time. This is done by clicking Take Snapshot under Instance Actions, as shown in Figure 8-33.

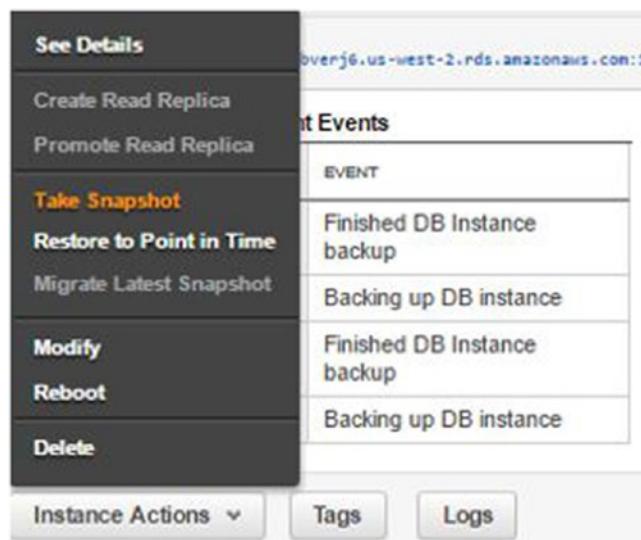


Figure 8-33. Amazon RDS: Take Snapshot option

Provide the name for your custom snapshot and click on Take Snapshot, as shown in Figure 8-34.

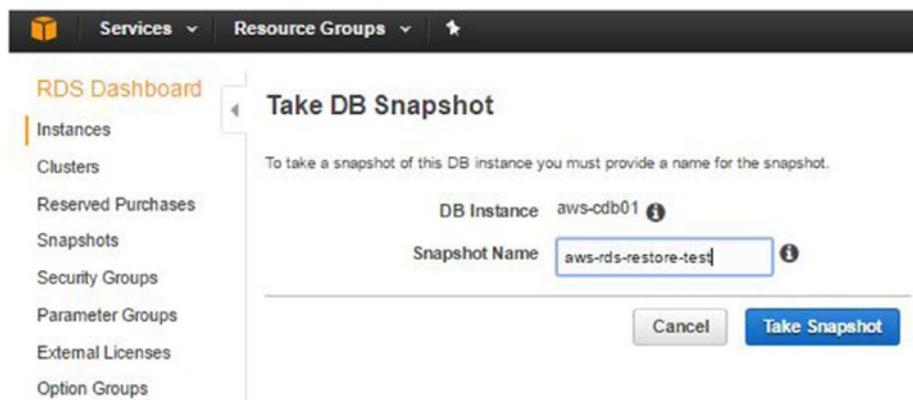


Figure 8-34. Amazon RDS: Take Snapshot

For the instances provisioned under RDS, the backup snapshot is automatically triggered once a day, as shown in Figure 8-35. It also shows the custom/manual snapshot details. First snapshot backup is created just before the instance is made available for use, post provisioning.

Viewing 6 of 6 Owned by Me									
	Snapshot	DB Instance or Cluster	Snapshot Creation Time	Status	Progress	VPC	Type	Engine	Storage Type
<input checked="" type="checkbox"/>	aws-rds-restore-test	aws-cdb01	Nov 22, 2016, 5:43:21 AM	creating	0%	vpc-04553d9f	manual	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self
<input type="checkbox"/>	rds.amazonaws.com-2016-11-22-00-13	aws-cdb01	Nov 22, 2016, 5:43:26 AM	available	Completed	vpc-04553d9f	automated	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self
<input type="checkbox"/>	rds.amazonaws.com-2016-11-21-00-13	aws-cdb01	Nov 21, 2016, 5:42:26 AM	available	Completed	vpc-04553d9f	automated	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self
<input type="checkbox"/>	rds.amazonaws.com-2016-11-20-00-34	aws-cdb01	Nov 20, 2016, 5:42:55 AM	available	Completed	vpc-04553d9f	automated	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self
<input type="checkbox"/>	rds.amazonaws.com-2016-11-19-00-13	aws-cdb01	Nov 19, 2016, 5:43:33 AM	available	Completed	vpc-04553d9f	automated	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self
<input type="checkbox"/>	rds.amazonaws.com-2016-11-18-06-23	aws-cdb01	Nov 18, 2016, 11:54:55 AM	available	Completed	vpc-04553d9f	automated	Oracle SE Two	General Purpose (SSD) 10 GB us-west-2a Self

Figure 8-35. Amazon RDS: snapshot details

CHAPTER 8 ■ BACKUP AND RESTORE

The available snapshot can be used to restore the snapshot as a new database instance using the Restore Snapshot button, as shown in Figure 8-36.

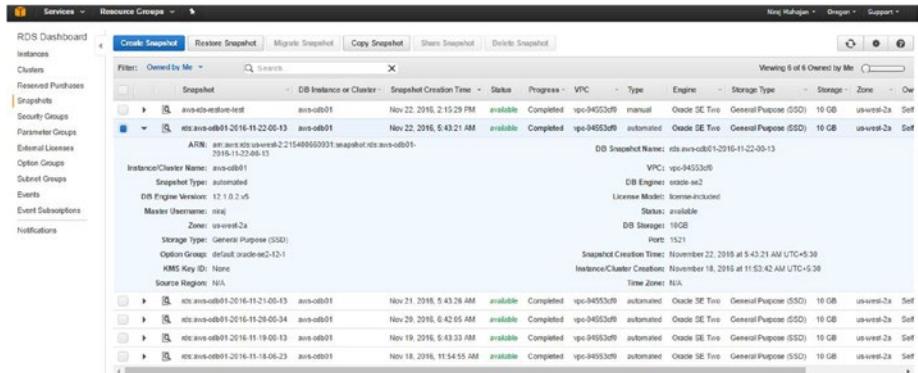


Figure 8-36. Amazon RDS: Restore Snapshot button

This restore creates a new database instance, which can be modified for some properties as per the requirements, as shown in Figure 8-37.

Restore DB Instance

You are creating a new DB Instance from a source DB Instance at a specified time. This new DB Instance will have the default DB Security Group and DB Parameter Groups.

Instance Specifications

DB Engine	oracle-se2
License Model	license-included
DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	No
Storage Type	General Purpose (SSD)

! Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

Figure 8-37. Amazon RDS: restore DB instance

You can provide the new database instance name and tag it to the snapshot from which it is being created, which can be used for future references, as shown in Figure 8-38.



Figure 8-38. Amazon RDS: restore DB instance customize settings

Upon submitting the Restore DB Instance request, the new DB creation is triggered. It will include the data as of the snapshot time. Figure 8-39 shows the restore in progress.



Figure 8-39. Amazon RDS: restore DB instance in progress

Once the restoration is complete, the database can be used, as shown in Figure 8-40.

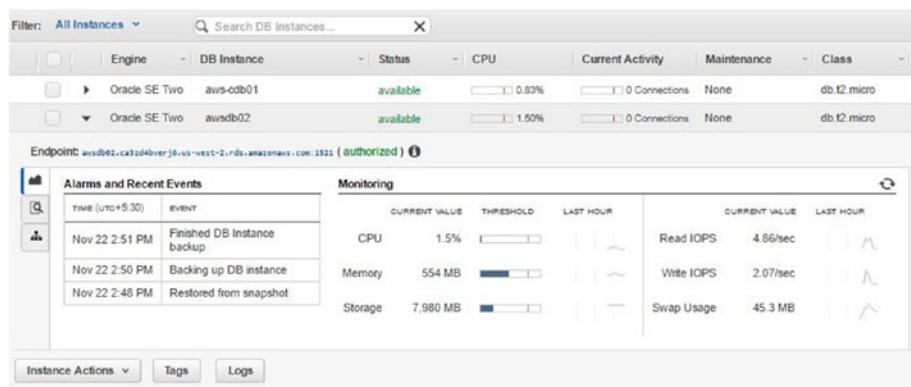


Figure 8-40. Amazon RDS: restore DB instance completion

Using snapshot backup, you can recover and clone your database quickly and easily, along with the choice to select the restore timestamp.

DB Backup and Restore in Amazon EC2

Backing up and restoring Oracle database hosted on an EC2 instance is easy because Oracle Secure Backup Cloud Module comes preconfigured with the EC2 instance.

Backup and Restore Best Practices

The following list shows some of the backup and restore best practices:

- Use compression wherever possible, as it will reduce the network transfer time.
- Use encryption before transferring the backup to the cloud.
- RMAN backup compression and encryption features can be utilized with Oracle Secure Backup Module without any extra license; however, it is recommended you check with Oracle representative for more details.
- Use more than one RMAN channel for faster backup and restore. Oracle Cloud Backup Module is licensed as part of Oracle Secure Backup and it is calculated per RMAN channel, so special consideration is required in this regard.
- Oracle Secure Backup (OSB) Cloud Module is a licensed module, so ensure that you have the right license before using it. You can refer to Oracle support note OSB Cloud Module: FAQ (Doc ID 740226.1) for more details.

Summary

In this chapter, you learned about Oracle database backup and restore options in Oracle Cloud and AWS Cloud. You read an overview of database backup and restore from the cloud perspective, including taking an on-premise DB backup to the cloud, backing up and restoring within the cloud, and taking DB snapshots in Oracle Cloud. You are now also conversed with DB backup and restore in AWS using S3 and other readily available tools. You can also use Oracle Secure Backup Cloud Module for AWS for your backup.

The next chapter covers the important aspects of how to administer, manage, and monitor your database in the cloud.

CHAPTER 9



Manage and Monitor

This chapter covers database administration and monitoring options for your database in Oracle Cloud and AWS. It is important for a support DBA to be able to monitor database health quickly. Proactive and customized alerting is the very important. You will learn how various traditional tools and some new tools can be used to control your database running on AWS or Oracle Cloud.

Overview of Cloud DB Monitoring and Management

Oracle database monitoring and management is not entirely different from the on-premise database paradigm. In fact, many of the tools, such as SQL Developer, TOAD, and OEM 12c Cloud Control, allow you to connect to your cloud database and perform most of the tasks as before. However, there are some specific tools, such as DBaaS Monitor in Oracle Cloud and command-line options such as AWSCLI and DBaaSCLI, that are only applicable for the cloud. In this chapter, you will learn how to use the existing tools with a cloud database and learn how to use the new, cloud-specific tools.

Table 9-1 provides a snapshot of multiple tools available for Oracle Cloud and AWS, along with overviews of the tools.

Table 9-1. Available Tools and Purpose

Tool Name	AWS	Oracle Cloud	Tasks
DBaaS Monitor	Not applicable	Readily available	Database and OS monitoring
EM Express	Readily available	Readily available	Monitoring and administration
Command-Line Interfaces	Not applicable	Dbaascli, raccli, oracle-dbcs-cli (Linux only)	Much finer control compared to GUI
SQL Developer	Free local download	Free local download	Basic monitoring and administration
CloudWatch	Paid subscription	Not applicable	Monitoring and custom notifications
AWS Database Metrics	Readily available	Not applicable	Database and OS health monitoring

DB Monitoring and Management in Oracle Cloud

Oracle Cloud provides several options for monitoring your database hosted on the cloud. To use these monitoring tools, you need to open the specific ports for your cloud database instance. This can be done from the Access Rules settings, as shown in Figure 9-1.

The screenshot shows the Oracle Cloud My Services interface. At the top, there's a navigation bar with 'Dashboard', 'Users', and 'Notif'. Below it, a sidebar on the left lists 'Overview' (1 Node), 'Administration' (0 Patches available, View available backups, 0 Snapshots available), and 'Nodes' (1 Node). The main area has a 'Summary' card showing 1 Nodes, 5 GB memory, and 150 GB Storage. A context menu is open over the 'Nodes' card, listing 'Open DBaaS Monitor Console', 'Open Application Express Console', 'Open EM Console', 'SSH Access', 'Access Rules', and 'Replace Database using Backup'. Below the summary card, details for 'cdb04' are shown: SQL*Net Port: 1521, Public IP: 129.144.149.244, SID: CDB04, PDB Name: PDB1, OCpus: 1, Memory: 7.5 GB, Storage: 150 GB. At the bottom, an 'Additional Information' section provides details about the identity domain (dbacloud), edition (Enterprise Edition), service level (Oracle Database Cloud Service), and subscription type (Monthly).

Figure 9-1. Access rules

You can connect using SQL*PLUS, SQL Developer, TOAD, or some other locally available tool. The interface of these local tools remains the same as when working on a local database. In addition to these, you can use the DBaaS Monitor and EM Console to monitor your database in Oracle Cloud. Both of these web-based interfaces are readily available once you enable the access ports, as shown in Figure 9-2 for the DBaaS Monitor.

Access Rules							
You can use access rules to control network access to service components. On this page, you can manage your access rules.							
Results per page: 10 ▾ 9 result(s) as of Nov 23, 2016 6:48:40 AM UTC							
Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type
OK	ora_p2_ssh	PUBLIC-INTERNET	DB	22	TCP		DEFAULT
OK	ora_p2_dblistener	PUBLIC-INTERNET	DB	1521	TCP		DEFAULT
OK	ora_p2_http	PUBLIC-INTERNET	DB	80	TCP		DEFAULT
OK	ora_p2_httpsl	PUBLIC-INTERNET	DB	443	TCP		DEFAULT
OK	ora_p2_httpadmin	PUBLIC-INTERNET	DB	4848	TCP		DEFAULT
OK	ora_p2_dbconsole	PUBLIC-INTERNET	DB	1158	TCP		DEFAULT
OK	ora_p2_dbexpress	PUBLIC-INTERNET	DB	5500	TCP		DEFAULT
OK	sys_infra2db_ssh	PAAS-INFRA	DB	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM
OK	ora_trusted_hosts_dbli...	127.0.0.1/32	DB	1521	TCP	DO NOT MODIFY: A securi...	SYSTEM

Figure 9-2. Access rules: enable option

You need to confirm the change in the access rule to enable it, as shown in Figure 9-3.



Figure 9-3. Access rules: enable option confirmation

This will change the icon for that port and will enable access using the DBaaS Monitor for that database.

Access Rules								Create Rule
Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type	Actions
	ora_p2_ssh	PUBLIC-INTERNET	DB	22	TCP		DEFAULT	
	ora_p2_dblistener	PUBLIC-INTERNET	DB	1521	TCP		DEFAULT	
	ora_p2_http	PUBLIC-INTERNET	DB	80	TCP		DEFAULT	
	ora_p2_httpsl	PUBLIC-INTERNET	DB	443	TCP		DEFAULT	
	ora_p2_httpadmin	PUBLIC-INTERNET	DB	4848	TCP		DEFAULT	
	ora_p2_dbconsole	PUBLIC-INTERNET	DB	1158	TCP		DEFAULT	
	ora_p2_dbexpress	PUBLIC-INTERNET	DB	5500	TCP		DEFAULT	
	sys_infra2db_ssh	PAAS-INFRA	DB	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM	
	ora_trusted_hosts_dbli...	127.0.0.1/32	DB	1521	TCP	DO NOT MODIFY: A secur...	SYSTEM	

Figure 9-4. Access rules: enable completion

DBaaS Monitor

Now that you have enabled DBaaS Monitor, you can access it, as shown in Figure 9-5.



Figure 9-5. Accessing the DBaaS Monitor console

This takes you to the login page for DBaaS Monitor, as shown in Figure 9-6. The default username is `dbaas_monitor` and the password is the same admin password provided during database creation.

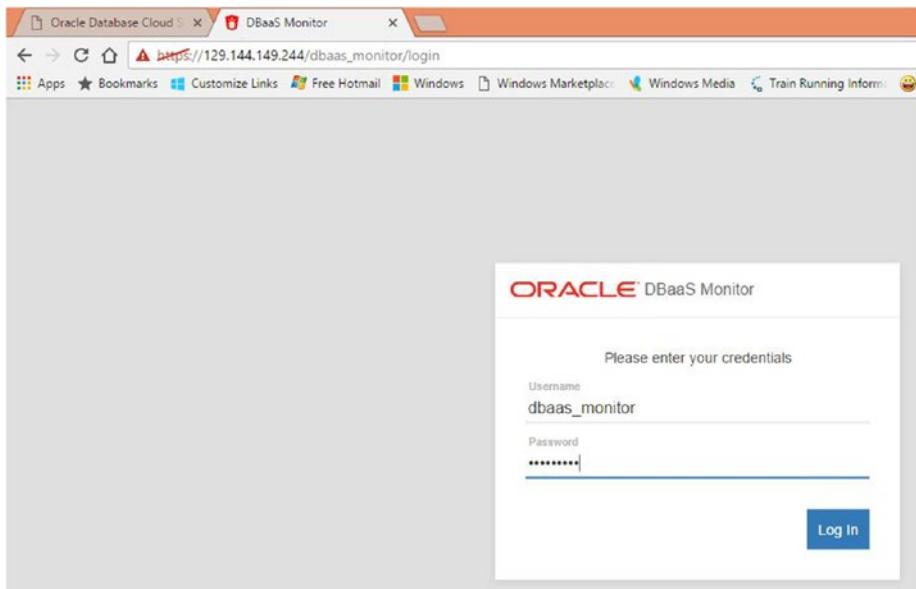


Figure 9-6. DBaaS Monitor: sign in

The home page is simple and it provides details about the database and listener status. It also provides information about the last backup, storage consumption, and sessions, as shown in Figure 9-7.

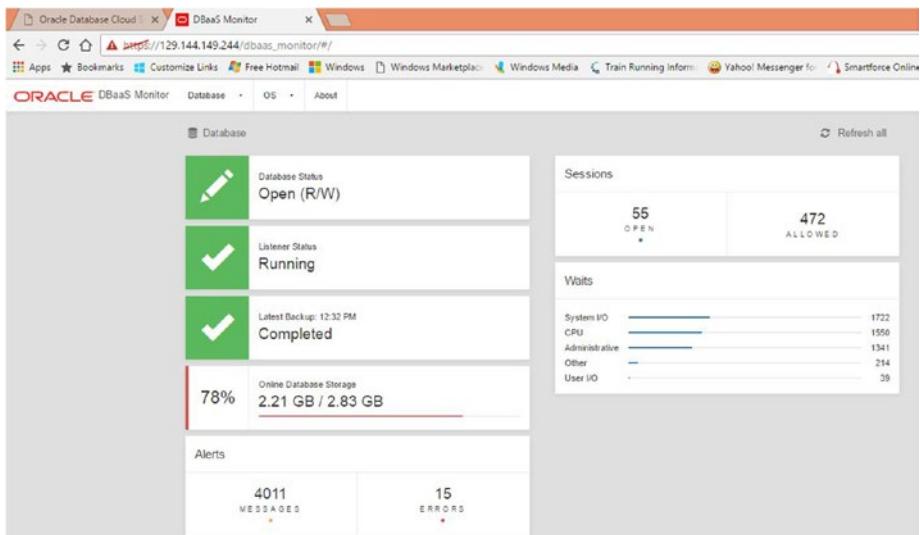


Figure 9-7. DBaaS Monitor: database info

On the lower side of the page, you can get operating system information like memory, CPU, and filesystem usage, as shown in Figure 9-8.

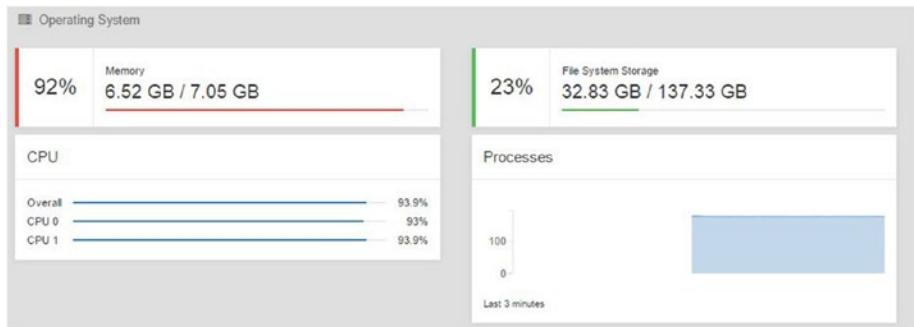


Figure 9-8. DBaaS Monitor: OS info

Using the DBaaS Monitor, you can perform all the activities listed in Table 9-2.

Table 9-2. DBaaS Options

-
- Database
 - Manage
 - Listener
 - Storage
 - Backups
 - Alerts
 - Sessions
 - Waits
 - Movements
 - Parameters
 - Real-Time SQL Monitor
 - OS
 - Memory
 - CPU
 - Storage
 - Processes
-

As part of the manage process, you can stop or start the CDB database. You can also clone, unplug, drop, or modify the state of any existing PDBs. There is also a provision to either create a new PDB or plug in a PDB, as shown in Figure 9-9.

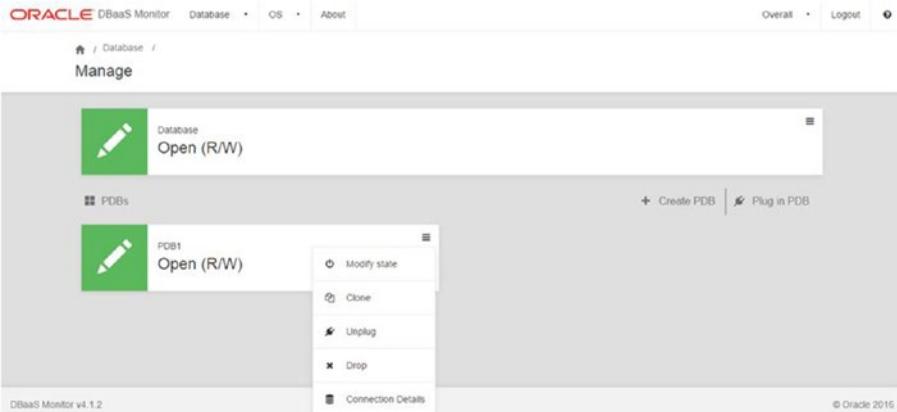


Figure 9-9. DBaaS Monitor: PDB options

Using the Listener menu, you can stop or start the listener and view details about the listener configurations, as shown in Figure 9-10.

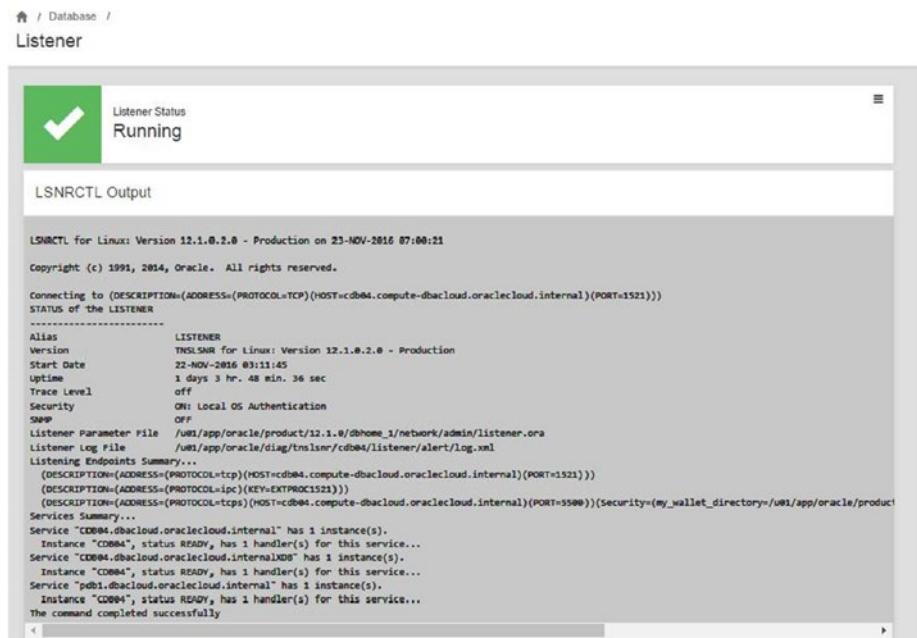


Figure 9-10. DBaaS Monitor: listener options

From the Storage menu, you can get the storage consumption details by CDB and all PDBs and can drill down to the table space level. You can further see the segment level details by getting more details about a particular table space (Figure 9-11).

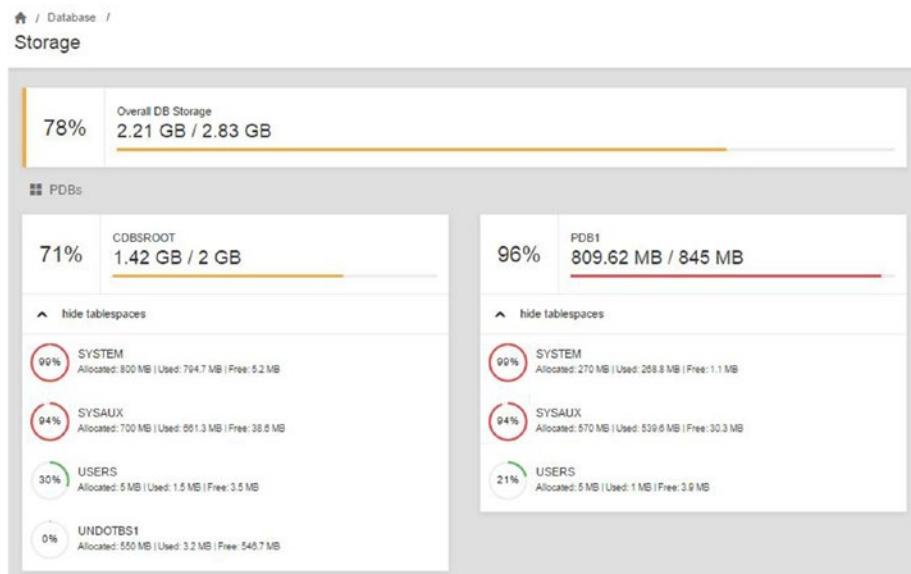


Figure 9-11. DBaaS Monitor: storage options

Backups are very important for ensuring data availability in case of a crash or accidental data loss. These details are available from the Backups menu of the DBaaS Monitor. You can get details of completed and ongoing backups along with process IDs and timings. As you can see in Figure 9-12, there is a daily autobackup running, which is offered as part of Oracle Cloud service.

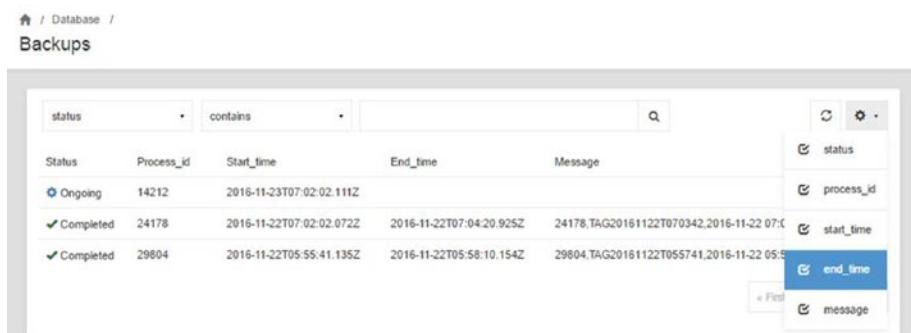


Figure 9-12. DBaaS Monitor: backup options

As a best practice, you can create an external table in your local database showing the contents of the alert log. A similar view is available for your Oracle Cloud database, to see the details on database errors under the Alerts menu, as shown in Figure 9-13.

The screenshot shows the 'Alerts' section of the DBaaS Monitor. At the top, there is a search bar with dropdown menus for 'type' (set to 'contains') and 'Message_text'. Below the search bar is a table with four columns: 'Type', 'Timestamp', 'Host', and 'Message_text'. The table contains four rows of data:

Type	Timestamp	Host	Message_text
Message	2016-11-23T08:30:09.052Z	cdb04.compute-dbacloud.oraclecloud.internal (10.16.211.14)	Archived Log entry 74 added for thread 1 sequence 54 ID 0x72b7579d dest 1:
Message	2016-11-23T08:30:08.457Z	cdb04.compute-dbacloud.oraclecloud.internal (10.16.211.14)	Current log# 1 seq# 55 mem# 0 /u04/app/oracle/redo redo01.log
Message	2016-11-23T08:30:08.456Z	cdb04.compute-dbacloud.oraclecloud.internal (10.16.211.14)	Thread 1 advanced to log sequence 55 (LGWR switch)
Message	2016-11-23T08:30:08.245Z	cdb04.compute-dbacloud.oraclecloud.internal (10.16.211.14)	ALTER SYSTEM ARCHIVE LOG

Figure 9-13. DBaaS Monitor: alerts info

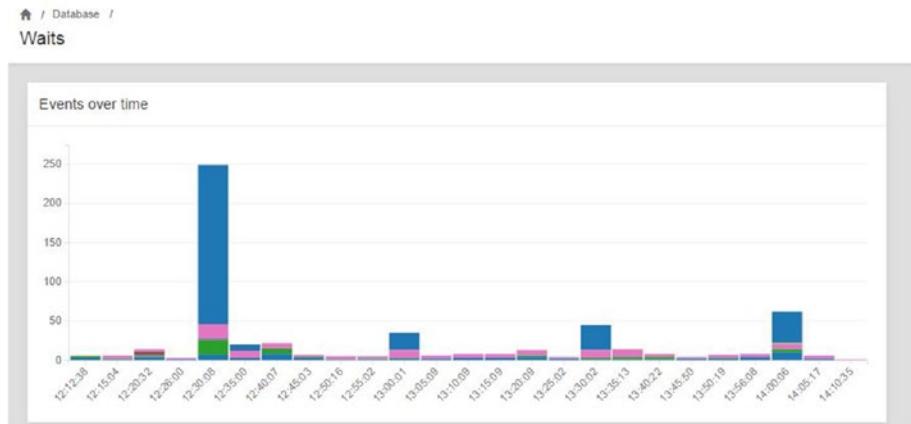
You can search for a particular session and get the details for all active and inactive sessions using the Sessions menu, as shown in Figure 9-14. Each of these pages is customizable to show only the columns of interest, by selecting the settings for that menu page.

The screenshot shows the 'Sessions' section of the DBaaS Monitor. At the top, there is a search bar with dropdown menus for 'sid' (set to 'contains') and 'Username'. Below the search bar is a table with seven columns: 'sid', 'Serial', 'Sql_id', 'Os_user', 'Username', 'Status', and 'Module'. The table contains six rows of data:

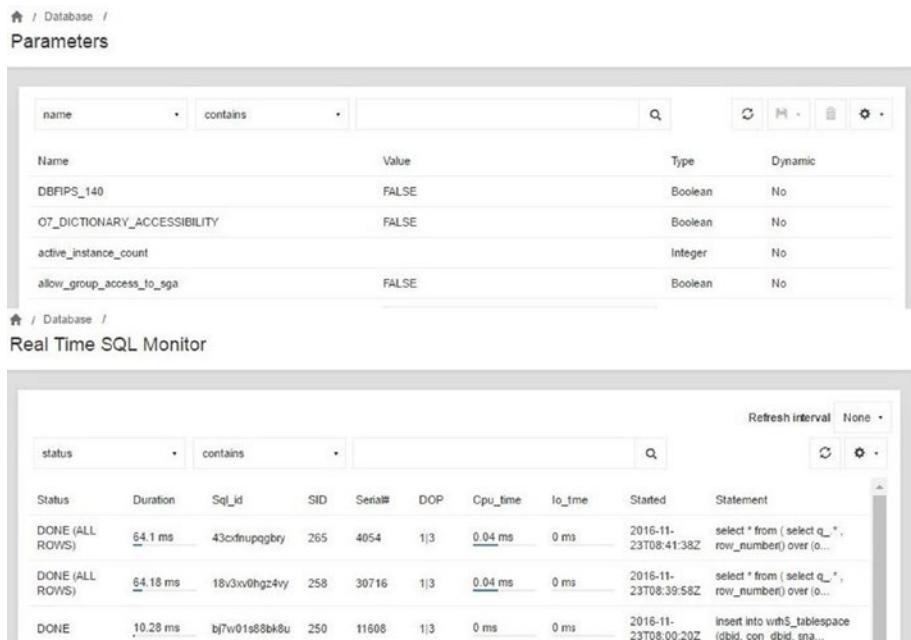
Sid	Serial	Sql_id	Os_user	Username	Status	Module
1	41098		oracle		active	
2	61519		oracle		active	
272	2381	a7wsa1pjv1ns9	oracle	C##DBAAS_MONITOR	active	APEX Listener
269	28828		oracle		active	
284	14333		oracle		active	Streams
260	10313		oracle		active	

Figure 9-14. DBaaS Monitor: session info

In case of performance issues or to monitor the health of the database, you must understand where your database is spending time. This is available on the Waits page and you can get the details on the SQLs and SIDs on the lower side of the page by clicking the specific color bar, as shown in Figure 9-15.

**Figure 9-15.** DBaaS Monitor: waits info

Whenever you take support ownership of any database, you should know the different parameters and their values. The Parameters menu contains this critical information, as shown in Figure 9-16.

**Figure 9-16.** DBaaS Monitor: initialization parameters and the real-time SQL Monitor

The Real-Time SQL Monitor is also available under the Database menu, as shown in Figure 9-16. This is a relatively new feature. It provides details on the currently running SQLs in your database and can be useful in identifying a bad SQL.

With this, you can move to the operating system level monitoring available using DBaaS Monitor.

Memory allocation plays an important role in database performance and is very useful for data caching to improve response time for SQLs. This can be viewed from the Memory menu under the OS, as shown in Figure 9-17.

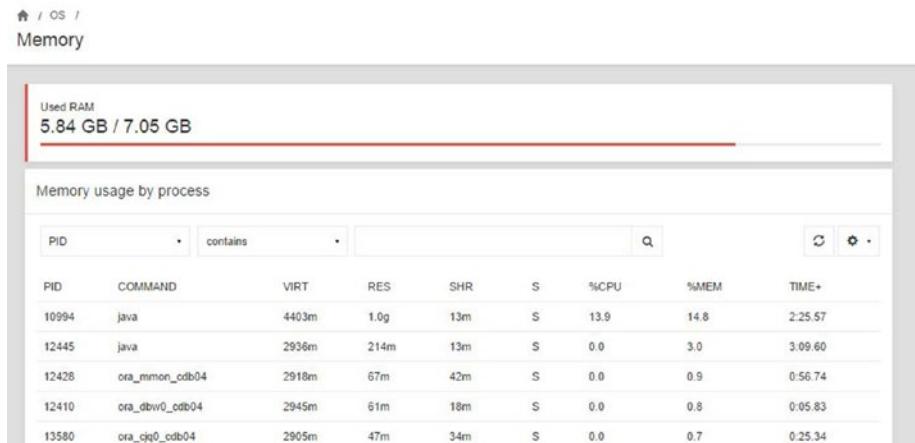


Figure 9-17. DBaaS Monitor: memory details

CPU workload details can be monitored using the OS CPU monitoring page, as shown in Figure 9-18. CPU utilization information can be useful for identifying the need of additional processing power.

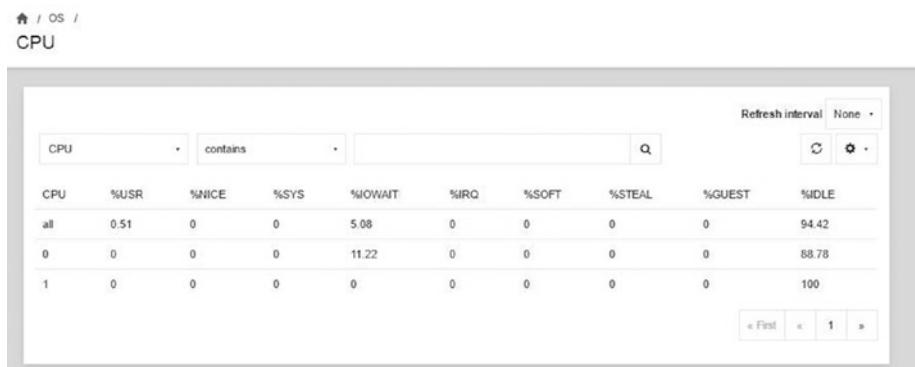


Figure 9-18. DBaaS Monitor: CPU details

You have already seen the storage utilization at the CDB and PDB level from the Database menu. Similar details at the OS level are available under OS Storage, as shown in Figure 9-19. They can be useful for planning for additional storage depending on utilization and growth. This also provides the mount point level details.

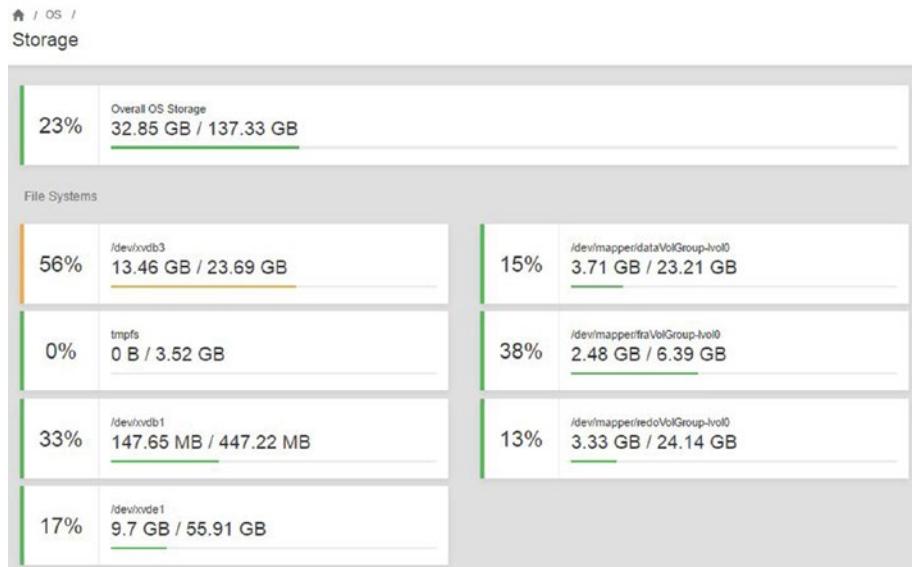


Figure 9-19. DBaaS Monitor: storage details

You need to look at OS level processes to identify the top resource-consuming processes and to look for ways to improve system performance. These details are available in the Processes menu, as shown in Figure 9-20.

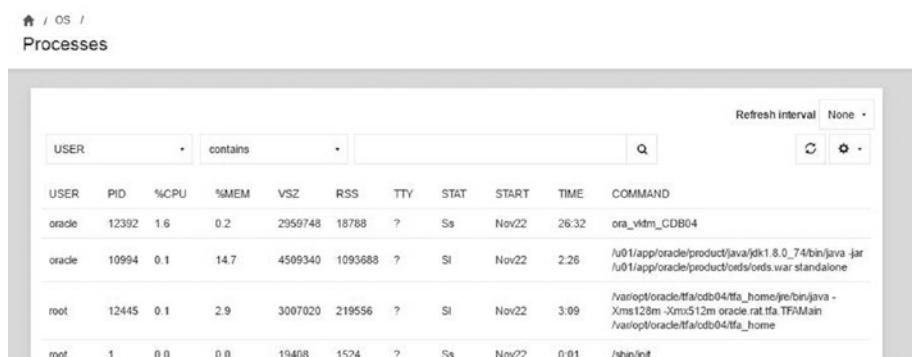


Figure 9-20. DBaaS Monitor: processes details

Most of these details under the Database and OS menus can be viewed at different levels, such as overall, via CDBroot, or at the individual PDB level, using the settings shown in Figure 9-21.

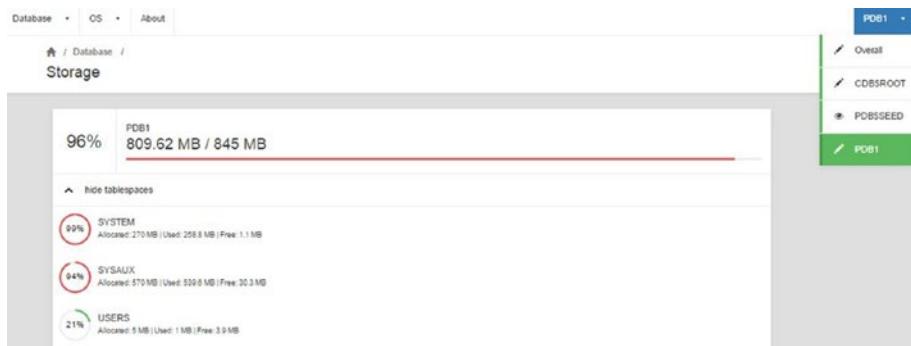


Figure 9-21. DBaaS Monitor: monitoring levels

This concludes our discussion of the DBaaS Monitor capabilities. This is an excellent tool for monitoring your database in Oracle Cloud.

EM Express

Enterprise Manager (EM) Express is another GUI-based tool that's readily available and can be accessed once you open port 5500. This tool is very similar to OEM and can be used for administration and monitoring.

You need to log in to EM Express as a sys or system user and using the password provided during DB creation, as shown in Figure 9-22.

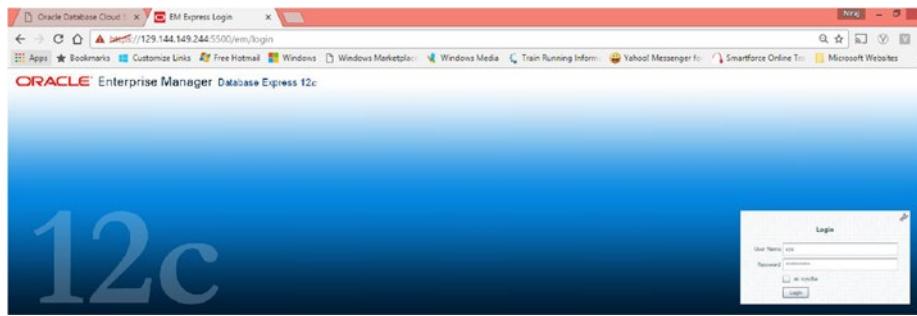


Figure 9-22. EM Express sign in

The dashboard of EM Express is shown in Figure 9-23. It provides information about the database's status, jobs, and performance details, such as waits and I/O.

CHAPTER 9 ■ MANAGE AND MONITOR

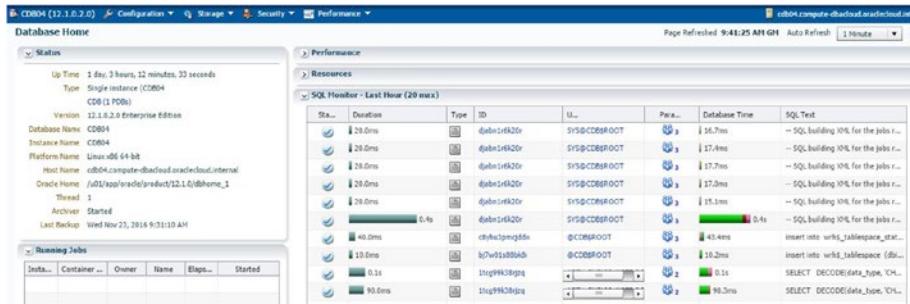


Figure 9-23. EM Express dashboard

Database performance and health can be assessed using details on waits, I/O, and CPU utilization, as shown in Figure 9-24. You can get details for active sessions and memory as well.

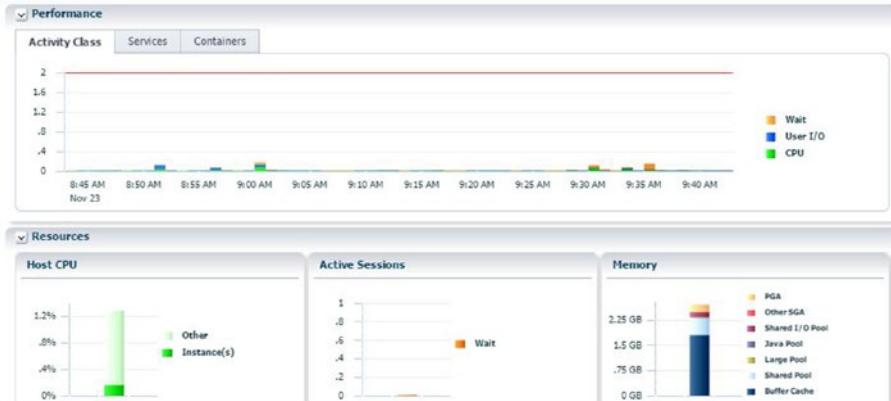


Figure 9-24. EM Express DB performance dashboard

There are many options for monitoring and administrating Oracle Cloud database using EM Express, as shown in Table 9-3.

Table 9-3. EM Express Options

- | | | |
|--|---|---|
| <ul style="list-style-type: none"> Configuration <ul style="list-style-type: none"> Initialization Parameters Memory Database Feature Usage Current Database Properties | <ul style="list-style-type: none"> Storage <ul style="list-style-type: none"> Undo Management Redo Log Groups Archive Logs Control Files | <ul style="list-style-type: none"> Security <ul style="list-style-type: none"> Users Roles |
| | | <ul style="list-style-type: none"> Performance <ul style="list-style-type: none"> Performance Hub SQL Tuning Advisor |

Some of the menu pages are discussed in depth here. EM Express is used not only for monitoring but can also be used to modify the initialization parameter settings, as shown in Figure 9-25.

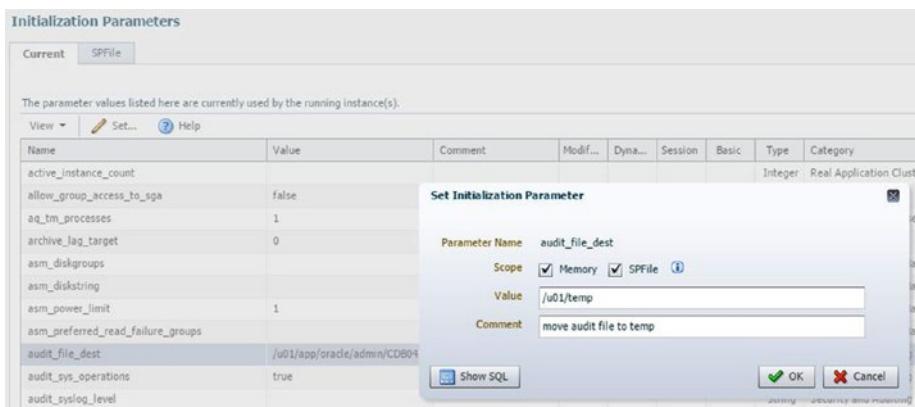


Figure 9-25. EM Express initialization parameter change

As shown in Figure 9-26, the Memory management page provides a lot of information on SGA and PGA allocation and advisors. You can also see the details of the different memory components along with the allocations. The top PGA consuming sessions can be drilled down to get more information.

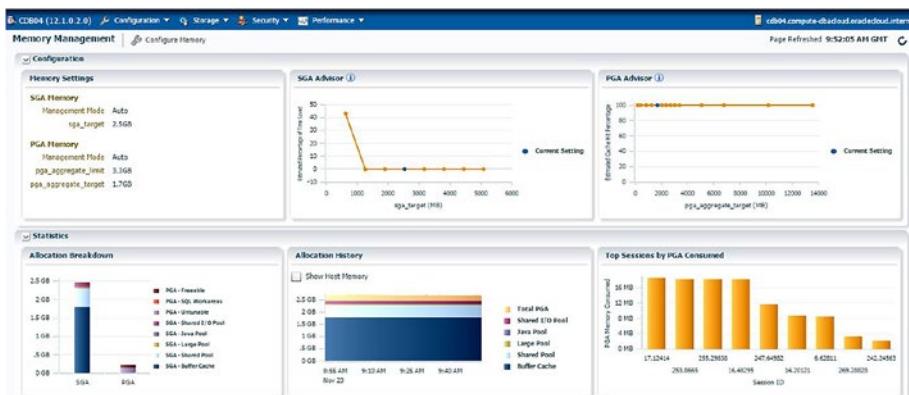
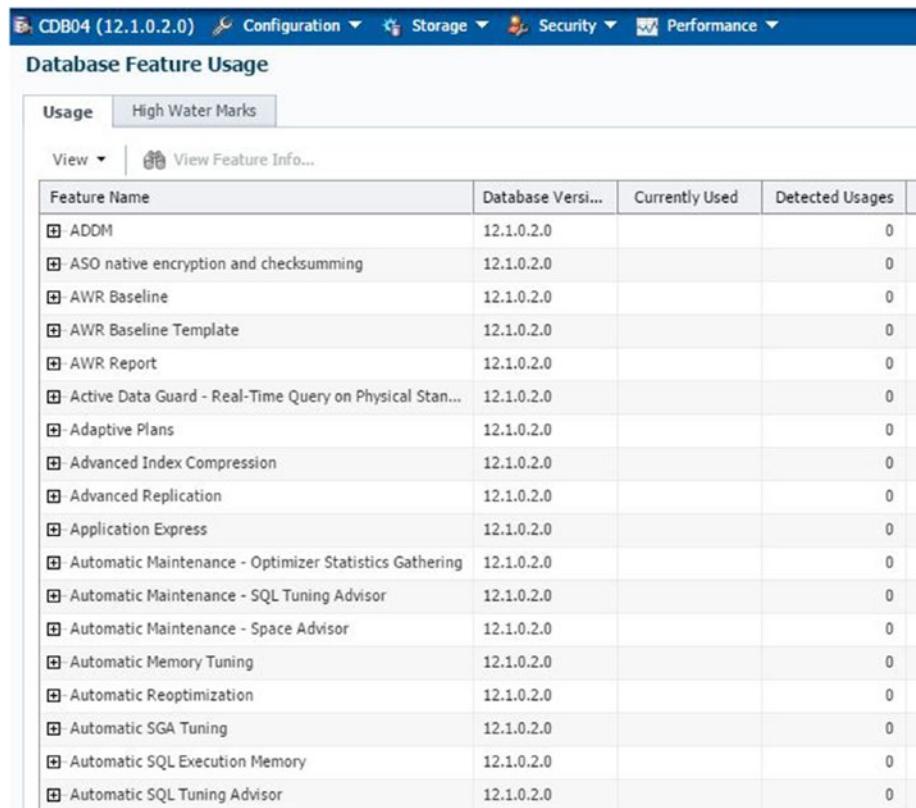


Figure 9-26. EM Express memory management

The Database Feature Usage page, as shown in Figure 9-27, is almost the same as its corresponding view in any Oracle database. It can be used to track the different features used, primarily for licensing purposes.



The screenshot shows the Database Feature Usage page in the Enterprise Manager Express interface. The top navigation bar includes tabs for Configuration, Storage, Security, and Performance, along with the system identifier CDB04 (12.1.0.2.0). Below the navigation bar, the title "Database Feature Usage" is displayed. A secondary navigation bar below the title has two tabs: "Usage" (which is selected) and "High Water Marks". Underneath these tabs is a "View" dropdown menu and a "View Feature Info..." link. The main content area is a table with the following columns: Feature Name, Database Version, Currently Used, and Detected Usages. The table lists various Oracle features, all of which are currently unused (0) and have been detected at version 12.1.0.2.0. The features listed include ADDM, ASO native encryption and checksumming, AWR Baseline, AWR Baseline Template, AWR Report, Active Data Guard - Real-Time Query on Physical Stan..., Adaptive Plans, Advanced Index Compression, Advanced Replication, Application Express, Automatic Maintenance - Optimizer Statistics Gathering, Automatic Maintenance - SQL Tuning Advisor, Automatic Maintenance - Space Advisor, Automatic Memory Tuning, Automatic Reoptimization, Automatic SGA Tuning, Automatic SQL Execution Memory, and Automatic SQL Tuning Advisor.

Feature Name	Database Versi...	Currently Used	Detected Usages
■ ADDM	12.1.0.2.0		0
■ ASO native encryption and checksumming	12.1.0.2.0		0
■ AWR Baseline	12.1.0.2.0		0
■ AWR Baseline Template	12.1.0.2.0		0
■ AWR Report	12.1.0.2.0		0
■ Active Data Guard - Real-Time Query on Physical Stan...	12.1.0.2.0		0
■ Adaptive Plans	12.1.0.2.0		0
■ Advanced Index Compression	12.1.0.2.0		0
■ Advanced Replication	12.1.0.2.0		0
■ Application Express	12.1.0.2.0		0
■ Automatic Maintenance - Optimizer Statistics Gathering	12.1.0.2.0		0
■ Automatic Maintenance - SQL Tuning Advisor	12.1.0.2.0		0
■ Automatic Maintenance - Space Advisor	12.1.0.2.0		0
■ Automatic Memory Tuning	12.1.0.2.0		0
■ Automatic Reoptimization	12.1.0.2.0		0
■ Automatic SGA Tuning	12.1.0.2.0		0
■ Automatic SQL Execution Memory	12.1.0.2.0		0
■ Automatic SQL Tuning Advisor	12.1.0.2.0		0

Figure 9-27. EM Express Database Feature Usage page

The other tab (High Water Marks) on the same page is shown in Figure 9-28. It provides details about the frequency with which a particular feature is used.

Database Feature Usage

Page Refreshed 9:53:46 AM GMT

This screenshot shows the 'Database Feature Usage' section of the Oracle Enterprise Manager Express interface. It displays a table of system statistics with columns for Name, High Water Mark, Last Value, and Database Version. Key entries include Maximum Number of Active Sessions, Maximum Number of CPUs, Maximum Number of Concurrent Sessions, Maximum Number of Datatypes, Maximum Number of Materialized Views, Maximum Number of Partitions belonging to an User Index, Maximum Number of Partitions belonging to an User Table, Maximum Number of SQL/IC/MR Columns, Maximum Number of Tablespaces, Maximum Query Length, Maximum Size of the Database (Bytes), Number of User Indexes, Number of User Tables, Number of Global Services, Number of job runs per day, Number of physical disks, and Oracle Database instances.

Name	High Water Mark	Last Value	Database Vers.
Maximum Number of Active Sessions seen in the system	0	0	12.1.0.2.0
Maximum Number of CPUs	2	2	12.1.0.2.0
Maximum Number of Concurrent Sessions seen in the database	23	23	12.1.0.2.0
Maximum Number of Datatypes	4	4	12.1.0.2.0
Maximum Number of Materialized Views (User)	0	0	12.1.0.2.0
Maximum Number of Partitions belonging to an User Index	1	1	12.1.0.2.0
Maximum Number of Partitions belonging to an User Table	1	1	12.1.0.2.0
Maximum Number of SQL/IC/MR Columns	113	113	12.1.0.2.0
Maximum Number of Tablespaces	5	5	12.1.0.2.0
Maximum Query Length	1516	1516	12.1.0.2.0
Maximum Size of the Database (Bytes)	21128805640	21128805640	12.1.0.2.0
Number of User Indexes	3720	3720	12.1.0.2.0
Number of User Tables	954	954	12.1.0.2.0
Number of Global Services	0	0	12.1.0.2.0
Number of job runs per day	347	347	12.1.0.2.0
Number of physical disks	0	0	12.1.0.2.0
Oracle Database instances	1	1	12.1.0.2.0

Figure 9-28. EM Express Database Feature Usage details

The Undo Management option available from the Storage menu is useful for viewing Undo settings and statistics, as shown in Figure 9-29. Charts like Undo Advisor and Undo Generation Rate can be used to plan your undo space.

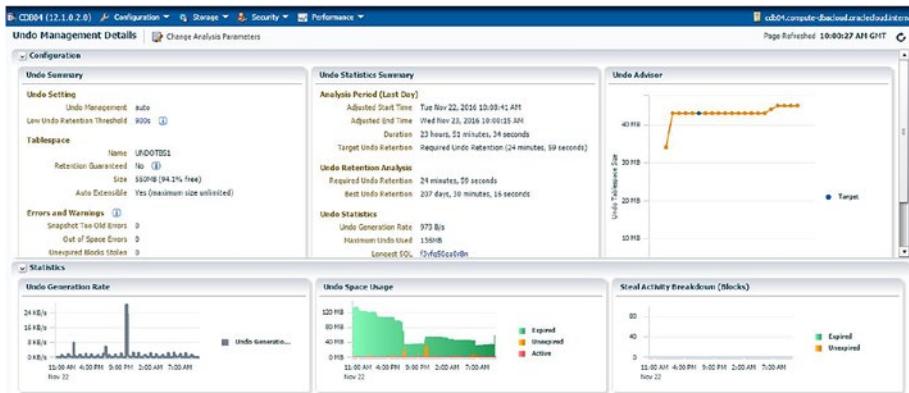


Figure 9-29. EM Express undo management details

You can also run the analysis for a specified duration and tune your undo allocation per need, as shown in Figure 9-30.

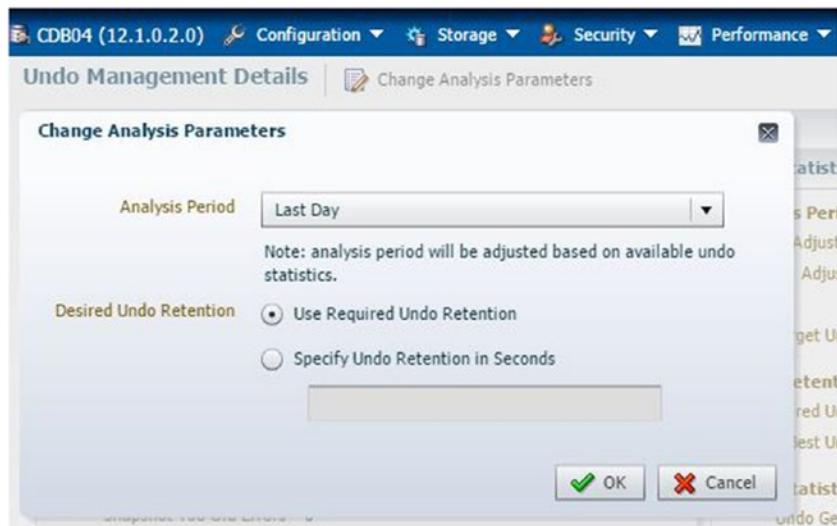


Figure 9-30. EM Express undo analyzer

The Control Files option, as shown in Figure 9-31, provides details on control file space consumption by various types and can also be used to back up the control file to trace.

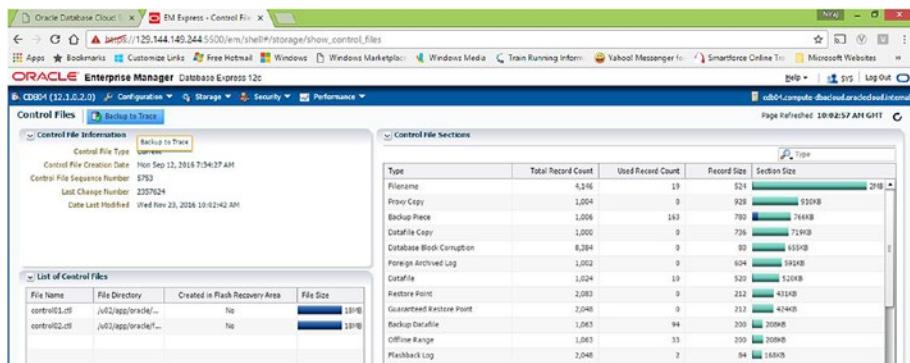


Figure 9-31. EM Express control files management

User administration and access control can be done using the Users menu on the Security tab, as shown in Figure 9-32. You can create, view, drop, or modify users and manage privileges using this option.

	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace
		Mon Jul 7, 2014 6:52:04 AM*	SYSAUX	TEMP
		Mon Sep 12, 2016 8:13:59 AM*	SYSAUX	TEMP
		Sun May 21, 2017 3:20:16 AM*	USERS	TEMP
		Sun May 21, 2017 3:20:16 AM*	USERS	TEMP
		Sun May 21, 2017 3:20:16 AM*	USERS	TEMP
		Mon Jul 7, 2014 6:03:38 AM*	SYSAUX	TEMP
		Mon Jul 7, 2014 5:39:13 AM	USERS	TEMP

Figure 9-32. EM Express user management

Similar to users, roles can also be managed very easily using this GUI, as shown in Figure 9-33.

AQ_USER_ROLE
AUDIT_ADMIN

Figure 9-33. EM Express roles management

The Performance Hub is a set of metrics that helps you understand the health of your database, as shown in Figure 9-34. It has multiple options and can assist you in drilling down and identifying the root cause of a performance bottleneck.

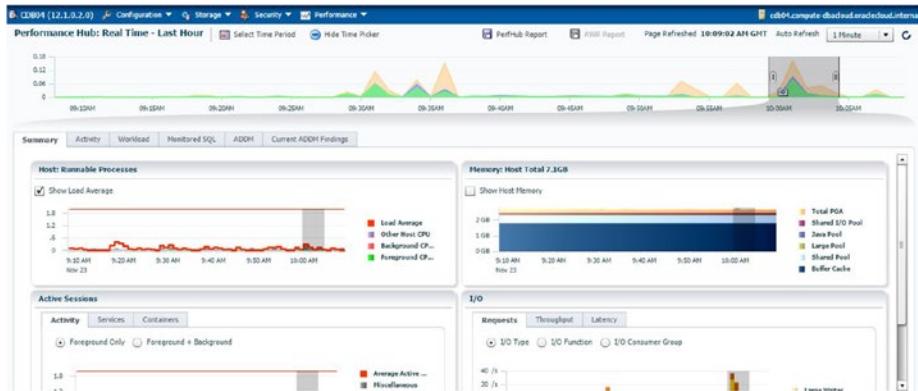


Figure 9-34. EM Express performance hub

Figure 9-35 shows the wait details for a particular duration, broken down into various types of resources like CPU, I/O, network, etc.

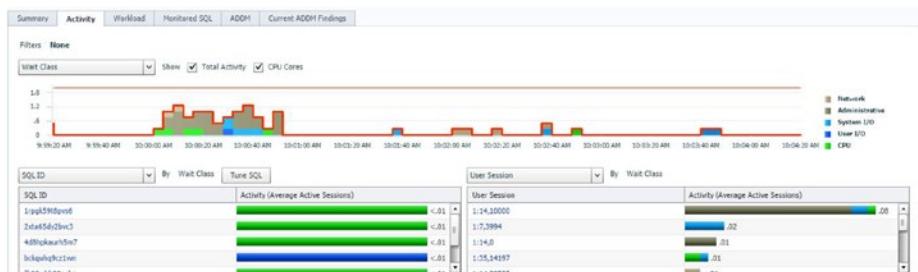


Figure 9-35. EM Express session activity details

The Workload tab provides a detailed view on user calls, session details, and linked SQLs, as shown in Figure 9-36.

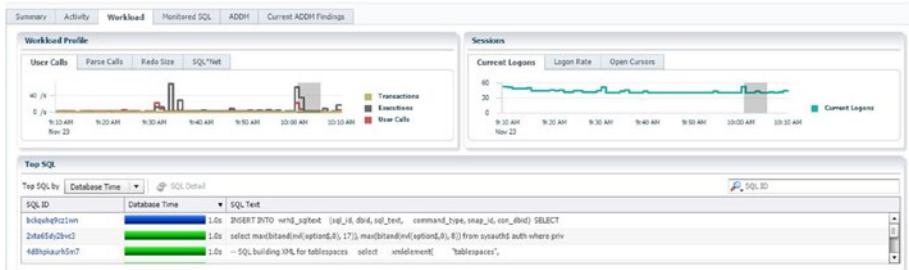


Figure 9-36. EM Express workload details

The other very useful feature of the EM Console is the SQL Tuning Advisor. It analyzes all the SQLs executed against the database and provides suggestions for tuning with potential benefits, as shown in Figure 9-37.



Figure 9-37. EM Express SQL Tuning Advisor

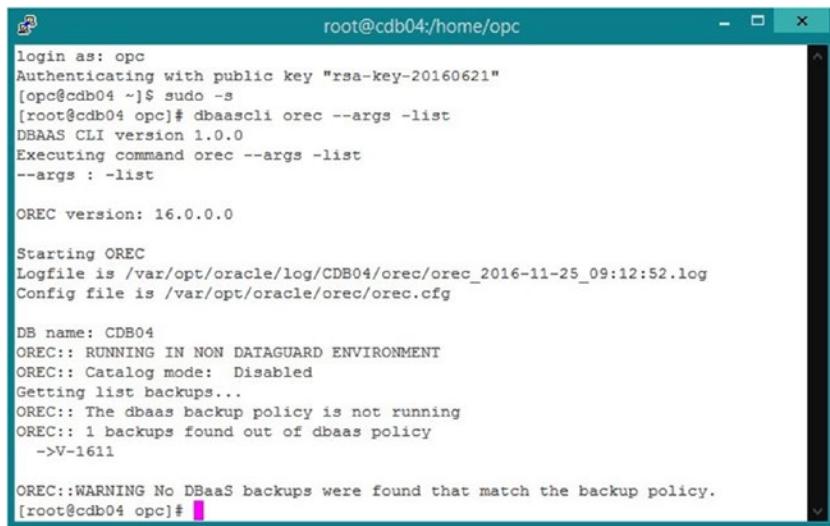
This concludes the discussion of EM Express. As you have read, EM Express is a very effective tool for monitoring and will help you troubleshoot and analyze root causes.

Command-Line Interfaces

Using the dbaascli utility, you can perform the following operations:

- Change the password of the SYS user
- Check the status of Oracle DataGuard configuration
- Perform switchover and failover in Oracle DataGuard configuration
- Patch the database deployment
- Recover a database
- Rotate the master encryption key

When you connect to the host (compute node) on Oracle Cloud using PuTTY, this utility is readily available to you as the root or Oracle user. See Figure 9-38 for details on how to log in.



```

root@cdb04:/home/opc
login as: opc
Authenticating with public key "rsa-key-20160621"
[opc@cdb04 ~]$ sudo -s
[root@cdb04 opc]# dbaascli orec --args -list
DBAAS CLI version 1.0.0
Executing command orec --args -list
--args : -list

OREC version: 16.0.0.0

Starting OREC
LogFile is /var/opt/oracle/log/CDB04/orec/orec_2016-11-25_09:12:52.log
Config file is /var/opt/oracle/orec/orec.cfg

DB name: CDB04
OREC:: RUNNING IN NON DATAGUARD ENVIRONMENT
OREC:: Catalog mode: Disabled
Getting list backups...
OREC:: The dbaas backup policy is not running
OREC:: 1 backups found out of dbaas policy
->V-1611

OREC:::WARNING No DBaaS backups were found that match the backup policy.
[root@cdb04 opc]#

```

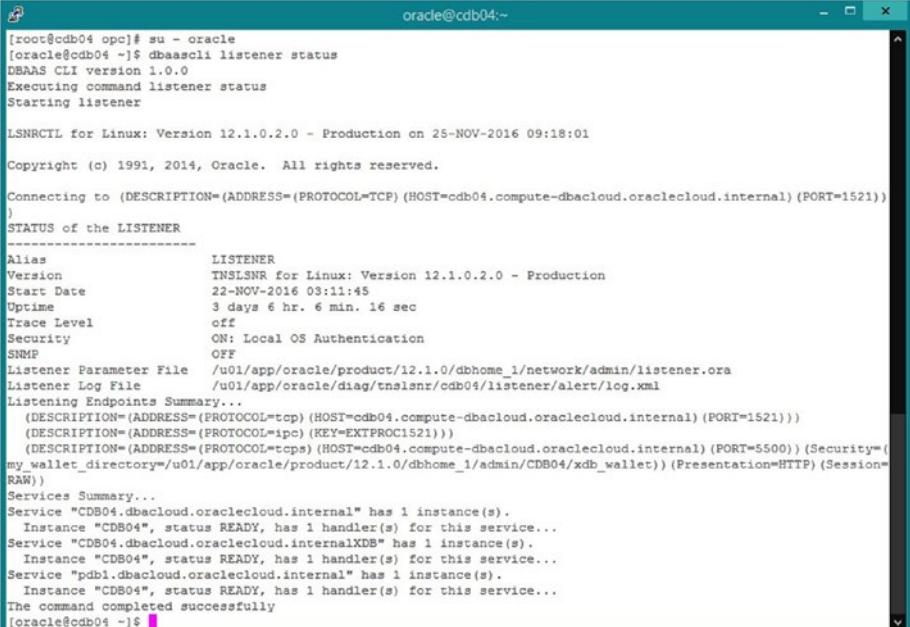
Figure 9-38. DBaaS CLI

Patch management and recovery activities need to be performed as the root. All the other activities listed here can be performed as Oracle user:

- Database restart
- Password reset
- DataGuard switchover/failover
- Database vault enable/disable

- Golden Gate replication setup
- Listener management, such as start/stop
- Network security configurations, such as encryption
- TDE master encryption key change

As shown in Figure 9-39, you can get listener status using the dbaascli utility.



```
oracle@cdb04:~$ su - oracle
[oracle@cdb04 ~]$ dbaascli listener status
DBaaS CLI version 1.0.0
Executing command listener status
Starting listener

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 25-NOV-2016 09:18:01
Copyright (c) 1991, 2014, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=cdb04.compute-dbacloud.oraclecloud.internal)(PORT=1521)))
STATUS of the LISTENER
-----
Alias           LISTENER
Version        TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date     22-NOV-2016 03:11:45
Uptime         3 days 6 hr. 6 min. 16 sec
Trace Level    off
Security       ON: Local OS Authentication
SNMP           OFF
Listener Parameter File  /u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File   /u01/app/oracle/diag/tnslsnr/cdb04/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=cdb04.compute-dbacloud.oraclecloud.internal)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=cdb04.compute-dbacloud.oraclecloud.internal)(PORT=5500))(Security=(my_wallet_directory=/u01/app/oracle/product/12.1.0/dbhome_1/admin/CDB04/xdb_wallet))(Presentation=HTTP)(Session=RAM))
Services Summary...
Service "CDB04.dbacloud.oraclecloud.internal" has 1 instance(s).
  Instance "CDB04", status READY, has 1 handler(s) for this service...
Service "CDB04.dbacloud.oraclecloud.internalXDB" has 1 instance(s).
  Instance "CDB04", status READY, has 1 handler(s) for this service...
Service "pdb1.dbacloud.oraclecloud.internal" has 1 instance(s).
  Instance "CDB04", status READY, has 1 handler(s) for this service...
The command completed successfully
[oracle@cdb04 ~]$
```

Figure 9-39. DBaaS CLI checking listener status

Similar to the dbaascli utility, raccli is also readily available when you do RAC deployment in Oracle Cloud. raccli can be used for backup, recovery, patching, and security management. This utility is covered in Chapter 4, “High Availability Options.”

oracle-dbcs-cli is available only for Linux and can be downloaded using your OTN account, as shown in Figure 9-40.

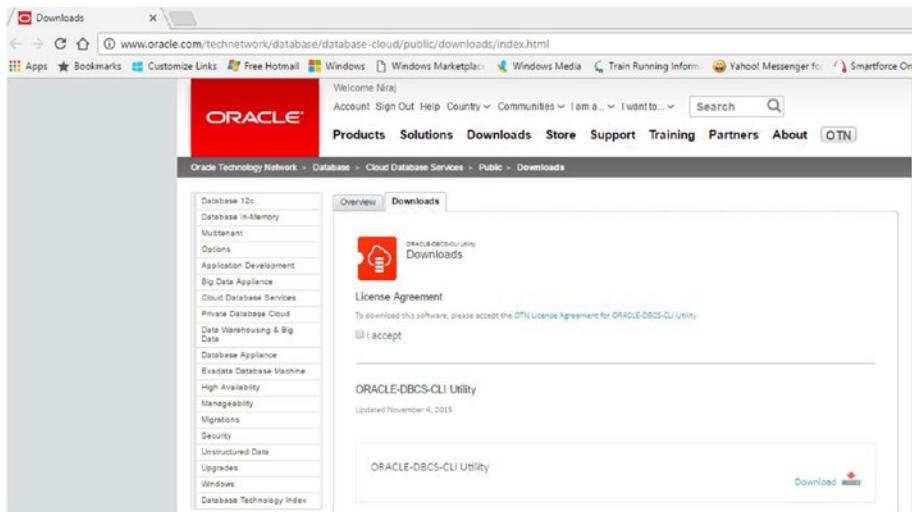


Figure 9-40. Oracle DBCS CLI download

With this utility, you can perform multiple management and operational activities:

- *Create*: Create a new CDB in Oracle Cloud.
- *Delete*: Delete an existing cloud database instance.
- *List*: Get the details of all/specific instance(s) in your domain.
- *Patch Apply*: Patch your cloud database.
- *Patch Check*: Check the prerequisites before you patch a database.
- *Patch Rollback*: Roll back a patch.
- *Patch Status*: Show the patch status for a database.
- *Patch List*: List all the patches you can apply to your database.
- *Scaleup*: Increase the compute capacity for the host of your database instance.

You get much finer control when using command-line interfaces and can define many configurations compared to using GUI options. Oracle official documentation on `oracle-dbsc-cli` is very helpful for understanding the full potential of this command-line interface. It is recommended that you refer to the documentation before using the interface.

SQL Developer

You can connect to SQL Developer by giving the information similar to what's shown in Figure 9-41. Once you are connected, you can perform all the operations similar to what you do with your local (on-premise) databases. The same is true when using TOAD.

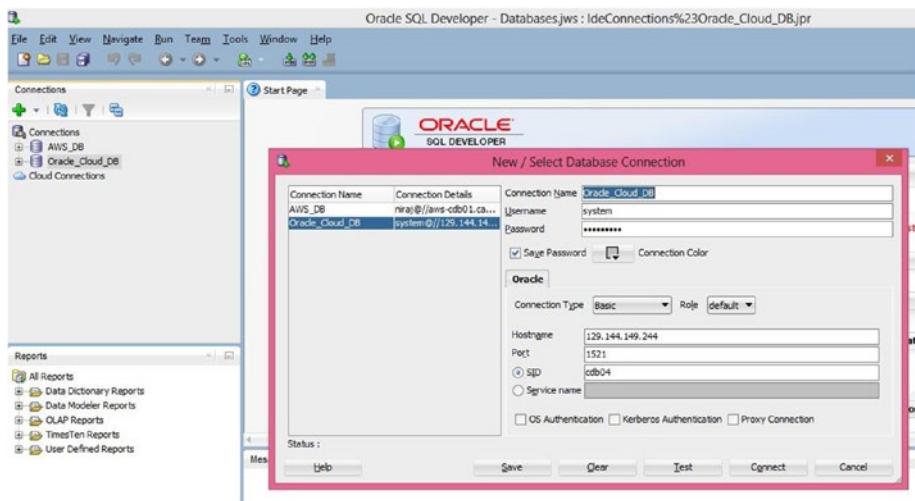


Figure 9-41. Connecting using SQL Developer

You can use SQL Developer to migrate data from a local database to the cloud database or vice versa. You can also use this feature to move data from one cloud provider to another, as shown in Figure 9-42.

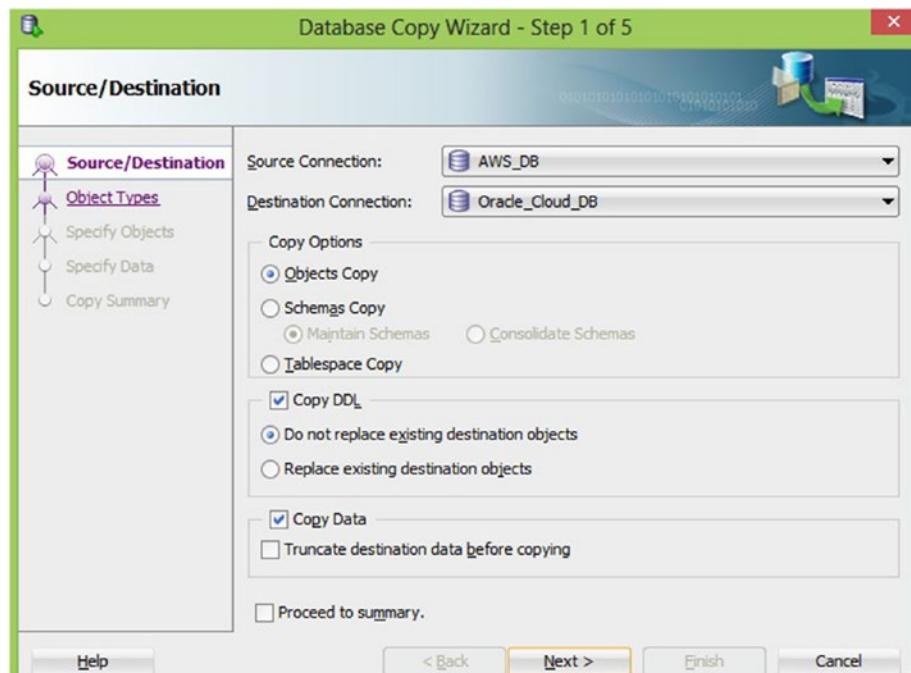


Figure 9-42. Database Copy wizard

This Database Copy feature is available from the Tools menu.

You can use the Clone PDB to Oracle Cloud option to move your PDB from the local database to an existing database in Oracle Cloud, as shown in Figure 9-43.

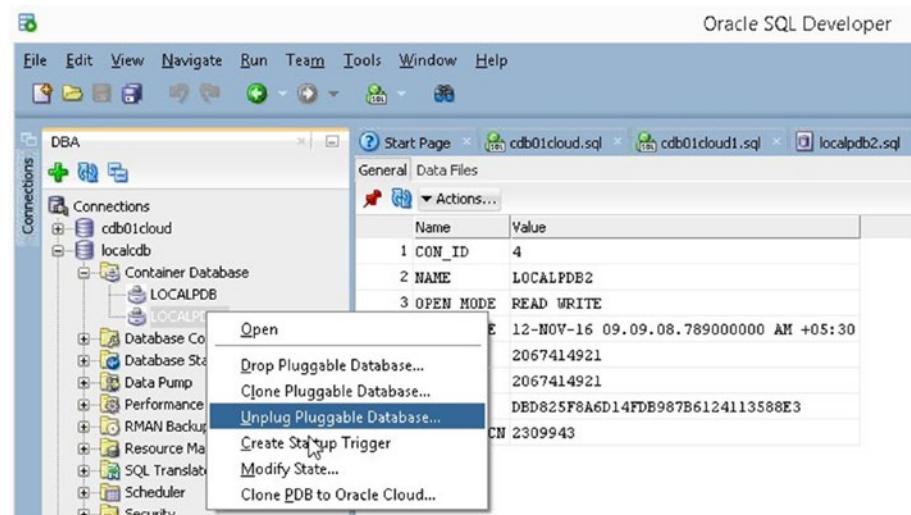


Figure 9-43. Moving a clone PDB to Oracle Cloud

SQL Developer is a free tool available from Oracle with very good features and is cloud-ready as well.

DB Monitoring and Management in AWS

SQL Developer can be used to connect to the databases in AWS, as shown in Figure 9-44. Once you're connected, you can perform activities as if it were a local database.

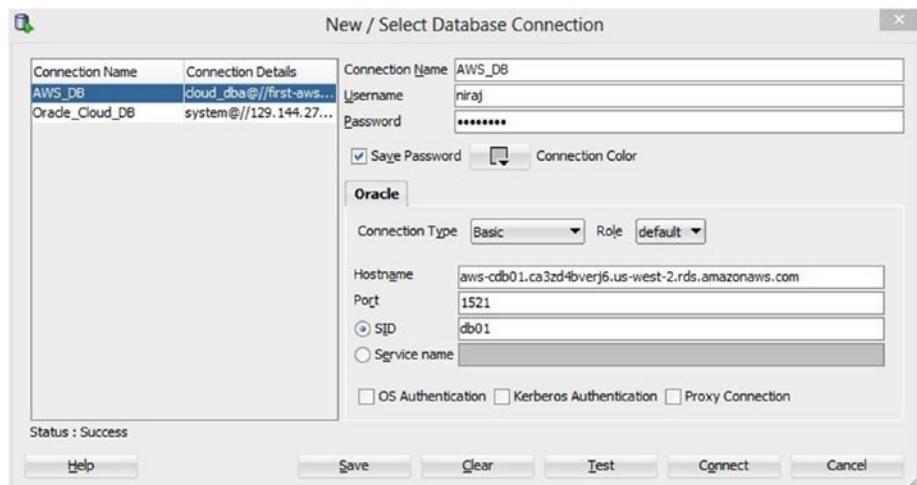


Figure 9-44. SQL Developer connection to Oracle in AWS

AWS RDS also provides a web GUI to monitor your databases running in RDS. You can monitor a single database or select multiple databases under the same VPC to be monitored together, as shown in Figure 9-45.

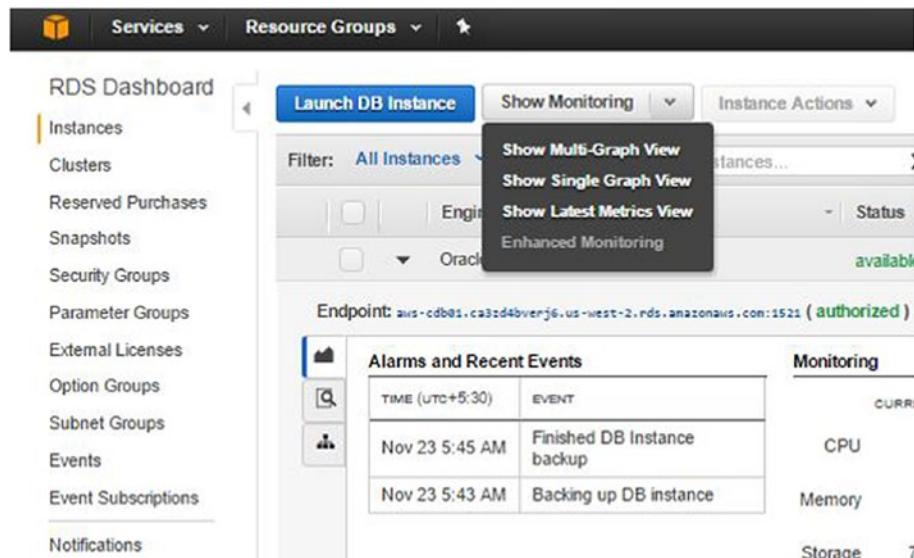


Figure 9-45. AWS monitoring display options

This AWS-provided monitoring tool can capture multiple metrics, providing insight about the database's health. This covers CPU utilization, I/O details, session details, storage consumption, and more (Figure 9-46).

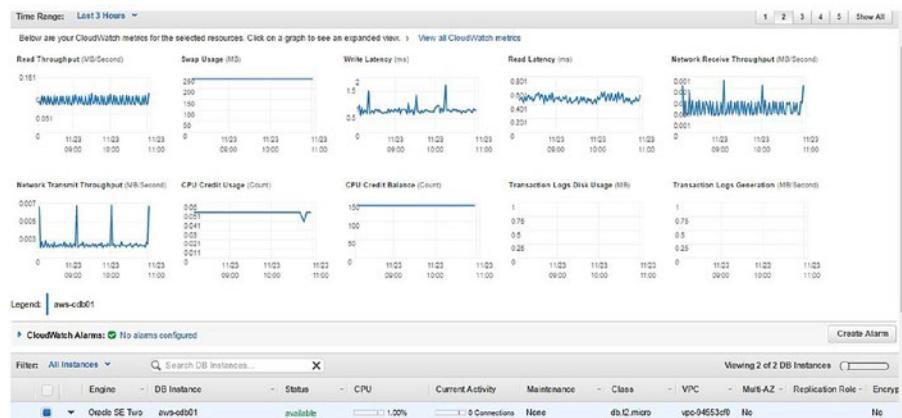


Figure 9-46. AWS monitoring metrics

AWS RDS Monitoring Using OEM

OEM Express 12c is a very effective tool for database monitoring. You have seen how it can be used for monitoring your database in Oracle Cloud. Along similar lines, you can use it to monitor your AWS RDS databases. As the first step, you need to create an option group, as shown in Figure 9-47.

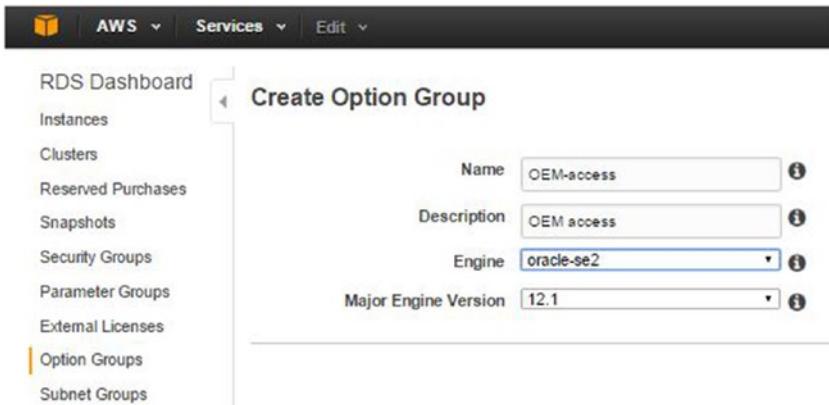


Figure 9-47. Using OEM Express to create an option group

Once you select the correct engine and version and create the group, you must use this newly created group to set up OEM access with the Add Option button, as shown in Figure 9-48.

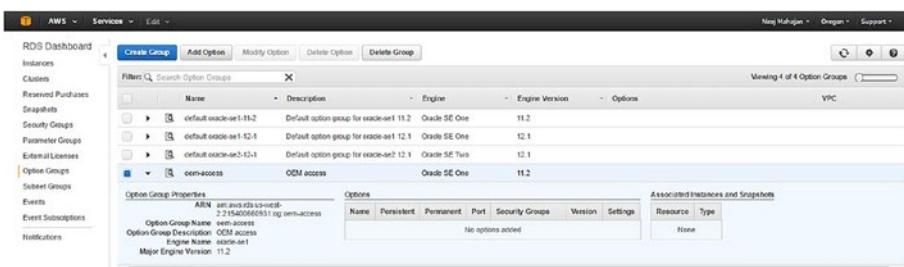


Figure 9-48. Using the OEM Express option, group details

Under Add Option, select OEM to enable access to your RDS database using OEM. There are other options as well, as shown in Figure 9-49.

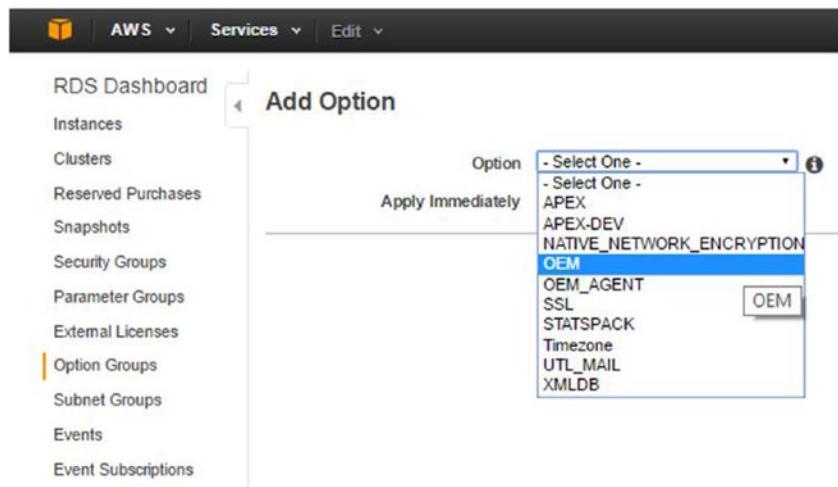


Figure 9-49. Using OEM Express to enable OEM access

Select the security group. You can apply the changes immediately, which requires database downtime, otherwise the changes will be applied during the next scheduled maintenance window. See Figure 9-50, where the Apply Immediately option is selected.

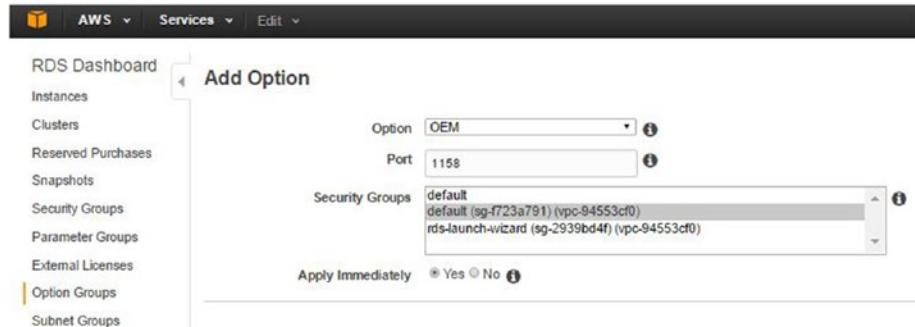


Figure 9-50. Using OEM Express and the add OEM option

To use this option group for database access, you need to modify the database properties and set the option group to the newly created OEM group, as shown in Figure 9-51.

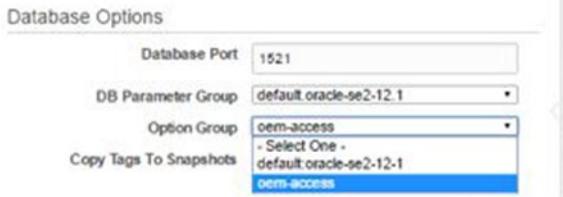


Figure 9-51. Using OEM Express for mapping the OEM option group

Figure 9-52 shows that the database is unavailable and the changes are being applied immediately.

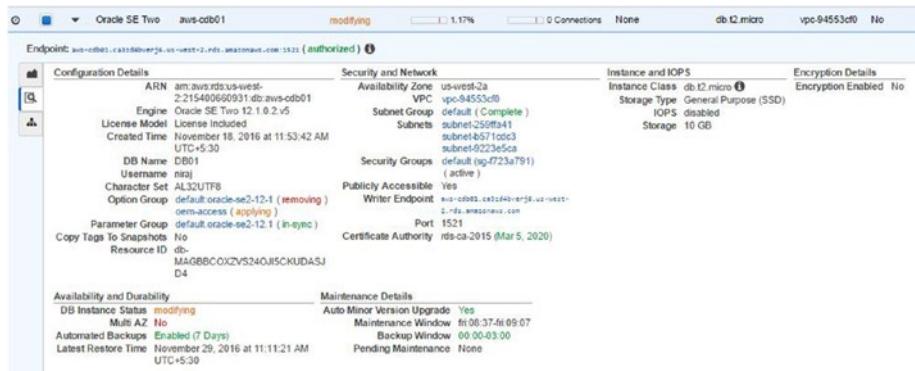


Figure 9-52. Using OEM Express: OEM option enabling in progress

This includes the removal of the existing group and the addition of a new option group. This activity can be done for multiple RDS databases to monitor them using OEM.

Once all the settings are completed, you are ready to use OEM to administer your RDS database. The group linkage with the database option is visible, as shown in Figure 9-53.



Figure 9-53. Using OEM Express: option group linkage details

To access the OEM, the link is `https://<END POINT URL>:<Option group PORT>/em/console/logon/logon`.

Replace the endpoint URL for your database and the port number used for OEM, as visible in the previous two figures.

`https://aws-cdb01.xxxxx.us-west-2.rds.amazonaws.com:5500/em/?login?returnUrl=/em/console/logon/logon`

You need to log in to the OEM using the master username and the password provided at the time of database creation. There is no SYSDBA access provided in Amazon RDS. OEM, as always, is a very powerful tool and provides lots of information and metrics for your database, as shown in Figure 9-54.



Figure 9-54. Using OEM Express: the OEM dashboard

Figure 9-55 provides details for tablespaces for the RDS database. In a similar fashion, you can explore all the metrics and drill down to collect more details for your RDS database.



Figure 9-55. Using OEM Express: table space details

AWS CloudWatch

AWS CloudWatch is another very effective tool that is readily available for monitoring, not only for databases but also for all infrastructure components hosted on AWS.

You can search for CloudWatch by using the quick search menu that is present under AWS Services (Figure 9-56).

AWS services

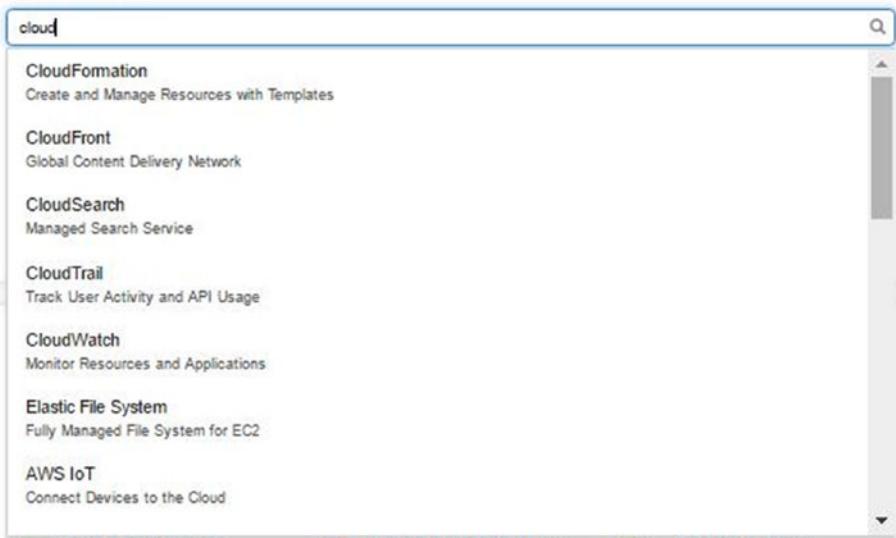


Figure 9-56. CloudWatch services quick search

Multiple metrics are readily available. Based on your AWS services, you can decide to set up your CloudWatch monitoring. As shown in Figure 9-57, you can configure an alarm for CPU utilization or for any other parameter, as needed.

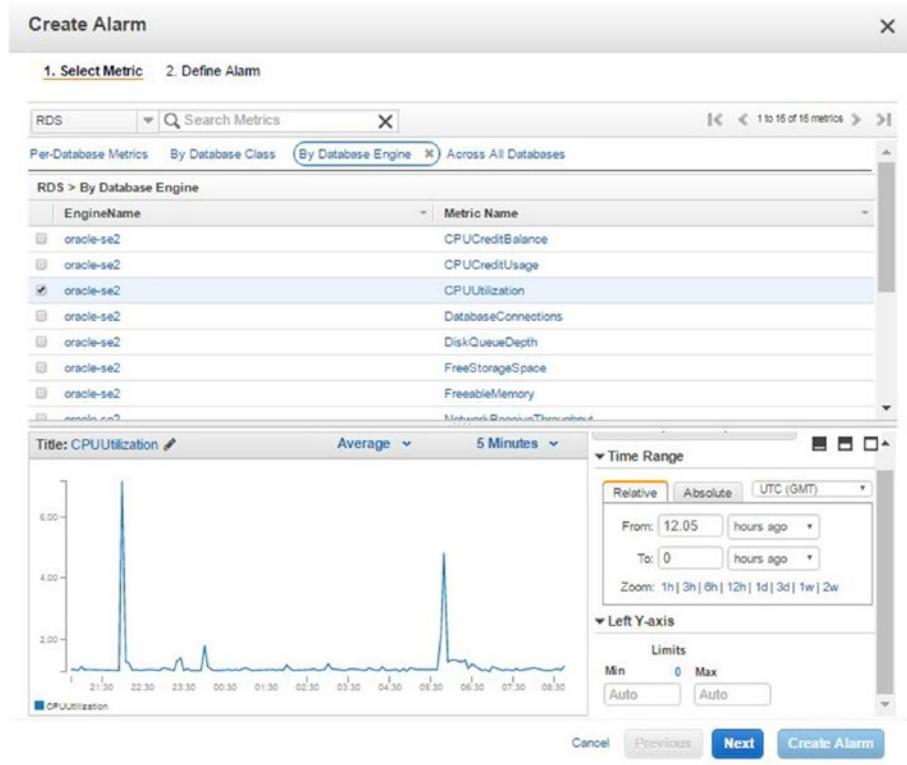


Figure 9-57. Using CloudWatch to create an alarm

Figure 9-57 shows the CPU utilization details. Once a metric is selected for monitoring, you can set the thresholds and set one or more actions that are expected to be performed when the threshold is breached.

As shown in Figure 9-58, you can set an e-mail notification to the primary DBA group when CPU utilization shoots up.

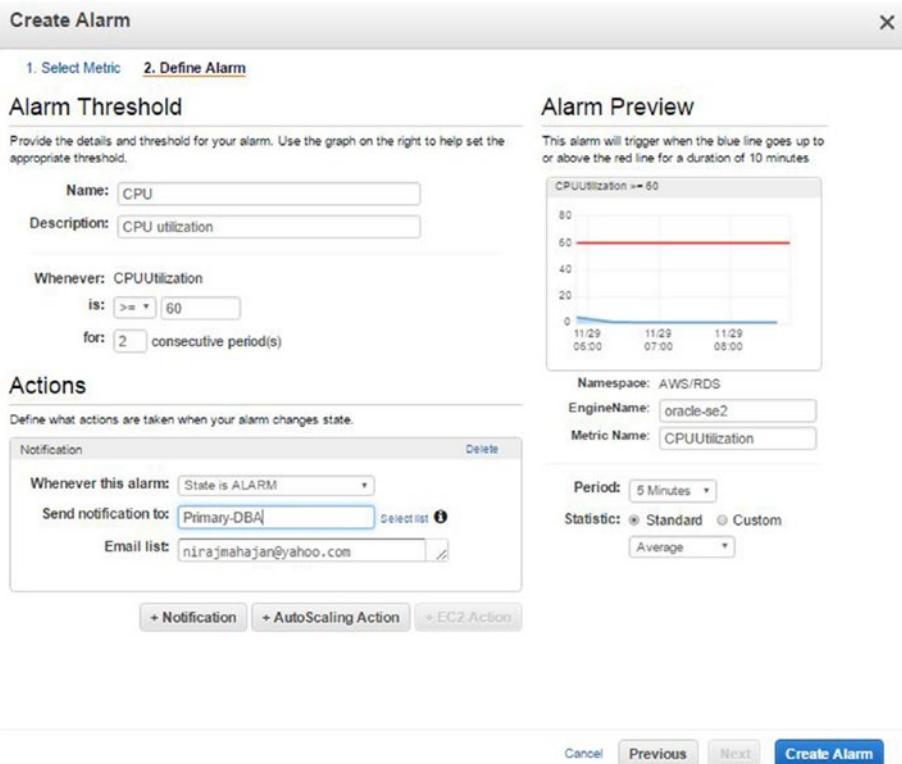


Figure 9-58. Using CloudWatch to define an alarm

Once the alarm is created, an e-mail is sent to the intended recipient, as shown in Figure 9-59. The respective user must confirm the subscription.

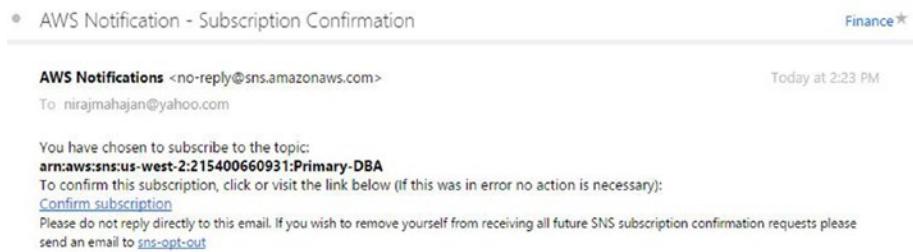


Figure 9-59. AWS subscription confirmation notice

Once the subscription is confirmed, confirmed users will receive e-mail notification about violation of the threshold. A confirmation sample is shown in Figure 9-60.

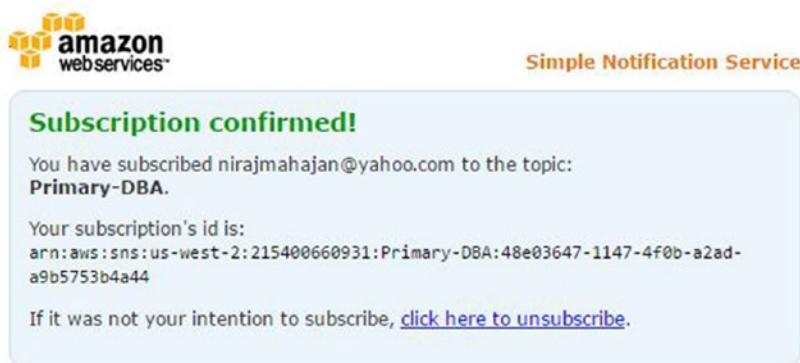


Figure 9-60. AWS subscription confirmation message

You can see the details of the alarms that are set and modify/copy/delete them as required. The alarm frequency in this case is five minutes, as shown in Figure 9-61.

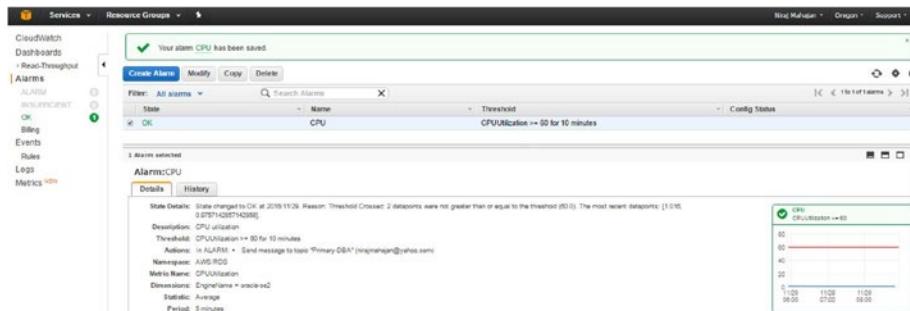


Figure 9-61. CloudWatch alarm details

CloudWatch is a very handy tool and is readily available with AWS. You can use it to monitor various parameters of database health as well as for other infrastructure components. Remember, it is a paid option.

DB Monitoring and Management Best Practices

The following list includes some of the monitoring and management best practices for Oracle database running on Oracle Cloud and AWS:

- Use on-premise tools whenever possible to save licensing costs related to buying new tools on the cloud.
- Use the latest versions of tools such as SQL Developer, TOAD, etc., as older versions do not have cloud DB support or cloud DB-related features.
- Use CloudWatch for monitoring at shorter intervals. Use a notification service along with CloudWatch.
- Use command-line utilities (such as DBaaSCLI and RACCLI) for automation and a wide variety of features.
- Use an OEM kind of tool to monitor and manage multiple databases from a single screen.
- Be careful while opening the required ports for tool access.

Summary

In this chapter, you learned how DB monitoring and management differs for a cloud database and learned about the various tools and options available in Oracle Cloud and Amazon Cloud.

After completing this chapter, you know about various tools like DBaaS Monitor, EM Express, and command-line interfaces for Oracle Cloud databases. You read how traditional tools like OEM and SQL Developer can be used for managing and monitoring AWS and Oracle Cloud databases. Additionally, you now know about AWS-specific tools like CloudWatch and AWS database metrics.

Epilogue

Congratulations—you have reached the end of the book! Let's do a quick recap of what you learned in this book.

In Part I, you learned about cloud computing fundamentals that are essential for becoming “The Cloud DBA” and in Part II, you learned how to manage Oracle database in the cloud.

In Chapter 1, you learned the meaning of cloud computing, its benefits, challenges associated with cloud computing, cloud computing service models, and deployment models. At the end of that chapter, you gained a solid understanding of cloud computing basics.

In Chapter 2, you learned the meaning of cloud computing from a database perspective by exploring Database as a Service (DBaaS) concepts and the various deployment models and available offerings. You also learned when to choose which offering from Oracle Cloud and Amazon Web Services (AWS). At the end, you read about the various terminologies used in Oracle Cloud, AWS, and Azure. After completing Part I (Chapters 1 and 2), you started Part II, where the actual implementation is covered.

In Chapter 3, you learned about the various provisioning options, supported editions and versions, the available virtual machine types and available offerings, etc. At the end, you created your first database in the cloud and connected to it using various tools.

In Chapter 4, you went through various high-availability options in Oracle Cloud and AWS. You also learned how to implement Oracle RAC in Oracle Cloud and deploy Oracle database in a Multi-AZ scenario. At this point in the book, you have a fair idea of how Oracle high-availability is ensured in the cloud.

In Chapter 5, you went through various disaster recovery options in Oracle Cloud and AWS. You learned how to implement Oracle DataGuard in Oracle Cloud, along with how to switch over and reinstate a failed primary DB.

In Chapter 6, you learned the cloud security model and read about the key considerations for Oracle database security in the cloud.

In Chapter 7, you learned about the most challenging job of database migration to the cloud. This chapter covered migration considerations, migration options, and the steps to migrating PDB to the cloud.

In Chapter 8, you learned how Oracle database backup, restore, and recovery works in the cloud. You also saw a few examples of point-in-time recovery.

In this last chapter, you learned how to monitor your database in Oracle Cloud and AWS. You also learned about various tools that are available for day-to-day administration and looked at many of the features of these tools.

Thanks for reading this book. We hope this book was useful in enabling you to become “The Cloud DBA”.

Index

A

Account-level security, 91
Administration. *See* Monitoring and management
Amazon EC2, 95
 correct source IP, 96
 dedicated hosts, 97
 encrypted EBS volume, 95
 VPC and private subnets, 96–97
Amazon RDS
 advanced settings, 93
 enable encryption, 94
 master user, 93
Amazon Web Services (AWS)
 backup and restore
 EC2, 172
 on-premise backup, 162
 options, 162
 RDS, 167
 choices, 18
 DB instance, 39
 disaster recovery (DR), 85
 EC2, 16
 engine selection, 39–40
 environment specification, 41
 instance specification, 41
 licensing model, 17
 monitoring and management
 CloudWatch, 204
 SQL developer, 199
 Oracle RAC provisioning, 64
 provisioning DB
 advanced settings, 44
 availability page, 46
 AWS RDS instance, 47
 configuration, 45
 creation, 45

details, 41
inbound rule, 48
instance name, 44
monitoring page, 47
query execution, 49–50
server options, 44
SQL developer, 49
status creation, 46
version options, 43
RDS, 17, 39
snowball, 135
step process, 39
steps, 16
terminologies, 21
versions and editions, 17
ARCHIVELOG mode, 99

B

Backup and restore processes, 139. *See also* Recovery database
AWS
 EC2, 172
 on-premise database, 162
 options, 162
 RDS, 167
 benefits, 140
 best practices, 172
 ODBS, 142
 offering comparison, 141
 on-premise DB backup approaches, 140
 opc installation, 143
 Oracle database backup service
 backups list, 154
 configuration, 150
 confirmation page, 151
 customization, 152
 in progress, 153

Backup and restore processes (*cont.*)
on-demand backup, 153
password, 154
REST endpoint identification, 149
OTN page, 142
requirements, 139, 141
snapshot backups
completion, 149
confirmation page, 148
creation, 147
detail page, 147
in progress, 148

C

Capital expenditure (CAPEX), 4
Chargebackmodel. *See* Metering and chargeback model
Cloning, 118
close LocalPDB, 123
EM express-plug LocalPDB
CDB home page, 130
console screen, 129
home page, 129
plug PDB, 130
high-level steps, 118
LocalPDB, 120, 131
multitenant setup, 119
objective, 118
Oracle 12c PDB database, 120
READONLY mode, 123
READWRITE mode, 122
tools, 118
unplug LocalPDB, 124
XML and data files
file copy and permissions, 128
file transfer screen, 127
private key file, 126
WinSCP, 124-125

Cloud computing, 3
benefits of, 4
challenges, 4
definition, 3
deployment (*see* Deployment models)
metering and chargeback concepts, 10
overview, 3
service models
IaaS, 5
PaaS, 6
SaaS, 6

Cloud service, 51
CloudWatch, 204
alarm creation, 206
alarm details, 208
confirmation message, 208
define alarm, 207
services quick search, 205
subscription confirmation notice, 207
Command-line interfaces
activities, 196
checking listener status, 195
DBaaS CLI, 194
download, 196
operation, 194
patch management and recovery activities, 194
Community cloud model, 9

D

Database as a service (DBaaS), 11
definition, 11
key concepts, 11
managed services model, 19
Oracle database
choices, 14
Oracle virtual image shapes, 15
service levels, 14
tools, 14
private cloud, 19
public cloud offering
AWS, 16
Microsoft Azure Cloud, 18
Oracle, 12
service levels, 13
service models, 20
steps, 20
subscription, 20
terminologies
Amazon Web Services, 21
Microsoft Azure, 22
Oracle, 21
vendor selection, 20
virtual image, 13
Database migration flowchart, 136
Database migration service (DMS)
AWS schema conversion tool
console, 132
database driver download, 134
download, 133
drivers setting, 135

installation, 133
 project creation, 134
DataGuard provision
 dbaascli commands, 78
 DB switchover operation, 76
 DG configuration details, 76
 edition selection, 74
 failover operation, 82
 primary database, 83
 RAC configuration, 73
 release selection, 74
 service level, 73
 standby database, 75
Deployment models
 community cloud model, 9
 hybrid cloud, 8
 private cloud, 8
 public cloud, 6
 types, 6
Disaster recovery (DR), 71
 AWS, 85
 cloud service, 71
 configuration, 72
DataGuard (see DataGuard provision)
 options, 72
 overview, 71
 software releases and editions, 73
 standby database, 73

■ E, F, G

Elastic Compute Cloud (EC2), 16
Enterprise Manager (EM) express
 control files management, 190
 dashboard, 186
 database feature
 details, 189
 page, 188
 initialization parameter change, 187
 memory management, 187
 menu pages, 187
 options, 186
 performance dashboard, 186
 performance hub, 192
 roles management, 191
 session activity details, 192
 sign in, 185
 SQL tuning advisor, 193
 undo analyzer, 190
 undo management details, 189
 user management, 191
 workload details, 193

■ H

High availability (HA), 51
 AWS, 64
 cloud service, 51
 MAA and option, 52
 multi-AZ deployment (*see* Multiple-availability (AZ) deployment)
 option, 52
 overview, 51
provisioning
 build progress status, 58
 configuration details, 58
 confirmation, 57
 edition type, 56
 level selection, 55
 login page, 53–54
 RAC option, 56
 service creation, 54
 shape related validation failure, 57
 software release, 55
 steps, 53
PuTTY (see PuTTY)
RAC database
 access rules, 61–62
 error message, 60–61
 output, 62
 query output, 63
 second instance, 63
 testing RAC, 64
Hybrid cloud model, 8

■ I, J, K

Identity and access management (IAM), 91
 AWS console, 91
 dashboard, 92
Infrastructure as a Service (IaaS), 6

■ L

Lesser-known methods, 106

■ M, N

Maximum availability architecture (MAA), 52
Metering and chargeback model, 10
Microsoft Azure cloud, 18
 terminologies, 22
 versions and editions, 18

- Migration, 99
 - approach
 - backup, 102
 - phases, 102
 - restore, 103
 - transfer, 102
 - AWS snowball, 135
 - best practices, 137
 - cloningPDB (*see* Cloning)
 - comparison, 107
 - DMS (*see* Database migration service (DMS))
 - factors/considerations, 99
 - lesser-known methods, 106
 - lifecycle, 101
 - local database
 - Amazon RDS, 107
 - database copy, 112
 - local_machine_user code, 107
 - log file, 113
 - object types, 110
 - schemas selection, 111
 - source/destination selection, 109
 - SQL developer, 109
 - summary screen, 112
 - options, 103
 - Oracle data pump utilities
 - comparison, 117
 - file system, 113
 - Oracle Enterprise Manager (OEM), 104
 - RMAN backup restore, 117
 - SQL developer, 103
 - traditionalmethod (*see* Traditional migration method)
 - Monitoring and management
 - access rules
 - completion, 176
 - confirmation, 176
 - enable option, 175
 - setting, 174–175
 - AWS
 - CloudWatch, 204
 - connection, 199
 - display options, 200
 - metrics, 200
 - OEM, 201
 - best practices, 209
 - command-line interfaces
 - activities, 196
 - checking listener status, 195
 - DBaaS CLI, 194
 - download, 196
 - operation, 194
 - patch and recovery activities, 194
 - DBaaS monitor
 - alerts info, 181
 - backup options, 180
 - console, 176
 - CPU details, 183
 - database info, 177
 - initialization parameters, 182
 - listener options, 179
 - memory details, 183
 - monitoring levels, 185
 - options, 178
 - OS info, 178
 - PDB options, 179
 - processes details, 184
 - real-time SQL Monitor, 182
 - session info, 181
 - sign in, 177
 - storage details, 184
 - storage options, 180
 - waits info, 182
 - EMExpress (*see* Enterprise Manager (EM) express)
 - SQL Developer
 - database copy wizard, 198
 - PDB-Oracle cloud, 198
 - TOAD, 197
 - tools, 173
 - Multiple availability (AZ) deployment
 - automatic failover, 69
 - completion, 70
 - DB reboot, 69
 - reboot, 69
 - method
 - advanced settings, 67
 - completion, 68
 - deployment option, 66
 - request submission, 68
 - selection, 66
 - option, 65

O

 - OEM express 12c
 - add option, 202
 - dashboard, 204
 - enable access, 202

- in progress, 203
- mapping groups, 203
- option group
 - creation, 201
 - details, 201
 - linkage details, 203
- table space details, 204
- On-premise database backup, 162
 - download and configuration, 164
 - RMAN backup channel, 166
 - S3 access
 - completion message, 164
 - fourth step, 164
 - second step, 163
 - third step, 163
 - user creation, 162
- Operating expenditure (OPEX), 4
- Oracle cloud services
 - configuration details
 - backup and recovery
 - configuration, 35
 - create option, 36
 - creation status, 36
 - database configuration, 34
 - details, 36
 - service configuration, 33
 - shape option, 34
 - summary screen, 35
 - database access
 - enable access rules, 37
 - option, 37
 - query execution, 38
 - SQL developer, 38
 - database cloud offering, 29
 - HA option, 53
 - service creation
 - edition, 32
 - process creation, 30
 - release, 32
 - software edition selection, 33
 - step process, 31
 - subscription, 31
 - steps, 26
 - subscription
 - ACCESS details, 29
 - contact details, 27
 - contact verification details, 28
 - DBaaS offering, 27
 - form completion, 28
 - PaaS offering, 26
 - services, 26
- Oracle database
 - choices, 16
 - cloud terminologies, 21
 - Database as a Service, 13
 - exadata service, 15
 - Schema as a Service, 12
 - virtual Image option, 12
- Oracle Database Backup Service (ODBS), 142
- Oracle data pump utilities
 - comparison, 117
 - file system, 113
- Oracle Enterprise Manager (OEM), 104
- Oracle Virtual Machine (OVM), 19

■ P, Q

- Platform as a Service (PaaS), 6
- Private cloud model, 8, 19
- Provisioning, 25
 - Amazon Web Services (AWS)
 - advanced settings, 44
 - AWS RDS instance, 47
 - DB instance, 39
 - details, 41
 - engine selection, 39–40
 - environment specification, 41
 - RDS, 39
 - step process, 39
 - HA option
 - build progress status, 58
 - configuration details, 58
 - confirmation, 57
 - create service, 54
 - edition type, 56
 - level selection, 55
 - login page, 53–54
 - RAC option, 56
 - shape related validation failure, 57
 - software release, 55
 - steps, 53
 - Oraclecloud (*see* Oracle cloud services)
 - overview, 25
 - step-by-step process, 25
- Public cloud model, 6
- DBaaS
 - AWS, 16
 - Microsoft Azure Cloud, 18
 - Oracle, 12

■ INDEX

- PuTTY
 - configuration details, 59–60
 - RAC database, 59
- **R**
 - Real application clusters (RAC), 52
 - Recovery database
 - completion, 158
 - date selection, 157
 - drop table, 156
 - initiation, 156
 - progress, 158
 - PuTTY, 155
 - scenario 2
 - alter table, 161
 - completion, 161
 - date and time, 159
 - in progress, 160
 - table alter, 159
 - time selection, 157
- Recovery Point Objective (RPO), 52
- Recovery Time Objective (RTO), 52
- Relational Database Services
 - (RDS), 7, 16–17, 39
 - completion, 171
 - customization, 171
 - DB instance, 170
 - in progress, 171
 - recovery option, 167
 - restore snapshot button, 170
- snapshot details, 169
- snapshot option, 168
- take snapshot, 169
- time option, 168
- **S**
 - Schema as a Service, 12
 - Security model, 87
 - AWS
 - account-level security, 91
 - Amazon EC2, 95
 - Amazon RDS, 93
 - best practices, 98
 - cloud-based DB, 87
 - components, 87
 - configurations, 88
 - layers, 89
 - SQL*Net encryption, 90
 - TDE, 89
 - steps, 88
 - Software as a Service (SaaS), 6
 - SQL developer, 103
- **T, U, V, W, X, Y, Z**
 - Traditional migration method, 105
 - hard drive, 106
 - import/export, 105
 - RMAN backup restore, 106
 - Transparent data encryption (TDE), 89