

DATABASE BACKUP AND RECOVERY MANAGEMENT IN CLOUD

Mr. Navendu Mishra¹
Team Leader

Osource Global Private Limited, Mumbai, India.

Mr. Jayant Sawarkar²
Assistant Professor

Datta Meghe College of Engineering, Navi Mumbai, India.

Mr. Suraj Dubey³
COO,

Osource Global Private Limited, Mumbai, India.

Abstract: Managing large chain of hotels and ERP database comprises of core areas such as HRMS & PIP. HRMS (Human Resource Management System), which further includes areas such as Soft Joining, Promotion, Transfer, Confirmation, Leave Attendance and Exit, etc. PIP (Payroll Information Portal), wherein employees can view their individual Salary details, submit investment declaration, Reimbursement claim & CTC structuring, etc. Management of Large Chain of Hotels and ERP Database in AWS Cloud involves continuous monitoring with regards to the areas such as Performance of resource usages and optimization techniques relating to the use of PL/SQL. High Availability (HA) of data is accomplished through the Backup and Recovery mechanism and security of the data by Encryption & Decryption mechanism.

Broad Academic Area of Work: CLOUD COMPUTING

Key words: AWS Cloud, Database, ERP, Hotels, PL/SQL, S3 bucket and Query.

INTRODUCTION

Oracle Database works by utilizing the various components of the Hardware resources such as CPU, Memory (RAM), HDD (SAN-STORAGE), Networks, etc. The Backup and Recovery, plays a very important role in maintaining the overall health of the database.

Most of the organizations prefer to use Oracle database on the Linux flavors world-wide, however, Oracle also provides database services for the Windows Operating System users as well.

Linux is more robust, and from the security point of view is more desirable as compared to the Windows O/S.

Oracle database and RAID works very closely. **RAID** is known as **Redundant Arrays of Inexpensive Disks**. This is used for Load Balancing and data protection across multiple disks. There are various used RAID levels, such as , RAID 0, RAID 1, RAID 0+1, RAID 1+0 or 10 and RAID 5.

RAID 1+0 (RAID 0 and RAID 1), is the best RAID option or Oracle. For Oracle database control Files, Online Redo Logs, Archived Redo Logs, RAID 1+0 is preferred, as need to have multiple copies of the same. The Temporary data files, RAID 0 is preferred, as if the data file is lost, the same can be recreated, as it has no data to restore.

The latest version of 19c provides the database in the form of CDB & PDB's on ASM.

Oracle's **ASM (Automatic Storage Management)** is the most widely used and preferred solution to provide data striping and mirroring. Here ASM disks are protected by the hardware RAID, and are defined using external redundancy and has no failure groups. In general two disk groups namely DATA and FRA are used. DATA stores all the database related files and FRA stores the fast recovery area, which includes multiplexed copies of online redo logs and control files.

One of the best feature called **LVM**, which stands for **Logical Volume Manager**, is a highly efficient used framework in the Linux distributions, owing to the facility of providing logical volume management for the Linux kernel. The basic function of LVM, is to provide allocation of disks, mirroring and striping, and re-sizing the logical volumes.

Also, keeping the data files on the SAN box is mandatory not only for performance (I/O), but also in case any damage happens to any of the disks due to block corruption, the other mirrored disks can be used , and hence, there will not be any data loss or any DB outage.

RAID levels are used for the data redundancy, owing to the fact that the data is always critical.

Oracle Golden Gate is used for the data migration services to replicate data from one database to the other database across different geographic regions world-wide.

Oracle Golden Gate allows to migrate the committed transactions in the database across multiple heterogeneous systems. Oracle Golden Gate comes as a separate software product, which needs installation & configurations.

As I being a DBA (Database Administrator), my core responsibility is to manage all the company's database pertaining to the very important and critical areas such as Database **Backup** and **Recovery** (High Availability), as per the industry standards. The solutions implemented in the database focuses in the core areas of backup and Recovery, keeping in view the wide variety of clients world-wide. Before the implementation of any solutions on the production, the same is first tested on the UAT environment by our Team and in consultation with the IT Team.

Now I am working on the fore-front of the implementation of solutions in the **OCI** (Oracle Cloud Infrastructure) in conjunction with the AWS cloud. Also working for the migration of databases to a higher version which presents various enhanced database features, which helps the company's growth immensely. Another solution which needs to be implemented for the database is the migration of the current database from Non-Container database to **container database (CDB / PDB's)**, and also to utilize the **ATP (Autonomous Transaction Processing)** of Oracle database. **OCI (Oracle Cloud Infrastructure)** provides the **Real Application Clustering (RAC)** environments, wherein the automatic storage management is done by using **ASM disks**, wherein the High Availability (HA) is provided continuously, without any database outage (downtime). This is a very important feature of Oracle database which helps the business to the maximum possible extent.

All the necessary steps are being taken for making sure that the core areas of the database remain within the threshold limit, so that the resource usage is not hampered, and which will automatically enable a better performance of the database in totality, by bringing into the methodologies for the database Backup and Recovery.

AWS RDS/EC2, provides solutions which have been successfully implemented curtailing to the company's business areas world-wide, and which has helped the company to establish itself as a fore-front in the Hospitality, ERP, PAYROLL SERVICES, DMS, BE-SPOKE industry.

As a DBA it's my and my team's responsibility to provide the 24x7 database availability to all of the client's, by following all the DB standards as set in the industry.

Amazon Web Services (AWS) RDS (Remote Database Services), provides various services and which is shown below.

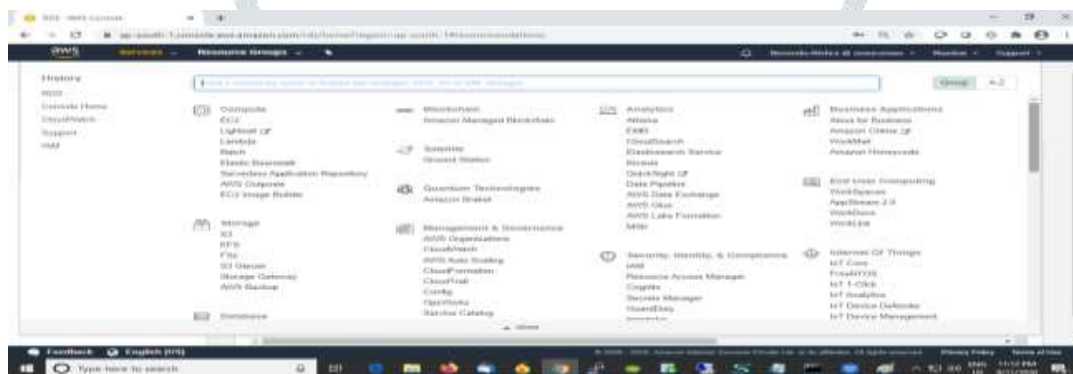


Figure 1.1 Amazon RDS Dashboard

LITERATURE REVIEW

DATABASE BACKUPS

The database backup comprises of two different types of backup mechanisms one **RMAN (Recovery Manager)** which deals with the physical database files backups, and the other is the **DATAPUMP (EXPDP & IMPDP)** utility which deals with the logical (data & meta-data) backups of the schema, tables.

In the Oracle database, **RMAN (Recovery Manager)** is a tool which is the core for taking the physical backups and restoring of the database. It backs up the physical database files in the Oracle binary backup files. There are three different types of RMAN backups one is the **Full Backup**, the second is the **Incremental** backup and the third is the **Archivelog** backups. Full backup also known as Level 0(L0) backups is taken once which takes the full database backups (all database files, controlfile, spfile), and the subsequent, Incremental backups also known as Level 1(L1) backups, are taken as many times as defined in the backup policy, and the third is the **Archivelog** backups which deals with the transactional changes and which are stored in the online redo log files.

The backup policy is defined as per the organizational business rules.

RMAN FULL BACKUP (LEVEL 0)

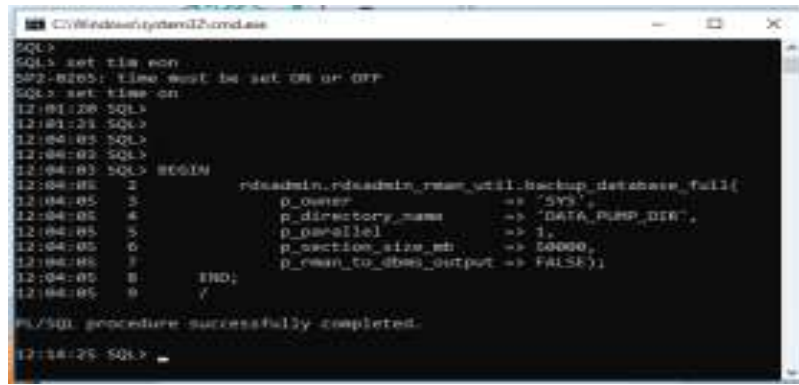


Figure1: Full (Level 0) database backups

AUTOMATED BACKUPS IN AWS RDS

Automated Backups is turned on by default, in AWS RDS. The automated backup feature of AWS RDS backs up the databases and transaction logs. AWS RDS automatically creates a storage volume snapshot of the DB instance, and which backs up the entire DB instance and not just individual databases. This automated backup occurs during a daily user-configurable 30-minute period which is known as the backup window. Automated backups are stored for a number of days and which is configurable and is called as the backup retention period. Automatic backup retention period can be configured to up to thirty-five days in AWS RDS. The automated backups are stored in Amazon S3 for a user-specified retention period (configurable).

Automated backups follow the below mentioned rules.

- DB instance must be in the AVAILABLE state for automated backups to occur. Automated backups don't occur while the DB instance is in a state other than AVAILABLE.
- Automated backups and automated snapshots don't occur when a copy is executing in the same region for the same DB instance.

Automated backups can be temporarily disabled in certain situations, like, the loading large amounts of data in the database.

The following are the limitations of related to retaining automated backups:

- The maximum number of retained automated backups in one AWS Region is 40. It is possible to have 40 running DB instances and an additional 40 retained automated backups at the same time.
- The retained automated backups in AWS RDS, don't contain information about parameters or option groups.
- A retained automated backup can't be modified because it consists of system backups, transaction logs, and the DB instance properties.

The retained automated backups can be deleted when they are no longer needed.

Below table lists the time blocks for each region from which the default backup's windows are assigned.

Backup Window

Region	Time Block
US East (Ohio) Region	03:00-11:00 UTC
US East (N. Virginia) Region	03:00-11:00 UTC
US West (N. California) Region	06:00-14:00 UTC
Asia Pacific (Hong Kong) Region	06:00-14:00 UTC
Asia Pacific (Mumbai) Region	16:30-00:30 UTC
Asia Pacific (Singapore) Region	14:00-22:00 UTC
Asia Pacific (Sydney) Region	12:00-20:00 UTC
Europe (Frankfurt) Region	20:00-04:00 UTC
Europe (London) Region	06:00-14:00 UTC
Europe (Paris) Region	07:29-14:29 UTC

Figure 2 Backup Window

DB SNAPSHOT IN AWS RDS

The DB SNAPSHOT in RDS AWS, is used to create the backup of the entire DB instance and not the individual databases.

When a DB snapshot is created, it's required to identify which DB instance need to back up, and then give the DB snapshot a name so that the same be used to restore from it later. The amount of time taken to create a snapshot varies with the size of the databases.

Since the snapshot includes the entire storage volume, the size of files, such as temporary files, also affects the amount of time it takes to create the snapshot.

Manual snapshots aren't subject to the backup retention period like automated backups. Snapshots don't expire. DB snapshot can be created by using the AWS Management Console, the AWS CLI, or the RDS API.

PAPER REFERENCES

Mostly, I have used my knowledge and working experience to present the details in the paper, however, there were instance, wherein I do have taken the reference of some of the websites pertaining to the database such as the Oracle Base repositories (<https://oracle-base.com/>), and, AWS repositories, (<https://aws.amazon.com/>) which in itself is very huge for the database knowledge know-how, and have presented all the details in my own words after understanding the concept of the said area (**Backup and Recovery**) which refers to in this paper.. Also checked and verified the database blogs, and then used some information from the blogs to present the same in my own words for this paper.

Also I have taken the references from my own prepared notes during the day-to-day working in the database fields, as the challenges which comes into the DB area, and post, the resolution (errors) of the same it brings the solution which is required for the database to work efficiently.

LEARNINGS FROM THIS PAPER

This paper covers the aspects of database Backup and Recovery, in AWS cloud & Oracle Cloud Infrastructure (OCI), which is an important factor in the database management area. The work which has been done to prepare this paper has helped me to a large extent to enhanced my database knowledge not only to the previous versions of database but also to the latest version of **Oracle DB 19c**, which has many newly added database features (**CDB /PDB's**)which will definitely help me (**DBA**) and also the developers.

The implementations of all the features of database **Backup** and **Recovery** has helped me a lot for making the data availability all the time. It provides the **HA** (High-Availability).It is a very good working experience through this paper as it involved live implementations, including development and UAT.

For the preparation of this paper, I have gone into details of each and every components of database aspects pertaining to the overall DB Backups & Recovery using both **RMAN & DATAPUMP** utility.

IMPLEMENTATIONS

DATABASE RESTORE AND RECOVERY

For the Database Restore and Recovery to take place, the first and foremost important are the **RMAN (Recovery Manager)** backups which needs to be taken from the source DB server. The RMAN time taken for the backup and the time taken for the Restore and subsequent Recovery would vary a bit, as in the case of the RMAN backups, physical database files are backed up, whereas in the case of the Restore and Recovery, on the destination server, RMAN first Restores the full backup, and then after that the Incremental backups are applied and finally the archivelogs are applied, and, so, due to this the database backup and Restore & Recovery differs in time ratio.

All the available RMAN backups can be restored on to a separate server with same configurations as that of the primary server. RMAN backups are the physical database backups which actually stored the backup of the physical datafiles.The RMAN backups are the binary files, and which is used for the restoration and recovery purpose depending whether full database backups needs to be restored or recovered.

RMAN provides the facility of restoring the Incremental changes also known as Level 1 backups and the archivelogs backups, which can be used for the point in time recovery of the database.It also provides the option of taking the Archivelog backups between the from and to archivelog sequence number.Another option through which the functionality of RMAN used for the database restore and recovery is the **SCN (System Change Number)** backups.Here the SCN from the destination server be it Standby or DR is taken, and RMAN backups are taken from the source database server from that particular SCN to the point in time till the time the backup has started on the Source DB server.Then this SCN backup is used for the Restoration & Recovery at the destination database server.

Here the database backup is first restored by using the Full database backup. After the completion of the restoration process, the Incremental & Archivelog backups are applied to sync the database till the time of the available backups.

The very important for the database Restore and Recovery to work properly the below points needs to be executed in a sequence.

- Export the database id (DBID) in the command prompt in linux server.
- Start the database with the SPFILE.
- First the controlfile needs to be restored from the RMAN backup in the nomount state.

- Mount the database, wherein the database server is now in a position to access the controlfile and have at this point all the database details.
- Start the database restore by using the appropriate command.
- Once the database restoration is completed, then at this point of time, the remaining option is to execute the database Recovery command.
- Post completion of the database recovery, then start applying the archives to sync it with the Primary database server.
- Check the consistency of the database and recovery, open the database in the Read Only mode. Post completion of this step, the DB is in consistency with the primary database server, and further archives can be started applying, and it remains in sync with the primary database till the Physical standby or DR site is not opened in the read write mode.

Also RMAN Restore & Recovery provides the facility of Restoring & Recovering a particular tablespace or a datafile. The validity of the RMAN database backup can be verify by using the validate command in the command prompt of the Recovery Manager utility.

All the RMAN command which are necessary for the database Backup, Restore, Recovery can be called from shell script rather than doing it manually, and the same can scheduled at a specific time by using **CRON JOBS**, which is an in-built linux utility for task scheduling at the O/S level. This is similar to the task scheduler in Windows O/S.

Also another in-built utility which can be used is the **NOHUP (Linux O/S utility)**, wherein the command can be written in a file, and the sessions can be started by using the NOHUP command and the actual file name which has the backup and restore and recovery RMAN commands, the program will execute internally in the background automatically, without any sort of interruptions, and the logs of the actual backup, restore and recovery is made available and the same can be easily retrieved.

SCHEMA BACKUP AND RESTORE (DATAPUMP UTILITY)

The schema Backup & Restore in Oracle is a **logical** mechanism where in the **table data** and structure (**metadata**) is accomplished, in which a separate schema is created in the destination server by using the source schema backup. There are many options which are available with the **EXPDP/ IMPDP** which is a logical schema backup and recovery mechanism wherein the physical database backup files are not used (like RMAN), but instead this is the binary backup in the form of. DMP files, which only the **EXPDP & IMPDP** utility would be able to read and write into the SAN disks.

The EXPDP & IMPDP utilities very handy in terms of schema wise migration from one server to another server. It can also be used to migrate only the meta-data from one schema to another, or it can also be used for table data transfer only.

These famous utilities (EXPDP / IMPDP), are used very frequently in the Oracle database technology, and are extremely fast, the speed of which depends upon the total amount of data being migrated.

The best feature of EXPDP / IMPDP, is that it can be used for data migration between different Operating Systems like from **Windows (Source) to Linux (Destination)**, with very minimum changes that needs to be carried out at the destination server.

The two famous **DATAPUMP** utility which is being used very frequently and efficiently are **EXPDP & IMPDP**. The working of which are very similar, owing to the fact one is used for the **Export Backup (EXPDP)** and the other is used for the **Import Backup (IMPDP)**.

The **EXPDP** utility is used to take the full schema backup (data & meta-data both), only table data from a schema, only meta-data (structure) of the schema, etc.

Below is the complete process (Steps) of EXPDP in AWS RDS database.

- Get the schema details from the Source database server such as the Schema name, when the schema was created.
- The default tablespace (used for the data storage). It will show all the tablespace wherein the objects of the schema are residing. This is very important as far as the migration is concerned to another server as it would be used during mapping of the tablespace in the destination tablespace.
- The size of the schema, also plays a very significant role, as by knowing this value, it becomes clear, what would be the schema size at the destination server. This is very helpful in deciding the tablespace size at the destination server.
- Take the count of the total objects available in the schema before the start of the export backup by using EXPDP utility. The total objects counts in the source before the export backup must match with the total objects counts after the schema Import in the destination server by using the IMPDP utility.
- Find the database role that has been assigned to the source schema, which is very important, as the roles defines the access level privileges to the newly already exists schema **@Source** or newly created schema **@destination** database server.

- Finally execute the **EXPDP** command to take the full backup of a particular schema by using the EXPDP utility.

The **IMPDP** utility is used to restore the full schema backup (data & meta-data both), only table data from a schema, only meta-data (structure) of the schema, etc.

The very basic idea for IMPDP is very much similar to EXPDP i.e., in IMPDP, the same steps is followed like taking the schema details, etc. Here the exception is in the IMPDP command which used some other parameters which are not used in EXPDP

Below is the complete process (Steps) of IMPDP in AWS RDS database.

- Depending upon the Source schema size create the new schema along with the tablespace on the destination database server. The tablespace size should be more than the schema (source schema backup) size.
- The Source & Destination schema tablespace locations or mount points could be different, which depends on the NAS drives, which is in use.
- The default tablespace will be used for the data storage physically in the SAN disks. The default tablespace created in the destination server would be used for mapping from the source tablespace.
- The size of the schema, also plays a very significant role, as by knowing this value, it becomes clear, what would be the schema size, which would be imported in the destination schema. This is very helpful in deciding the tablespace size or the destination schema at the destination server.
- By using the predefined role, create the schema along with the default tablespace with the size which is more than the schema size of the source schema size.
- Perform the Import (**IMPDP**) operation on the destination schema by using the source schema backup dump file (.dmp).
- Once the Import is completed, then the source schema gets created as it's in the destination schema after the Import operation is completed. This operation is carried out through the IMPDP utility.
- Take the count of the total objects available in the schema after the import is completed through the IMPDP utility.
- The total objects counts in the source before the export backup must match with the total objects counts after the schema Import in the destination server by using the IMPDP utility.

DATABASE DR (DISASTER RECOVERY) SETUP IN ORACLE CLOUD

Full database backup along with the ArchiveLogs and Incremental backup by using the RMAN (Recovery Manager) utility is taken from the primary production server and is transferred on to the DR (Disaster Recovery) database server, which is located at a different geographical locations.. The full backup (Level 0) is first restored and after that the archiveLogs and the Incremental backups (Level 1) are applied to make the database in consistent mode. The database can be recovered to the point of time till which the ArchiveLogs backups are available. To test the database consistency, the DR DB can be opened in read only mode. To permanently make the DR as a primary (READ/WRITE), the DR database needs to be opened in READ/WRITE mode, after which the ArchiveLogs and Incremental backups cannot be applied any further, as the DR DB role has now been changed to Primary mode from Mount mode.

The details of the **Database DR** setup at **Oracle Cloud Infrastructure (OCI)** from **AWS Cloud** is provided below.

- Startup the database in nomount state. This is the mandatory step to be performed for any DR setup
- Start the RMAN utility. The Recovery Manager (RMAN) utility is the gem of an utility as provided in the Oracle databases, and which is the core and used mainly for the database backups and recovery, and is always the recommended practice in the Oracle database technology
- Set the exact configurations of the RMAN. Here there are a number of parameters that can be setup for the RMAN to function upon.
- Allocate the RMAN channel. Here channel needs to be defined which the RMAN will use internally to process the backup and recovery at the DR site @OCI.
- Restore controlfile & SPFILE from the L0 backups. The full backup or L0 backup has the controlfile (consists of database details) and SPFILE (parameters to be used by the DB during database mount), which needs to be restored into the DR site @OCI, at this point

- Now mount the database @OCI, by using the SPFILE. Once the database is mounted it means that the DB has accessed to the controlfile , but not the physical datafiles.
- Catalog the backup files with the DR database. This is the final step before starting the database restore and recovery. At this step, the RMAN confirms to Oracle server the details of the backup files which will be first restored (creating physical files in the disks) and then perform recovery (applying the transactional changes) on to the DR site @OCI.
- Execute the database restore, switch data files and database recovery commands which will executed in the background. Restore will copy all the physical datafiles, and datafiles switch will rename the datafiles names as provided, which will get associated to the particular tablespaces.
- Now to check the database consistency between the primary database server and the DR database server, need to open the DR database server in read only mode, and the steps of which are given below.
- Now at this point the DR database is in consistent state with the primary database server. Further syncing of the DB is done by using the Archivelog backups, by taking the DR again into the Mount mode as shown below.

Very important, in the **OCI (Oracle Cloud Infrastructure)**, for the **DR** restoration and recovery, the database backup is placed on to the OCI object storage area, and is then mounted into the database by using the **RMAN** utility in the form of tape backups.

The database backups which have been transferred to OCI from AWS RDS now needs to be loaded in the DR site @OCI.

First and foremost is to link all the backups with the tape library, and after the completion of this phase, the database restoration can be started.

Now the database restoration, can be started and after completion of the same, the next step is to create the data files names @DR site. Once these two steps are finished, then at this point of time all the physical database backup files which were @AWS, are now got created successfully @OCI.

Finally, the database recovery (applying changes) can be started.

After completion of the database recovery, the DR is now in sync or in consistent mode with the Primary DB.

To use the DR DB in the READ/WRITE mode, the DR DB should be opened, and then after which it can be used fully as a Primary database. At this point the DR role has shifted to Primary from Mount, and the DR DB is now at this point is fully ready to be used as Primary production database server.

Once the DR database has been opened, then in this case no Archivelogs or incremental backups can be applied on to the DR, due to the fact that the DR DB role has changed to Primary from Mount. The Archivelogs and Incremental backups can be applied only during the time when the DB is in Mount mode, in Primary mode the DB will not accept any applying of Archivelogs or the incremental backups.

DATABASE DR Setup at Oracle Cloud Infrastructure (OCI) from AWS RDS cloud, in a graphical manner is shown below.



Figure 3 DR Setup from AWS RDS to OCI

CONCLUSIONS AND RECOMMENDATIONS

This paper was immensely helpful in implementing the various features of Backup and Recovery with regards to the management of the **large chain of hotels & ERP Database world-wide**. The widely regarded and mandatory aspects of the database management in AWS cloud pertains to the Backup and Recovery of the DB server. Owing to the growing demand of the database size and taking into account the core areas of database it's very much required to migrate to **19c**. By migrating to **19c** the company will get the benefit of **Oracle's ATP (Autonomous Transaction Processing)** feature including **Container databases (CDB/PDB's)** and which provides the HA (High availability). This is highly appreciated and used globally. There is vast features

in **Oracle 19c**, and by utilizing which will increase the overall performance, backup and recovery & security of the database at large.

As far as recommendations is concerned, I would like to take this opportunity to implement new features through the use of latest version of **Oracle 19c in AWS cloud / OCI (Oracle Cloud Infrastructure)**, which also supports **Autonomous Transaction Processing (ATP)** feature which reduces the manual intervention of the DBA. Also would like to implement other available feature in **Oracle AWS cloud / OCI Container database (CDB) and pluggable databases (PDB's)**.

I have been assigned the task of setting up the **DR at the OCI (Oracle Cloud Infrastructure)** container database environment, where in the primary database is at AWS cloud. Also the **Payroll DB** would also be migrated to **Oracle 19c OCI (Oracle Cloud Infrastructure)**, which will eventually eradicate the issue of resource management manually, and will obviously enhance the DB performance.

Oracle database 19c has new features as compared to **Oracle 12c**, and depending upon the current industry trend it's the right time to migrate to **Oracle Cloud Infrastructure (OCI)**, which is a great player in the database cloud environment management.

BIBLIOGRAPHY

- [1] Badger, L., Grance, T., Robert, P. and Voas, J. *Computer Security. DRAFT Cloud Computing Synopsis and Recommendations, NIST Special publication 800-146.*
- [2] Barbara, J., Jo, A., Cynthia, S. and Adam, T. (2009). *Collaboration Using Cloud Computing and Traditional Systems. Western Carolina University, Vol. 10, No. 2.*
- [3] Dykstra, J. and Alan, T. (2012). *Acquiring Evidence form Infrastructure-as-a- Service Cloud Computing: Exploring The Evaluation Tools, Trust and Techniques. Digital Investigation S90-S98.*
- [4] Hofer, C., Karagiannis, G. (2011). *Cloud Computing Services: Taxonomy and Comparison. Received: 1 February 2011 / Accepted: 18 May 2011 / Published online: 19 June 2011 © The Author(s) 2011. Retrieved from <http://link.springer.com/article/10.1007%2Fs13174-011-0027-x>*
- [5] Lazewski, V., Javier, D., Fugang, W. and Geoffrey, C. (2012). *Comparison of Multiple Cloud Frame Works. Retrieved from http://www.academia.edu/2929921/Qualitative_Comparison_of_Multiple_Cloud_Frameworks.*
- [6] R. Buyya, C. Vecchiola, and S. T. Selvi, "Mastering cloud computing: foundations and applications programming," 2013.
- [7] P. K. Paul, K. Kumar, and D. Chaterjee, "Cloud computing emphasizing emerging possibilities to the entire Information Infrastructure," *Trends in Information Management (TRIM)*, vol. 9, no. 2, 2014.
- [8] *Worldwide Cloud Computing Market Shares, Strategies, and Forecasts, 2009-2015, Retrieved from: http://www.researchandmarkets.com/reports/1057978/worldwide_cloud_computing_market_shares, December, 2014.*