Strategic Database Cloud Service Decision: Oracle vs. AWS

by, Marc Staimer

14 April 2021

Databases are foundational infrastructure software. They are at the heart of many applications, ranging from e-commerce, data warehouses, big data analytics, document management, key value stores, to AI, machine learning, and deep machine learning.

Databases are complicated. They require expert knowledge to implement, configure properly, operate, and especially manage effectively. What's especially difficult is setting up a highly available, mission/business-critical database that minimizes or eliminates downtime—a non-trivial and essential task.

Database Cloud Services reduce the management effort required for database environments by moving the effort of automation and services to the Database Cloud Platform provider and reducing the effort required of high-cost experts such as database administrators (DBAs), data engineers, data scientists, and data architects.

A CIO's important strategic decision is choosing the best Database Cloud Service for their particular business needs. That choice determines the cost of the Database Cloud infrastructure, the functionality of database applications, the ability to automate business processes, the performance, elasticity, availability and security of core business-critical applications, and the speed of digital transformation.

The scope of this research is to evaluate the choice of migrating existing Oracle Database applications to a Database Cloud Service from the two largest Database Cloud Providers, AWS and Oracle.

Premises

Organizations with Enterprise or mission/business-critical environments built on Oracle Databases desiring a move to a Database Coud Service are likely to consider Oracle and AWS since these two giants both offer Oracle Database Cloud Services.

This research concludes that of the two, only Oracle provides a Tier-1[1] Database Cloud Service. A cursory examination of these two Oracle Database Cloud Service providers makes it crystal clear that the Oracle Database Cloud Services are superior to AWS Oracle Database Cloud Services for Enterprise or mission/business-critical environments.

CIOs desiring to move from Oracle Database on-premises to an Oracle Database Cloud Service will also find the Oracle Database Cloud Services considerably more capable. Oracle delivers more functionality, database applications, ability to automate business processes, performance, elasticity, availability, security, support for mission/business-critical applications, digital transformation speed, lower costs, and accelerated revenues. This is true for both the complete Database Cloud Service that includes the database license and the “bring your own license” or BYOL service.

This research establishes that the migration speed to the Oracle Database Cloud Service is appreciably faster at much lower cost than any combination of AWS Database Cloud Services or BYOL Oracle Database to the AWS cloud platform.

One other research conclusion is the flexibility of deploying the same Database Cloud Service on-premises or in the cloud. Oracle’s unique deployment flexibility simplifies an organization’s ability to meet national, international, legal and operational placement requirements for data and computing resources with an on-premises option that is architecturally equivalent to Oracle Cloud Infrastructure (OCI).

[1] Wikibon has determined the Tier-1 databases are IBM DB2, Microsoft SQL Server, and Oracle Enterprise DB. SAP with SAP Hana is attempting to reach Tier-1 status, but Wikibon believes it is unlikely to achieve this status for OLTP applications. Wikibon assesses that neither AWS nor Google have Tier-1 database offerings at this time. Wikibon has previously assessed that AWS will not achieve Tier-1 capabilities by using open-source databases and will need major long-term investment requiring a decade or more to achieve current Tier-1 status. This assumes the goal posts do not continually move, which is unlikely meaning it will take even more time. In the meantime, both AWS and Google will need to partner with Tier-1 database providers to provide a Tier-1 Database Cloud Service.

Executive Summary

This research is designed to help CIOs to make a decision on which Database Cloud Service to adopt for Oracle Database. It does so by taking an in-depth look at Enterprise Database Cloud Service market requirements. To be an Enterprise-class Database Cloud Service includes being mission/business-critical; no functionality loss versus on-premises; non-disruptive elastic scalability; non-disruptive operations; high performance; simplified data extraction, transformation and loading between database models; extensive automation; end-to-end security; very high availability; ACID compliance, and low total cost of ownership.

This document goes into depth on each of those requirements. Then it examines each of the Database Cloud Service offerings from AWS, details exactly what they deliver and reveals why they do not or cannot meet Enterprise Database Cloud Service requirements. Then it examines Oracle Exadata Cloud Service X8M, detailing its capabilities and how it not only meets but exceeds every aspect of all Enterprise mission/business-critical Database Cloud Service requirements.

The business conclusions will become obvious that Oracle Exadata Cloud Service X8M is an Enterprise-class Oracle Database Cloud Service. The AWS Database Cloud Services, including the Oracle Database and their much-hyped flagship services Aurora and Redshift, are not. These AWS Database Cloud Services deliver less performance, flexibility, availability, elasticity, scalability, functionality, security, data protection, and have no multi-model and multi-workload support. Although they deliver less than acceptable mission/business-critical capabilities, they do so at a considerably higher total cost of ownership. The AWS Oracle Database Cloud Services option within RDS is adequate for non-Enterprise-class requirements, specifically those that are not business/mission-critical.

The data makes it clear that Oracle Exadata Cloud Service X8M is superior to the AWS Database Cloud Service [2] in all measurable metrics.

[2] All technical specifications, capabilities, features, functions, and services described in this document are based on published materials on AWS and Oracle websites.

Table of Contents

Enterprise IT Database Cloud Service Requirements

Enterprise IT Database Cloud Service requirements start with mission/business-critical applications and workloads. Mission/business-critical applications and workloads cannot go down because of the significant cost to the organization. These applications and workloads nearly always depend on Oracle Databases. Moving them to a Database Cloud Service requires that service also be mission/business-critical capable. There are several baseline requirements common to mission/business-critical Database Cloud Services that not all cloud providers necessarily supply. They typically include:

- The database upon which the service is based must support **mission/business-critical applications**. This is obvious to the lay observer. This means the database supporting those applications must also be mission/business-critical. If the database is not mission/business-critical then logical reasoning makes it quite clear that any Database Cloud Service based on that database by definition is also not mission/business-critical, and nor are any of the applications depending on that service.
- **No functionality loss** compared to on-premises database capabilities. Enterprise customers should not have to compromise on the capabilities they already know and use. They typically have built their applications leveraging the capabilities of on-premises mission/business-critical databases. Taking away currently utilized functionality and capabilities and requiring significant modifications to those mission/business-critical applications and workloads is both expensive and time-consuming.
- **Non-disruptive elastic scaling** of performance and capacity in a single database and/or multiple databases. In order to meet availability and cost objectives, the platform must adjust dynamically to changes in workload performance and capacity requirements.
- **Non-disruptive operations**. Disruptive operations require downtime, scheduling, after hours or weekend work, skilled administrators, and time—all of which are in short supply. Therefore, minimizing or eliminating disruptive upgrades, patching, and on-demand elastic scaling of virtual CPUs and storage is a must.
- **Higher than or at least equal performance** as on-premises. A reduction in performance is generally unacceptable. It is extremely rare that an application or workload demand the database to slow down.
- **Simplified data extraction, transformation, and loading (ETL)** between diverse database or data models. Different applications and workloads may require different database models such as transaction processing versus analytics. The data required to be analyzed within a data warehouse is not created in that data warehouse. It typically comes from a relational transactional database. Moving that data automatically, error-free, in a timely manner, and without disruption is an absolute must for Enterprises to move their databases to the cloud. Not having to perform ETL at all is even better.
- **Highly automated with intuitive operational simplicity.** The goal is to exceed on-premises database operations and management by eliminating much of the administrative tasks and moving them to the cloud service provider or the database itself through AI/ML-driven automation.
- **Effective end-to-end security**. In these days of cyberattacks and ransomware, security is clearly top of mind to CIOs. End-to-end security, strong encryption, fool-proof authentication, and reduced surface area exposure all play a part in protecting data in a Database Cloud Service.
- **Exceptional availability, backup, DR, and seamless business continuity.** A key Enterprise rationale in moving mission/business-critical applications to the cloud is to improve availability. Database Cloud Services whose availability SLAs contain a laundry list of exceptions are not mission/business-critical.
- **ACID** (atomicity, consistency, isolation, durability). ACID guarantees data validity despite errors, system failures, power failures, and more. Many of the NoSQL Database Cloud Services claim eventual consistency is good enough. For some applications and workloads, that's true. For mission/business-critical applications, it is not.
- **Equal or preferably lower total cost of ownership (TCO).** Not just a conversion from capital to operating expenditures. But rather lower TCO over time compared to on-premises.
- **Avoiding complex mission/business-critical application conversion.** Changing the database to a mission/business-critical application is often perilous. First, the database data has to be extracted, replicated to the Database Cloud Service, then transformed, and loaded. This ETL process is frequently painstaking, tedious, subject to errors, and time consuming generally taking several months to years. And that's just the database. Those mission/business critical applications using the database are making SQL calls that will likely have to be changed to work with the new Database Cloud Service if it's

not using the same database. This requires combing through those applications and making adjustments to their programming. Detailed documentation showing what needs to change in the application commonly does not exist or is poorly documented. Figuring out what needs to be changed is a major time sink often boiling down to fixing the application fails or errors occur.

Prioritization, weighting, and ranking these requirements will likely vary by application, workload, DBA, and IT organization. Keep in mind that not all of these Database Cloud Service capabilities are required by every mission/business-application.

This research examines in depth the multiple AWS Oracle Database Cloud Services mission/business-critical capabilities and compares them with Oracle Exadata Cloud Service X8M. It concludes that CIOs will achieve a migration from an in-house Oracle Exadata platform or on-premises x86 server deployment to an Oracle Database Cloud Service considerably faster and at lower cost than any AWS Database Cloud alternatives. Oracle Cloud Infrastructure with Exadata X8M also provides higher availability and reliability.

Database Cloud Services

AWS Database Cloud Services

AWS has **16** distinct Database Cloud Services based on different databases and data models. These databases started out as open source, but Amazon has modified them with proprietary interfaces which means that once an application is developed on these databases it only runs on AWS. There are **7** different cloud relational database services (RDS). They are commonly utilized for online transaction processing (OLTP), built for fast access using indexes and normalized data models. Analytics and data warehouses also tend to use relational databases. They're commonly constructed on a dimensional model like Star Schemas and use parallelism to process data rather than indexes.

AWS Aurora is their premier relational Database Cloud Service based on an Amazon-modified open-source PostgreSQL database. AWS Redshift is their primary data warehouse cloud service also based on a modified PostgreSQL database. Most of the other RDS Database Cloud Services are based on popular or open-source databases.

1. **Oracle Database Standard Edition/Standard Edition 2** (SE/SE2).
2. **Microsoft SQL Server**.
3. Open-source **PostgreSQL**.
4. Open-source **MySQL**.
5. Open source **MariaDB** (MySQL fork).
6. Amazon's own variation of PostgreSQL or MySQL, called **Aurora,** is optimized for transaction processing.
7. Amazon's other PostgreSQL database, called **Redshift,** is optimized for data warehousing with massively parallel processing technology from Actian (previously ParAccel).

AWS has another SQL Database Cloud Service called **Athena** that's relational in the sense that it specifically enables schema and SQL queries on AWS S3 data.

AWS has an additional **6** other Database Cloud Services and **2** in-memory Database Cloud Services. The NoSQL Database Cloud Services include:

1. **DynamoDB** for key value databases.
2. **DocumentDB** JSON or document databases.

Both of these Database Cloud Services are based on document databases which store data in a de-normalized structure. Most document databases utilize JSON or something like it, to organize the data. Document databases are sometimes used in lieu of relational databases for OLTP applications because they can provide fast access to relatively simple data structures.

3. **Keyspaces** for wide column databases.
4. **Neptune** for graph databases. Graph databases are generally not relational. They are visual tools that are excellent for analyzing connections between data points such as social networks or tracing contacts between people.

5. **Timestream** for time series databases. Time Series databases are also relational. They're built to ingest data specifically in a time-oriented format, then provide time-oriented analysis on that data.
6. **QLDB** for blockchain ledger databases.

The in-memory Database Cloud Services include:

1. **ElastiCache for Memcached** in-memory database.
2. **ElastiCache for Redis** in-memory database. Most in-memory databases are relational. They're designed to enable fast ingest and analysis of data by storing the data in memory.

AWS commonly encourages Enterprise customers to move away from their on-premises Oracle Database to RDS Aurora. They point to their own migration away from the Oracle Database to Aurora. This is almost always extremely problematic for Enterprise customers and not a realistic alternative based on conversion time and cost.

Wikibon research has found large mission/business-critical application conversion to always be complicated. That is not an overstatement. If anything, it's understated. The effort is mostly calculated in years, minimally 2-3 years. Note that it took AWS five years and hundreds of employees to migrate many of its own mission/business-critical applications from Oracle to Aurora and they're still not fully off of the Oracle Database.

Applications must be frozen when their database is being converted. According to the research, the database applications lose about 12% of their value per year. Freezing those applications for 2-3 years means 24%-36% lost value. In addition, staff must be moved over to work on the conversion to get it done as quickly as possible. More value is lost as no new applications can be developed while the conversion to the Database Cloud Service is occurring. There are very few Enterprises willing to invest five years, with application value declines of 12% per year, and hundreds of costly developer hours for illusionary nominal gains at best, and significant losses at worst.

These are highly consequential costs that can easily run into many millions of dollars in lost value depending on the size of the organization. Those costs come from reallocation of and much reduced productivity of application developers, database administrators, end-users, and major time lost in getting applications to market. It generally takes at least 1-2 years to add the functionality that was put on hold into the converted programs.

Conclusion - Conversion loss is many times the IT budget. No savings are likely to be gained. Many large-scale conversions are known to lead to a change in CIO. Conversions take the IT organization's focus off creating value for the organization in a futile attempt to save money. It is not uncommon for a conversion to cause major business difficulties.

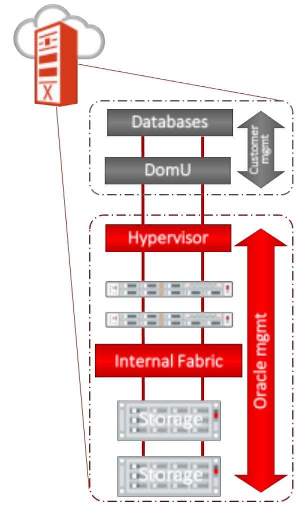
There are times when applications must be converted because the underlying database is no longer supported. That is not an issue with Oracle. Oracle is the leading tier-1 database and extremely unlikely to stop support anytime in the next 20 years.

Bottom line - application and database conversions should be avoided like the plague.

Oracle Exadata Cloud Service X8M

Oracle Exadata Cloud Service X8M is specifically engineered to run Enterprise-scale, mission/business-critical Oracle Databases. It supports OLTP, data warehouse, real-time analytics, AI/machine learning, and mixed database workloads at any scale individually or concurrently. The Exadata Cloud Service X8M is the state-of-the-art, fastest, lowest latency Oracle Database Cloud Service available. No other Database Cloud Service is remotely close.

Exadata Cloud Service X8M is built on Oracle Database Enterprise Edition (EE) and the Oracle Exadata X8M architecture. Oracle Database EE is a converged database that handles most database and data types including relational OLTP, data warehouse, in-memory, XML/object, JSON/document, key value, blockchain, time series, and graph/spatial. Exadata X8M is Oracle's highest-performance engineered system. It is co-engineered with the Oracle Database at the source code level to provide unparalleled synergistic results. It



delivers over 60 unique automation and performance enhancements which are only available when Oracle Database is running on Exadata. Customers can either subscribe to the complete Oracle Exadata Cloud Service X8M that comes with the Oracle Database EE plus all options, or they can bring their own license and run it on Oracle Cloud Infrastructure (OCI)-based Exadata X8M hardware. Oracle provides the same cloud services for both.

Oracle Exadata Cloud Service X8M is specifically designed to provide customers with unprecedented performance, functionality, automation, scalability, and security while enabling them to maintain control of their databases. All of this while Oracle manages the hardware, storage, network infrastructure, patches, hot fixes, and upgrades, as shown in the accompanying diagram. Note that Oracle does not do the same for user applications running in the guest VMs.

Oracle Exadata Cloud Service X8M runs on dedicated hardware. This eliminates all potential "noisy neighbors" issues. The most common Oracle Exadata Cloud Service X8M use case is the consolidation of dozens to hundreds of Oracle Databases. Customers can share or wall off each database and its data completely from one another via the Oracle Database container database (CDB) and pluggable databases (PDB), enabling comprehensively secure multi-tenancy.

Oracle also provides non-Exadata **Oracle Database Cloud Services** on VMs or bare metal servers on Oracle Cloud Infrastructure (OCI). These services deliver lower performance than Oracle Exadata Cloud Service X8M but have lower fees per Oracle CPU (OCPU). It is important to note that depending on the database job, these lower fee Oracle Database Cloud Services can actually have a higher TCO. More on that in the TCO section.

The **Oracle Autonomous Database** is Oracle's premier Database Cloud Service. In the Database Cloud Services market, there is simply nothing currently remotely comparable to Oracle's machine learning-powered, self-driving Autonomous Database (ADB) is optimized for three use cases: Autonomous Transaction Processing (ATP), Autonomous Data Warehouse (ADW), and Autonomous JSON Database (AJD). When utilizing the Oracle Autonomous Database, customers merely have to manage the data and schemas. The infrastructure is managed by Oracle while the database is self-driving. When utilizing the Exadata Cloud Service X8M with the Oracle Database and not the Autonomous Database, the database is not self-driving.

However, with the level of automation provided by Autonomous Database, some database administrator (DBA) professionals may be generally uncomfortable with turning over database control to a combination of AI, algorithms, and historic best practices in Autonomous Database. They're reluctant to trust their jobs and careers to processes outside of their control since they are ultimately responsible for the well-being of the databases. Responsibility and experience motivates many of them to keep control over several administrative tasks. It's no different from a driver being wary of a completely autonomous car. That doesn't mean they do not want automation. It means that they want more control over that automation while also spending a little less.

That's where the Oracle Exadata Cloud Service X8M comes into play. It's the same Oracle Database Enterprise Edition with all options running on the same peerless Exadata X8M engineered hardware. The performance is the same. The key difference between Exadata Cloud Service X8M running Oracle Database EE and Oracle Autonomous Database is who controls the automation. With Oracle Autonomous Database, the Oracle algorithms control the automation. With Oracle Database EE running in the Exadata Cloud Service X8M, DBAs control the automation. This is analogous to autonomous vehicles. In the fully autonomous vehicle, the driver does not drive or control the vehicle. The vehicle is controlled and driven by the built-in machine intelligence. In the semi-autonomous driving vehicle, the driver drives and controls the vehicle. Oracle offers organizations the choice and flexibility of both.

Mission/Business-Critical Database Cloud Service

It all starts by understanding what Enterprise IT organizations mean by mission/business-critical Database Cloud Services. The service must support mission/business-critical applications and workloads necessary for the success of the organization, a specific operation, or the entire business. The loss of a mission/business-critical application or workload for any reason is rapid, substantial, and potentially catastrophic. When mission/business-critical systems fail, are interrupted, or suffer an outage, the results are highly consequential. Mission/business-critical applications and workloads that have outages are likely to cause the organization to see a loss of customer satisfaction, customers, revenue, profits, and productivity. In addition, their hard-earned reputation can suffer, requiring years to rebuild. Therefore, to be considered mission/business-critical the Database Cloud Service itself must be highly available, non-disruptive or at least have minimally disruptive operations, no single points of failure, with built-in backups against any outage, disaster recovery, and preferably business continuity – to keep running during a disaster.

AWS Database Cloud Services

AWS has most of the Database Cloud Services an Enterprise might require. The question is whether or not any of these Database Cloud Services can be considered mission/business-critical.

The open-source databases which are the basis of AWS' primary Database Cloud Services, are not considered by many industry experts as mission/business-critical databases. They generally lack the built-in capabilities for performance, availability, scalability, business continuity, automation, and more.

However, AWS RDS also includes the Oracle Database, which has been the very definition of a mission/business-critical database for decades. However, AWS RDS does not include Oracle Database Enterprise Edition (EE). It only offers the Oracle Database Standard Edition (SE2) and customers are limited to a mere 10 instances of the Oracle Database SE or SE2. For a customer to run the Oracle Database EE in the AWS cloud requires them to bring their own license (BYOL) and run it in AWS EC2 instances. There are significant issues with running BYOL Oracle Database EE in the AWS cloud. The biggest is performance. AWS limits IOPS performance to a maximum of 80,000 per EC2 instance, 256,000 IOPS per elastic block storage (EBS) volume, and 4 GBps throughput within io2 Block Express. That's a low IOPS number for Oracle Database EE. If the applications utilizing the Oracle Database are IOPS-intensive, customers will find the performance unacceptable and definitely NOT mission/business-critical.

As previously mentioned above, the AWS RDS Database Cloud Service only supports Oracle Database SE2, a lower-end version of the Oracle Database EE. It is designed to compete with lower-end databases such as Microsoft SQL Server, PostgreSQL, MySQL, and MariaDB which are available on AWS RDS. Oracle Database SE2 delivers more functionality and performance at equivalent costs. But Oracle Database SE2 is not Enterprise-class. It lacks much of the Extended Edition functionality that makes an Oracle Database mission/business critical. In addition, Oracle Database SE2 running on AWS does not include Real Application Clusters (RAC) for unmatched availability and performance, Data Guard for seamless business continuity, parallel query, and more than 60 unique capabilities such as automated tuning, automated online non-disruptive patching/upgrades/hot fixes, automatic indexing, and automated root cause analysis. Running Oracle Database SE2 in AWS RDS generic hardware also negates much of the performance and scalability enhancements only available on Oracle Exadata X8M. The differences between SE2 and EE are massive in performance, scalability, availability, data protection, security, operations, business continuity, and much more. It's not close by any measure.

Referring to the current AWS Database Cloud Services as Enterprise-class or mission/business-critical is more marketecture than it is architecture at this time. It is possible to add some mission/business-critical capabilities through hardware infrastructure such as the AWS Advanced Query Accelerator (AQUA). AQUA is AWS' new distributed hardware accelerated cache that brings compute to the storage layer for Amazon Redshift (data warehouse). It noticeably accelerates Redshift query performance. But even this hardware cache acceleration is not co-engineered with Redshift. It is an attempt by AWS to use hardware fix to their software performance problem. It still does not match Exadata Cloud Service X8M performance, but it is closer[3]. It also does not increase reliability, availability, or business continuity. It actually makes Redshift's data protection more difficult. AWS' data protection is severely limited and optional to EBS snapshot replication to S3, multiple availability zones, and HA/failover. Fail over is not instantaneous and requires a disruptive outage – i.e., downtime.

AWS also enables Enterprise customers to bring their own Oracle Database licenses and run them in EC2 instances in the AWS public cloud. There are serious caveats to doing this. Many of the Enterprise Edition's most popular capabilities are not allowed due to AWS restrictions. Customers cannot utilize "Real Application Clusters aka RAC", nor consolidate multiple databases, or even implement Data Guard. AWS limits an Oracle Database to a single instance. Scaling the instance is promoted as a single button to push, but it is both disruptive (planned downtime) and very coarse grained, incremented by 2x the size of the current infrastructure instance. That means if the current instance is 32 vCPUs, the next increase is 2x or 64 vCPUs even if only 33 are required. The customer pays for 64 vCPUs even though they cannot use all of them.

In addition, the AWS cloud infrastructure costs for Oracle Databases are much greater than users anticipate, frequently 3x higher than Oracle Exadata Cloud Service X8M. This will be covered in more detail in the TCO section.

Oracle Exadata Cloud Service X8M

Oracle Exadata Cloud Service X8M is mission/business-critical by any definition. Patching, hot fixes, upgrades, troubleshooting/root cause analysis are performed online, non-disruptively, without outages. Indexing, tuning, and root cause analysis are all automated, requiring no admin intervention.

Business continuity (BC) and disaster recovery (DR) are also automated, and one touch leverages the highly-regarded, production-proven, industry-hardened Oracle Real Application Clusters (RAC) and Oracle Data Guard. Both BC and DR are non-disruptive or minimally disruptive in the case of a regional disaster.

Exadata Cloud Service X8M is based on Oracle Database EE, a converged database, and Oracle Exadata X8M. Oracle Database EE embodies the definition of a mission/business-critical database. Exadata X8M is co-engineered with Oracle Database EE to make sure the hardware is just as mission/business-critical as the database. Oracle has sold tens of thousands of Exadata systems for mission/business-critical deployments across all industries since it was introduced in 2008. There is no more resilient or performant Oracle Database platform anywhere. Exadata X8M's no single point of failure architecture and performance numbers make that clear. More on that in the performance section.

This combination of resiliency and high-performance capabilities that are highly desirable on-premises carry over to Exadata Cloud Service X8M, plus the true transparent non-disruptive cloud elasticity and fine-grained pay-as-you-use, per second billing fee structure makes it that much more attractive for customers looking to take Oracle Database to the cloud.

Oracle rewards customers that bring their Oracle BYOL to Oracle Exadata Cloud Service X8M by providing additional database functionality at no cost to the customer. This software includes Exadata-specific software, Transparent Data Encryption (TDE), Oracle Data Safe, plus Oracle Enterprise Manager's diagnostics, and tuning packs.

Conclusion: Of the Database Clouds available from AWS and Oracle, Wikibon classifies only the Oracle Database as a Tier-1 database platform. Both the AWS RDS Oracle Database Cloud Service and the AWS Oracle BYOL, degrades and or removes key mission/business-critical capabilities. Oracle Exadata Cloud Service X8M does not. It delivers all of the performance and high availability capabilities that make for a mission/business-critical Oracle Database Cloud Service.

No Database Functionality Loss

Moving from an on-premises database to a Database Cloud Service can be incredibly complicated if the Database Cloud Service delivers reduced functionality. This can be particularly problematic when the customer's database-dependent applications are utilizing the missing

[3] AWS comparisons on their website to Oracle are misleading because they're based on the Oracle Database SE2 running in AWS RDS on a VM. Queries to the Oracle Databases running in the Exadata Cloud Service X8M are up to orders of magnitude faster than queries to the Oracle Databases running in AWS RDS.

functionality. In this situation, the customer has to jury-rig or rewrite their applications to work around the dependency. This is not trivial. It takes extensive resources, quality assurance, testing, patching, updating, and most of all, time. Most organizations prefer to avoid that immensely costly situation.

AWS Database Cloud Services

For AWS open-source Database Cloud Services, AWS Microsoft SQL Server Database Cloud Service, and the AWS NoSQL Database Cloud Services, there is no apparent loss of functionality. However, for the AWS Oracle Database Cloud Service, as previously discussed, customers who have been using Oracle Database EE on-premises will definitively lose functionality. Even if they bring their Oracle Database EE license to AWS in a BYOL, they will lose a significant amount of performance. And if they were running the Oracle Database on an Exadata, they will lose an order of magnitude amount of performance plus more than 60 unique capabilities only available on Exadata platforms. Functions such as automatic indexing, automatic tuning, automatic root cause analysis, automatic in-memory, and much more.

There is generally no loss of database functionality in AWS Database Cloud Services with the glaring exception of their AWS RDS Oracle Database SE/SE2. However, AWS performance and capacity scalability limits should be construed as loss of functionality for several of their Database Cloud Services including all RDS Database Cloud Services.

And, as noted earlier, even AWS Oracle BYOL loses popular important functionality (no consolidation, RAC, Data Guard, etc.) when transitioning to AWS EC2 instances.

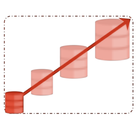
Oracle Exadata Cloud Service X8M

As previously discussed, Exadata Cloud Service X8M is based on the Oracle Database EE and Exadata X8M. As a converged database, Oracle Database EE includes every significant database model and data type built-in, including OLTP, OLAP, data warehouse, advanced analytics, AI/ machine learning, object/XML, key value, document/JSON, time series, graph/spatial, blockchain, and IoT. In contrast to AWS' multiple, special purpose Database Cloud Services, Oracle's are all part of the same converged Oracle Database in Exadata Cloud Service X8M. One service, period. The customer can share the data between databases and data types without ETL or isolate it in PDBs. It's their choice.

In addition, Exadata Cloud Service X8M includes all of those more than 60 unique capabilities that are only available when the Oracle Database runs on Exadata. Not only is there no loss of functionality when compared to on-premises, there is additional functionality not possible on-premises, the most important of which is automatic transparent elasticity. If the Oracle Database requires more OCPUs[4] than contracted (up to 3x), it gets them immediately without downtime.

Simply put, when customers migrate Oracle Databases from Exadata on-premises to Exadata Cloud Service X8M there is zero loss in Oracle Database functionality. In fact, customers gain cloud on-demand elasticity capabilities not available on traditional on-premises deployments.

Conclusion: AWS Database Cloud Services are a step back in functionality when customers migrate from Oracle Database on-premises. In contrast, Oracle Databases on Oracle Exadata Cloud Service X8M lose no functionality because it is architecturally equivalent to Exadata X8M on-premises.



Non-disruptive Elastic Scalability

Seems intuitive. One of the key benefits of the cloud is unlimited elasticity. Except there are differences in how the public cloud vendors interpret the meaning of the word "elasticity." The biggest qualifiers are whether or not it's disruptive and how granular the elastic increments are. Disruptive elasticity is familiar to on-premises customers. Disruptions often occur when an application needs more compute services, memory, or storage. Disruptions have to be scheduled so they do not affect or minimally impact business operations. That scheduling generally occurs on a weekend in the quarter where all the disruptive processes are scheduled at the same time. Most IT organizations want to eliminate or minimize disruptive processes. It's a big reason for moving to the cloud. The other issue is the level of granular elasticity. Is the cloud provider just provisioning traditional physical servers or actually delivering a true pay-for-what-you-use cloud service?

[4] Oracle OCPU is a full core.

AWS Database Cloud Services

The AWS Database Cloud Service databases are elastic, but not automatic nor non-disruptive. AWS Database Cloud Services run on VMs in specifically configured hardware shapes that Amazon calls instances. These instance shapes include a specific number of virtual CPUs (vCPU[5]) and memory. Moving from one instance shape to another is technically elastic and on-demand, but they're neither dynamic nor non-disruptive. It's a few simple clicks, but disruptive requiring a planned outage and moving data from one shape, read physical server, to another. Meanwhile, data is not available nor protected during this process. The same is true for patches and upgrades. AWS controls the process while requiring customers to plan downtime. To add insult to injury, these disruptions do not count toward the AWS availability SLA guarantees since it is a planned outage.

And, as noted previously, shape increases are not automatic and very coarse grain. Each shape is 2x the previous shape. If the current EC2 shape is 16 vCPUs and running out gas and a couple more vCPUs are required, that's not an option. The customer must move to the next EC2 shape size of 32 vCPUs. This coarse granularity means the customer is now paying for 30 additional vCPUs they are not using.

AWS Database Cloud Service's performance is generally constrained by their generic hardware, with the aforementioned AQUA query acceleration for RedShift. There are hard limits in IOPS and throughput to any specific EC2 instance and from elastic block storage (EBS). The EBS limitations are irrespective of the drive type—SSD or HDD. The maximums are a bit in flux ranging to approximately 64K IOPS for an EC2 and approximately 80K IOPS for EBS, and 256K IOPS for EBS io2 Block Express. That's not going to cut it when an application or multiple applications need to scale database performance as most Enterprise mission/business applications consistently do. This highly constrained performance is significantly lower than what can be achieved on-premises and orders of magnitude below the levels that Exadata Cloud Service X8M can reach. This is a difficult issue for AWS RDS Database Cloud Services.

But Amazon claims their RDS databases scale up and out. AWS RDS has limited scale-up. The Database Clouds scale in vCPUs and memory chunks called shape instances. There are more than 18 shape instances for RDS Database Cloud Services for scale up. Shape instances are only tied to vCPUs and memory, not IOPS or throughput. Storage is a separate issue, although it can be scaled non-disruptively online. AWS RDS databases are severely limited to 64TB and 128TB in an HA configuration. That's still only 64TB raw and approximately 50TB usable. That's as big as it gets scaling up.

Elastic scalability in Redshift data warehouse clusters is not nearly as disruptive. They scale manually or via an API call. Resizing generally takes around 15 minutes to complete, according to AWS. Some tasks (not many) can continue to run in the background with the Redshift cluster fully available during the resizing. There is still some disruption upon scaling, meaning it is still likely to be scheduled downtime. And there is no read or write elastic scaling.

To address these quite severe RDS scale limitations, AWS recommends specifically that customers increase read-heavy Aurora performance by creating read replicas. Then placing those replicas near the users in different availability zones. This will scale read performance only. It does not address write performance, IOPS, throughput, or database size. This is what Amazon means by scale-out. It's not really scale-out and is a narrow use case work-around, and a costly one that customers pay for. Complex master-master sharding is the only way they can scale write performance. Sharding with AWS is labor-intensive, difficult to set up, operate, maintain, patch, upgrade, and, as a result, prone to human error. And it does not address database performance or scale. Amazon's RDS databases are simply not very scalable in performance or capacity.

AWS NoSQL (schema-less) Database Cloud Services present a similar story. They are share nothing architectures but with surprisingly similar limitations to RDS. Take DocumentDB (a AWS JSON MongoDB-like Database Cloud Service) for example. It only scales out to 64TB per database cluster. That's exceptionally low for a document database that can easily grow to petabytes. Latency, IOPS, and throughput are similarly constrained and limited. As with Aurora, Amazon's answer to these limitations is for customers to create read-only replicas and spread them around availability zones to scale read IOPS to millions of IOPS and nothing else. These duplicate databases consume storage and networking, thereby adding significant cost while scaling only the read IOPS.

[5] AWS vCPU is ½ a core.

AWS in-memory Database Cloud Services are a different animal. They can scale up to 250 Redis nodes or 300 Memcached nodes. To scale beyond requires a special request. Each node has limited memory of no more than approximately 684GB. Max ElastiCache for Redis memory scales to approximately 171TB in the 250 nodes whereas max ElastiCache for Memcached memory is approximately 273TB in 400 nodes. The large number of nodes and memory makes AWS ElastiCache excessively costly. And adding nodes is a simple, but manual and disruptive process forcing scheduled downtime.

Oracle Exadata Cloud Service X8M

Oracle Exadata Cloud Service X8M is built from the ground up for non-disruptive elastic scalability. It's designed to support any size workload at any scale. Application and workload database scalability is both transparent and highly available. OCPUs are scalable at a fine grain-level, increasing on-demand, and non-disruptively, while decreasing as demand ebbs, while only charging for the resources used, when they are used.

A single database can scale compute and storage independently with up to 4,600 CPU cores (1,600 database server cores in 32 database servers) and 3,000 storage server cores in 64 storage servers, 9,200 threads, 44 TB of DRAM, 96 TB of persistent memory (PMEM) and 1.6 PB of flash SSDs. That translates into databases up to 2.5PB uncompressed and 25PB with hybrid columnar compression (HCC).

On Database Cloud Service available cores alone, the Exadata Cloud Service X8M delivers 2,500% or 25X more cores than AWS Aurora. Storage capacity is at least 2,000% or 20X more than AWS Aurora.

The Oracle Exadata Cloud Service X8M delivers breakthrough elasticity. A customer can start small with a minimum size HA configuration of two Exadata X8M compute database servers and three storage servers[6]. All Exadata hardware is dedicated to one customer--eliminating the possibility of noisy neighbors. Compute and storage servers can be added independently as needed. Each compute server adds 50 cores and 1.3 TB of DRAM. Each storage server adds 49TB of database storage, 25TB of flash SSDs, and 1.5TB of PMEM. Expansion is fully elastic and occurs completely online—no scheduled downtime is required.

Conclusion: The Oracle Exadata Cloud Service X8M offers true non-disruptive, dynamic, fine-grain elastic scalability whereas AWS does not.

Non-disruptive operations

As previously discussed, customers want to eliminate or radically minimize all disruptive processes. Disruptive processes cause outages. Outages are obviously bad and have to be scheduled. Operational processes such as patching, upgrading, and as just discussed, non-disruptive scaling up and down are the keys to eliminating disruptive outages. Disruptive outages or planned downtime have a major negative impact on productivity across the organization, partners, and customers, delaying time-to-market, and negatively impacting revenues.

AWS Database Cloud Services

Patches, upgrades, and scalability shape changes via the hypervisor and multiple VM instances are disruptive. They require planned downtime.

Reducing this type of disruption requires an AWS RDS high availability (HA) option to what AWS refers to as multiple availability zones (AZ) utilizing a standby replica in a different availability zone. AWS keeps the RDS replica up to date synchronously or asynchronously when the circuit mile distance between AZs is more than 200 km. Different methodologies are utilized depending on the underlying database. Amazon implements hardware shape instance changes, patches, and upgrades to the standby first and then the primary. When applying to the primary, RDS automatically fails over to the standby. The disruption with HA Multi-AZ ranges from 30 seconds to 2 minutes depending on the underlying database software and network speed. This is by definition an outage disruption requiring scheduling.

None of the AWS Database Cloud Services should be called non-disruptive regardless of whether or not they have implemented the HA Multi-AZ option. That's because HA Multi-AZ is supposed to keep the

[6] Exadata X8M utilizes both compute/database servers and storage servers. The storage servers uniquely co-process and offload several database functions such as SQL, JSON, and decryption freeing up the computer/database servers for database processing. The storage server cores are not counted in OCPU fee calculations even though they significantly increase Oracle Database performance.

standby replica up to date synchronously, it can add a huge response time penalty to the primary Database Cloud Service. For synchronous replication, the local database cannot acknowledge the write until the remote database acknowledges the write. This takes time. How much latency occurs depends on the speed of light, the distance between the AZs, and the communication line speed. This is why any AZ beyond roughly 200 km circuit miles will make that latency unacceptable. In addition, beyond just paying for the HA Multi-AZ option, this latency adds cost in job time to completion (more vCPU time), reduces user/worker/customer productivity and revenue, in addition to delaying time-to-market. Always remember, cloud services are charged based on time. Additional latency increases the amount of time and cost to complete tasks. Disruptive scheduling adds to that cost. Nobody willingly wants to spend more time on the cloud than needed. The goal is always to get jobs done as fast and inexpensively as possible.

Oracle Exadata Cloud Service X8M

Non-disruptive operations are a focus for Oracle Cloud Infrastructure and where the Oracle Exadata Cloud Service X8M shines. Patches, upgrades, backups, failovers, DR, and scalability are uniquely non-disruptive. Automation of Oracle Real Application Clusters (RAC), Oracle Ksplice, Oracle Data Guard, and more makes it so. AWS Database Cloud Services are nowhere near comparable.

The value of non-disruptive operations cannot be overstated. No more late nights, weekends, or holidays implementing disruptive operations. Better security. Faster time-to-market. Greater revenues.

Conclusion: Just as it is for scalability, the Oracle Exadata Cloud Service X8M is operationally non-disruptive. Conversely, the AWS Database Cloud Services have not yet currently managed to do the same.

Higher Performance

Database Cloud Service performance has different measures for different models. Transactional databases, or OLTP, are typically measured in latency and IOPS. Data warehouses and other analytic databases are more commonly measured in throughput.

The old adage "time is money" has never been truer than with Database Cloud Services. Performance directly impacts the costs of a Database Cloud Service. An essential pricing component of Database Cloud Services is CPU consumption per hour, minute, or second depending on the cloud service provider. Database Cloud Service jobs that complete faster utilize less CPU time units. Lower CPU time units consumed equates to lower costs. There are several other cost factors that will be covered in the TCO section.

AWS Database Cloud Services

AWS Database Cloud Services are notably poor when it comes to performance. As previously noted, AWS RDS is limited to at most 256,000 IOPS. That's it. Amazon RDS latency is best case 1 millisecond (ms). IOPS performance can only be classified for AWS RDS as mediocre at best, with a low top end.

The AWS Redshift data warehouse performance is focused on throughput. Amazon's published max performance tops out at 2.3GB/s. That's a relatively low throughput, but adequate for smaller data warehouses.

AWS NoSQL Database Cloud Services are a bit of a different beast. AWS purports that their DynamoDB key value document database can scale to any size with millisecond performance. And for a significant additional cost, customers can implement the DynamoDB Accelerator (DAX) in-memory to improve DynamoDB read performance by up to 10X. That reduces the latency from milliseconds to several hundreds of microseconds even at millions of requests per second. Keep in mind that the DAX in-memory capability is severely limited in capacity to the amount of DRAM in each DAX node. However, DAX nodes are a clustered share nothing architecture, which should theoretically enable their performance to scale linearly as VM instances and storage is added. The difference between "should" and "reality" is commonly quite vast. Unfortunately, AWS has not published any test results for DynamoDB as of fourth quarter 2020.

The ElastiCache in-memory Database Cloud Services, just like DAX, are by definition limited by the amount of DRAM in the shape instances provided. They are not meant for large databases, but rather for database sizes that fit the limited DRAM. Although ElastiCache performance is quite good, it is incredibly expensive. The ElastiCache Database Cloud Service is meant to front-end other AWS RDS Database Cloud Services. It can conceptually reduce the amount of processing required in the other services. When it does so, it can somewhat offset some of the RDS cost. However, ElastiCache has severely limited DRAM capacity per node that in turn demands quite a few nodes to make it work for even moderately-sized databases.

One more thing about all of the AWS Database Cloud Services and hardware. AWS services designed to improve database performance such as AQUA for Redshift data warehouse queries and the aforementioned ElastiCache for RDS Aurora and other RDS databases, are incredibly costly. These services are not specifically co-engineered with the databases they accelerate, meaning they are somewhat inefficient for their cost. This is also evident in the underlying AWS infrastructure for their Database Cloud Services. AWS offers vCPUs (one half of a real CPU core), SATA flash SSDs, and HDDs. All of this hardware is commodity general purpose white box. The assumption is that the hardware does not matter since the database is virtualized. That is a false assumption. Hardware plays an important role in database performance. Hardware specifically optimized for the database can have an oversized positive impact on the database performance in terms of latency, IOPS, and throughput. AWS database-optimized hardware services are generic, lacking more efficient co-engineering.

Performance is not a noted strength for any of the AWS Database Cloud Services. Price/performance is comparatively higher than customers expect. For traditional relational databases in RDS, performance is definitely a weakness.

Oracle Exadata Cloud Service X8M

Oracle Exadata Cloud Service X8M is built for performance. Exadata X8M is engineered to deliver the best possible performance at scale for Oracle Database. Consider these metrics based on Wikibon's analysis:

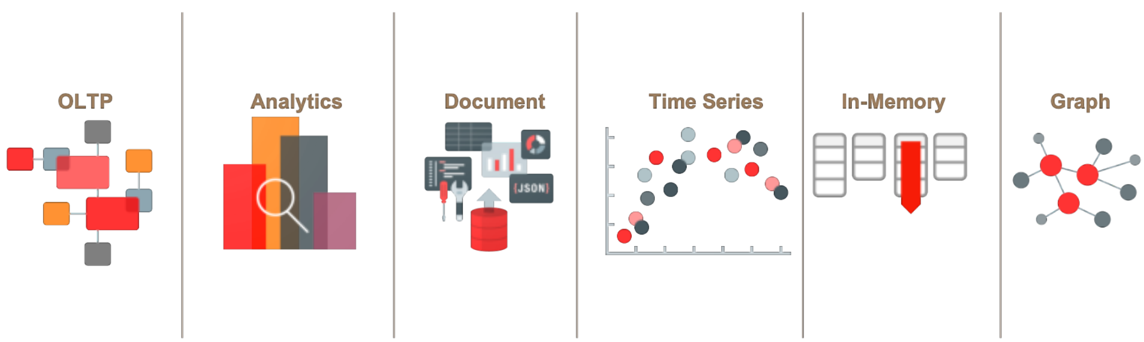
- 48 million SQL max read IOPS
- 30 million SQL max write IOPS
- 1.6 TB/s SQL max throughput
- Latency ≤ 19µs

These results make the Exadata Cloud Service X8M ideal for just about any workload including OLTP, data warehousing, key value document databases, XML object databases, time series databases, AI/machine learning, graph/spatial databases, and in-memory analytics, as well as consolidation of mixed workloads. All of these database models are built into the Oracle Database.

Conclusion: Exadata Cloud Service X8M's performance is multiple orders of magnitude greater than AWS Database Cloud Services, including Redshift, Aurora, and RDS. It's not close.

Simplified Data Extraction, Transformation, and Loading Between Database Models

There are multiple different database models. Database Cloud Service providers can take two different approaches to provide those different database models. They can either provide specialized Database Cloud Services for each model or a single Database Cloud Service that handles all data and model types in a converged database.



Each database model is structured differently. OLTP databases are the most common, generally relational, and built for fast access using indexes. Developers normally construct normalized data models for OLTP.

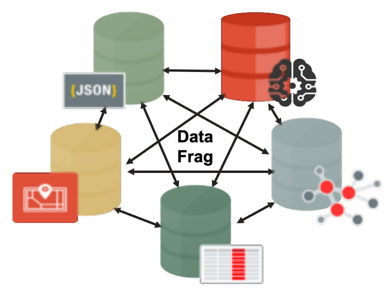
Analytic databases also tend to be relational, but typically use a dimensional model like Star Schemas and normally use massive parallelism to process data rather than indexes used in OLTP databases. Analytic databases such as data warehouses have very different data structures from OLTP. Optimizing for one will generally degrade the performance of the other.

Document databases store data in a de-normalized structure and frequently take advantage of JSON to organize the data. Document databases are also used for OLTP applications and can provide fast read access to relatively simple data structures.

Time Series databases are built to ingest data in time-oriented format, then provide time-oriented analysis on that data. Time series databases are also relational.

Most in-memory databases are designed for fast ingest and fast analysis of data by storing the data in memory. They're generally relational, however they're also occasionally used as cache for document databases.

Graph or spatial databases provide a completely different capability from most other databases. They're great for analyzing social networks, tracing contacts between people and places, or simply mapping out connections that are not readily apparent.



Disparate point-solution database models lead to data fragmentation and data silos. Each of these database models does reasonably well within their specific workload niche. The issue is that they don't work very well together. Data from different specialized models invariably requires integration. And, in reality, they frequently need to. This is especially evident for data warehouses and other analytical databases. Integration becomes a major and problematic challenge. This is not a new problem. It has been around for years, even decades. Those analytical databases are not where the data to be analyzed originates. It originates somewhere else, often in an OLTP database.

To get that OLTP data into the analytical database requires the data to be copied, extracted, transformed into a structure compatible with the data warehouse or other analytical database, and, finally, loaded into that analytical database. The shorthand for this process is called ETL or Extract, Transform, and Load. ETLs are not trivial. They require precious engineering resources to develop, document, maintain, and upgrade on an ongoing basis. They're constantly tweaked and modified as the different data from different sources is acquired. There are significant time lags in the data transfer causing delays. More importantly, the ETL process too often creates data integrity issues despite best efforts in error detection and correction.

This invariably leads to quite a bit of data movement across the network with data egress from one database followed by data ingress into another database. When the Database Cloud Services provider offers different database models in different services and charges for egress and ingress, this has consequences the most egregious of which is surprise bills each month.

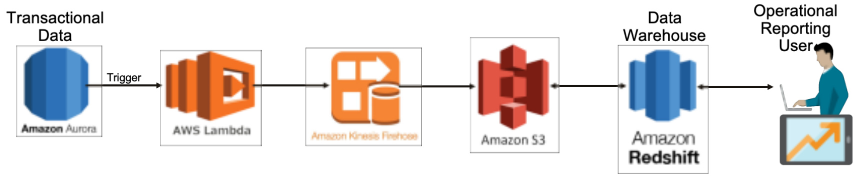
AWS and Oracle have taken take very diametrically opposed paths in their Database Cloud Services. AWS took the approach of a separate Database Cloud Service for each database model data type. Oracle took the approach of a single converged database with all database models and types included. Each approach delivers vastly diverse data movement, processes, and cost outcomes.

AWS Database Cloud Services

Because each of the AWS Database Cloud Services and third-party Database Cloud Services are independent and completely separate from one another, there is no native data interconnection. By definition, this creates the need for ETLs as previously discussed. AWS has a few different ways to accomplish that ETL, none of them technically elegant in their design.

Take a common situation where a user needs to run an operational report. The data resides in an AWS RDS database and, for the sake of argument, Aurora (the issue applies to every RDS OLTP database). OLTP databases are not designed to run this common type of report. It requires the power and technology of a data warehouse to process that data, AWS Redshift for example, but could just as easily be a third-party data warehouse such as Snowflake. This is not an outlier, but a frequent occurrence. Conversely, data warehouse-specific databases and most data warehouse cloud services are horrendous at transactions or OLTP. They're designed for massively parallel processing for reporting, analytics, and deep analytics.

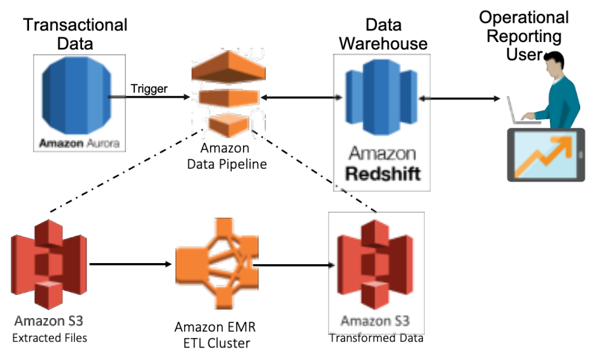
As shown in the figure below, the standard Amazon process to accomplish this operational report task is to write a trigger in the OLTP database to invoke what they call a Lambda function. Each trigger is typically only 10 to 20 lines of code, but the number of triggers can rapidly run into thousands of lines of code which consume chargeable Amazon compute resources. And Amazon also charges a few cents each time a Lambda function is called. This adds up quickly and can result in a surprisingly high bill each month. The Lambda function then calls Kinesis, yet another Amazon chargeable service. It's Amazon's equivalent of the Apache Kafka service. The Kinesis service then stores the data into Amazon's Object Storage S3 bucket. In other words, it duplicates the data adding yet more cost. Amazon Redshift can then finally query the data from that S3 bucket in parallel.



Many customers and prospects of AWS Database Cloud Services have objected to this complicated and non-trivial process. Amazon has acknowledged this and is now providing two additional paid managed services to simplify these ETLs. The first is called AWS data pipeline and the second more recent service is called Glue.

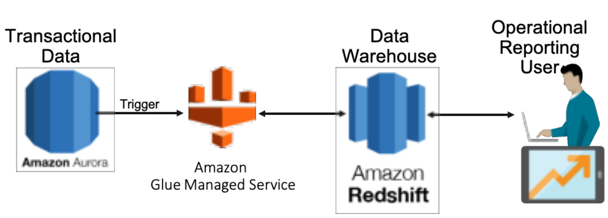
As shown in the figure below, the AWS data pipeline cloud service enables a semi-automated process to move data between different AWS compute and storage services at specified intervals. It's flexible with many built-in options for data handling. It can also control the instance and cluster types while managing the data pipeline. The data pipeline has built-in templates in the AWS console that attempt to simplify implementations. Depending on the business logic, condition check, and job logic they can be relatively user-friendly vs custom scripting.

However, there are several downsides to the AWS data pipeline. To begin with, the AWS data pipeline is not serverless and the pipeline internally triggers the EMR ETL cluster running behind the scenes, which rapidly increases fees. This can be more costly than the standard Lambda and Kinesis method. Job handling for complex pipelines is problematic, requiring significant skill sets. Sometimes it produces a non-meaningful exception error that makes troubleshooting quite difficult. And it's not even available in all regions – just refer to the AWS website.



Amazon's more sophisticated cloud ETL service is called Glue (shown in the figure below). This AWS managed service has three primary components: data catalog, ETL engine, and scheduler.

The data catalog is a common location that stores, accesses, and manages the metadata such as databases, tables, schemas, partitions, etc. Glue uses data source crawlers that infer schemas and objects within the data sources to create tables with metadata in the AWS Glue data catalog.



From the data catalog the ETL engine enables the DBA to create an ETL job by selecting the source and target data stores from the AWS Glue data catalog. Target schemas are provided by the data catalog. Then the DBA defines the ETL job where upon Glue generates PySpark code, which often needs to be customized based on validation and transformation requirements.

The Glue scheduler can then run the ETL jobs on-demand, or at a designated time, or triggered by the completion of a separate job. The scheduler can be set to automatically retry failed jobs.

Glue is completely serverless and does not carry any instance charges. It requires no resource management. Glue is charged for the query time and the data per unit, or DPU rate. It currently only works with Amazon Database Cloud Services. It does not currently work with third-party Database Cloud Services such as Snowflake, MongoDB Atlas, and others. Glue is a new service that is evolving. It's not recommended for complex ETL logic. It's also region limited – refer to the AWS website – similar to data pipe and, restricted internally to the Spark environment to process data.

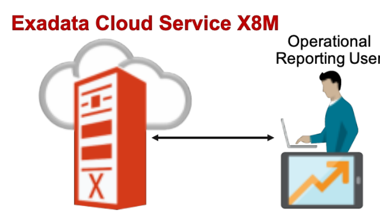
Spark has its own issues that affect Glue. Issues such as: no support for real-time processing; small files; no file management system; requires a lot of memory; inadequate number of algorithms; requires labor-intensive manual optimization; iterates in batches with each iteration scheduled and executed separately; relatively high latencies; no support for record-based window criteria – only time-based window criteria; and labor-intensive manual back pressure handling. One more thing about Glue, if dependent jobs and success/error handling is invoked, it requires in-depth knowledge of the other AWS data cloud services being utilized such as Lambda, CloudWatch, and data pipe. AWS Glue may simplify ETLs; however, it will add complexity, performance latency, and, most significantly, manual labor and cost.

Each one of the Amazon ETL services takes a substantial amount of time and skill sets. Time is the enemy of analytics, timely insights, and actions. That time frequently makes the data and analytical results stale, reducing their impact while adding increased costs as queries and data grow. And they always grow.

There are third-party ETL cloud services as well. Like AWS Glue, each adds processing, latency, and significant cost. Regardless of the service, multiple specialized Database Cloud Services require ETLs. They can be scripted by the user or a third-party ETL cloud service can be utilized.

In the case of AWS Database Cloud Services, Amazon is essentially charging its customers to solve a problem Amazon created with its distinct separate Database Cloud Services. In the end, each of the Amazon ETL offerings requires programming effort, knowledge, skill, resources, time, and money. Depending on the situation, it can translate into a lot of money, not even counting the impact of delayed insights or, perhaps worse, the use of stale data for analysis.

Oracle Exadata Cloud Service X8M

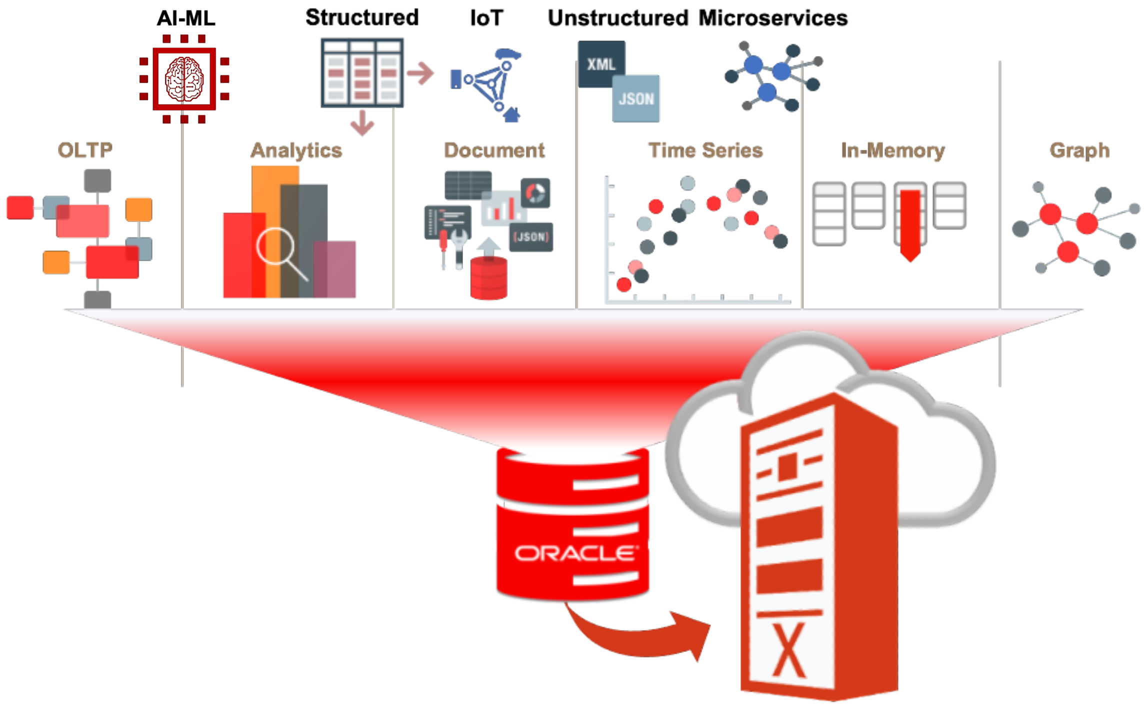


Oracle takes a very different approach with Oracle Database on Exadata Cloud Service X8M. Oracle engineered a Database Cloud Service that handles OLTP extremely well and which also includes the capability to process large volumes of data in parallel like a data warehouse for this type of operational reporting.

Making this work with high performance and high scalability is not easy. The market is littered with databases that attempted to do the same while having to compromise one or more database models, workloads, or model formats to make another acceptable.

Oracle has spent decades making its multi-model converged database work with best-in-class high performance for multiple models, workloads, and model formats. Exadata Cloud Service X8M delivers extensive database convergence:

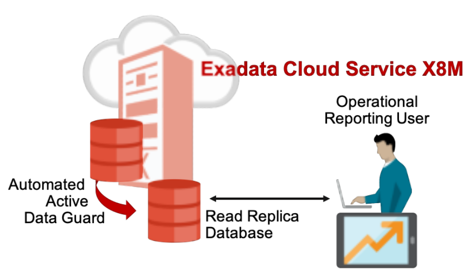
- Multi-Workload – OLTP, Analytics, Mixed.
- Multi-Model – Relational, Document, Graph, etc.
- Multiple Data Model Formats – 3NF, Star Schema, etc.



The advantages of building Exadata Cloud Service X8M around the Oracle Database EE and Exadata X8M architecture are conceptually analogous to Apple combining the hardware and iOS software in the multi-function iPhone. It combines discrete operations within a single managed service by:

- Mixing or splitting workloads, data types, and algorithms.
- Enabling SQL and transactions across any data type.
- Simplifying all database operations.
- Reducing cost.
- Increasing reliability.
- Unifying security and management across all data.
- Eliminating the need for ETL.
- Preventing data fragmentation and copy contagion.
- Removing initial and recurring integration costs.
- Empowering extensive synergy across database models, workflows, and data models.

This doesn't literally mean all of the data must be put in one database instance. Exadata Cloud Service X8M has extensive database isolation capabilities. Individual databases and workloads can be isolated. This enables tasks such as generating the aforementioned operational reports from conflicting with the OLTP transactions. Running this type of query in the OLTP database is isolated from that OLTP database so it doesn't impact transaction performance. However, this is not the only way to achieve isolation.



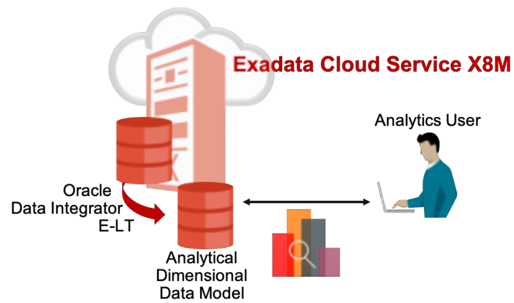
One way is to deploy a replica of the database for reporting using Autonomous Data Guard which is quite simple to deploy. It makes replicas easy for separate database models to access and utilize shared data without affecting the originating database.

Data warehouses and other analytical databases utilize different data models – data warehouses generally use a Star Schema or dimensional model—some data structure transformation is needed to enable data analysis. But because Exadata Cloud Service X8M is using the same underlying database in all database models, all the application coding, APIs, libraries, and the SQL dialect are the same for both the data warehouse and the OLTP database. This makes applications much easier to develop.

When it comes to in-memory database, Exadata Cloud Service X8M is very different from AWS ElastiCache. It's built-into the Oracle Database and is not a separately priced add-on. In-memory is automated, engineered with DRAM, PMEM, and NVMe flash SSDs. Most importantly, the customer determines what data, tables, etc. get put into memory. Exadata Cloud Service X8M does not force or require the customer to put the entire database into memory, only the parts that matter. Oracle in-memory empowers customer generated actionable operational insights and analytics directly on top of operational data. There is no ETL, or separate data warehouse required.

Exadata Cloud Service X8M additionally provides columnar flash cache to uniquely extend the in-memory into the storage servers. Combined with built-in smart scan and processing offload to the storage servers accelerates queries without additional cost. Oracle Database 21c introduced several new capabilities that greatly simplified and automated several processes greatly improving efficiency. Self-Managing In-Memory Column Store automatically manages the placement and removal of objects in the In-Memory Column Store. It then tracks usage patterns, moves, and evicts objects from the column store. Columns are automatically compressed based on usage patterns. The in-memory vector join algorithms clearly and significantly speeds up complex queries.

Some analytical models, workloads, and data model formats still require transformation to be usable. Oracle Cloud Infrastructure offers an intuitive and very easy-to-use automated method for transforming different data models called Oracle Data Integrator (ODI). ODI is based on a unique E-LT architecture (Extract - Load Transform). ODI delivers the highest performance possible for data transformation execution and validation processes while being extremely cost-effective.



ODI utilizes a new easier declarative design approach in defining data transformation and integration processes. This declarative design separates the declarative rules from the implementation details resulting in faster, simpler development, and maintenance. Data integration using declarative rules greatly reduces developer learning curves, improves their productivity, and appreciably simplifies ongoing maintenance. It separates processes definitions from implementation and the declarative rules (the "what") from the data flows (the "how").

The ODI declarative design delivers a higher level of data integrity. It detects and recycles faulty data before it's inserted into the target application without the need for programming. It does this by following data integrity rules and constraints defined in the application/database and ODI.

ODI also provides noticeably faster execution performance than traditional ETL integration software. In the cloud, time is literally money. ETL is generally based on proprietary engines performing data transformation row by row – which is tedious and slow. ODI E-LT architecture utilizes the built-in RDBMS engines and SQL to execute data transformations at a set-based level. This greatly accelerates those transformations on the target.

Efficiency and simplicity is another major ODI advantage. E-LT removes the requirement for an ETL Server to sit between the data sources and the target database. Source and target servers perform the complex transformations. Most of these transformations happen in batch mode when the server is not busy processing end-user queries.

ODI[7] saves a great deal of ELT hardware and software capital and operating expenditures cost through elimination of both the ETL server instance and ETL engine, maintenance, power, cooling, rack space, etc. It also greatly reduces overall labor costs via its reduced learning curves, accelerated development, increased developer productivity, less ongoing support and ongoing enhancements.

Conclusion: Exadata Cloud Service X8M is orders of magnitude easier when compared to the AWS Database Cloud Services. It's also easier in extracting, moving, transforming, and loading data between disparate database models. None of the AWS Database Cloud Services come close to Exadata Cloud Service X8M in

[7] Oracle Data Integrator Cloud Service listed at $0.7742/OCPU/hr. ODI Cloud Service BYOL listed at $0.1935/OCPU/hr. Oracle Cloud Price List.

simplifying data extraction, transformation, and loading between database models, because it's all coengineered within Oracle's Converged Database.

Automation

Automation is generally defined as technology that makes manual, and frequently labor-intensive, processes or procedures operate with minimal to zero human intervention. Automation condenses and simplifies multi-step processes and procedures. It's one of the primary motivations behind Enterprises moving to cloud-based SaaS and Database Cloud Services. Automation is extremely important in getting and keeping all manner of customers.

Not all automation is equivalent. The level of Database Cloud Service automation varies notably by cloud provider.

AWS Database Cloud Services

The philosophy behind many AWS services, including their Database Cloud Services, is self-service. Their idea is that the customer is knowledgeable, experienced, and skilled. AWS Database Cloud Services are designed to take over and automate much of the low-level database administration tasks such as the hardware infrastructure, patching, hot fixing, upgrading, and ongoing functioning of the baseline databases. But they fail to automate many of the more laborious database processes. There are no built-in automation processes for time-consuming tasks like security, tuning, indexing, re-indexing, backup, disaster recovery, root cause analysis, database virtual CPU scaling up or down, and recoveries.

AWS Database Cloud Services have automated high availability (HA). Automated HA only protects against hardware or availability zone (AZ) failures. It does not protect against database corruption, accidental or malicious deletions, and simple human error. That's where backup and DR come into play.

AWS also has automated database snapshots that back up the database and transaction logs once a day during a 30-minute backup window designated by the customer. Snapshots beyond the automated defaults are initiated by the customer manually. Keep in mind, snapshots are disruptive because they automatically quiesce the database, (stop it to ensure consistency), before the snapshot occurs.

And, as just discussed, ETLs are not automated. Even the Glue ETL service is limited in what it automates and usually requires that the customer make adjustments and frequent interventions to make it work effectively in the availability zones where it's available.

AWS Database Cloud Services are only modestly automated by design.

Oracle Exadata Cloud Service X8M

In contrast, Exadata Cloud Service X8M is highly automated with customer control. Exadata Cloud Service X8M automates significantly more than any other Database Cloud Service:

- One click online non-disruptive patch automation for databases and Exadata X8M infrastructure.
  - Simply "pre-check" and apply.
  - Patches are staged automatically when available (quarterly).
  - Patch history is readily available for each database and the Exadata X8M grid infrastructure.
  - Rolling patches are implemented leveraging Oracle Linux Ksplice and real application clusters (RAC), thereby eliminating disruptions.
  - One and done patching for multiple databases within a home instead of patching one database at a time.
- Intuitive Exadata Cloud Service X8M database homes creation.
  - Enables database homes creation from a very intuitive user interface (UI).
  - Or the migration of databases from and to other database homes.
    - Useful in moving databases from lower patch set homes to higher ones (19.6 to 19.7) and patch in process.
    - Also useful in moving databases from lower version homes (11.2/12.1/12.2/18c/19c) to higher version homes and upgrades in process.
- Automated database backup and recoveries per database on creation or any time thereafter.

- Notifications for failed backups.
- Database restores to latest backup or any point-in-time.

- Automated creation of highly available clusters to host RAC databases.
  - A process when manual typically takes time and expertise.
- Active Data Guard.
  - Automatically creates a standby or active database replica.
  - Empowers Data Guard instance creation straight from the console across Oracle Availability Domain (AD) or regions.
  - Hardware shapes and storage can be completely different for the primary and the Data Guard replica.
  - Full UI control for failovers, reinstatements, and switchovers.
- Automated dynamic OCPU scaling up and down.
  - Online and non-disruptive – no downtime required at all.
  - OCPUs appear instantly.
  - OCPUs charged 1-second billing in 1-minute minimum increments.
- Automatic indexing.
  - Uses AI and machine learning that outperforms the most experienced database administrators.
  - In tests it was able to outperform 15 years of highly skilled DBA indexing in NetSuite in 24 hrs.
- Automated tuning.
  - Optimizes performance based on user policies.
- Automated root cause analysis.
  - Dramatically decreases the time to finding and fixing root cause of database problems.

These are merely the principal automations. There are many more.

Conclusion: Oracle Exadata Cloud Service X8M is a very highly automated Database Cloud Service, second only to Oracles Autonomous Database Cloud Services. There are simply no equivalents in any of the AWS Database Cloud Services.

End-to-End Security

Security continues to be one of the biggest Enterprise anxieties when it comes to moving mission/business-critical applications to the cloud. Never more so than today with the increase in cyberattacks and the increasingly harsh privacy laws and regulations across the globe. Alleviating those concerns is paramount in getting customers to move Enterprise mission/business-critical databases to the public cloud.

AWS Database Cloud Services

AWS Database Cloud Services have several levels of security. Security includes data encryption at-rest, data encryption in-flight, access control, network isolation, database firewall, and database activity streams.

Databases are encrypted via keys the customer manages in the AWS Key Management Service (KMS). AWS RDS encryption covers the database, backups, replicas, and snapshots using AES-256 encryption. There are additional options to run in an AWS virtual private cloud (VPC), identity and access management (IAM), as well as cloud hardware security modules (CloudHSM) with FIPS 140-2 validated HSMs. However, Database Cloud Services do not mandate database encryption. It is a customer decision and must be selected by the customer.

Data encryption between the application and database instance uses standard SSL/TLS. Amazon RDS and their other Database Cloud Services are integrated with their IAM providing user- and role-based control on who can make changes. Amazon's VPC can isolate the database network limiting attack surfaces. It connects the AWS Database Cloud instance to on-premises applications via an IPsec VPN. In addition, network traffic entering and exiting each subnet can be allowed or denied via network ACLs and can be inspected by network firewalls, intrusion detection systems, and deep packet inspection on-premises.

AWS Database Cloud Service activity streams are designed to limit DBA data access. It limits access beyond database management stopping DBAs from collecting, transmitting, storing, or processing database activity streams. These streams are pushed to Amazon Kinesis Data Firehose where it can be analyzed by Amazon CloudWatch or third-party applications to provide audits and generate alerts.

Security measures generally consume time, adding latency to the database. Amazon does nothing today to mitigate the negative performance impact that their security has on their Database Cloud Services. AWS does not utilize any hardware offloads or optimizations to minimize the latency hits.

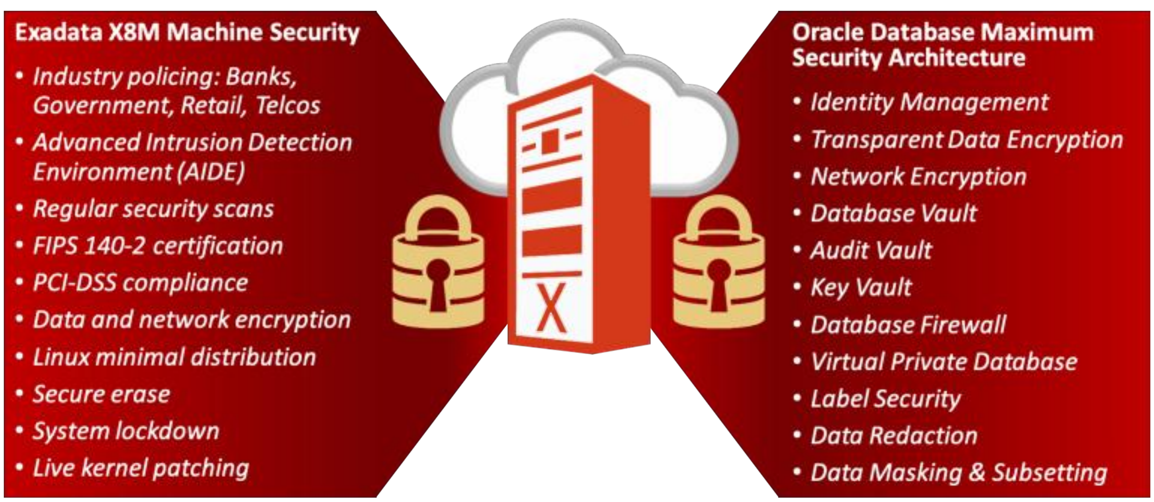
Oracle Exadata Cloud Service X8M

Oracle has a "security first" mentality in the design of all of its cloud services including the Exadata Cloud Service X8M. Encryption data protection is mandatory in all databases using TDE (Transparent Data Encryption). All backups are encrypted as well. SQL*Net connections are encrypted to and from the databases. OCI networking isolates the database services, i.e., no public IPs, with security lists and routing rules. And that's just the beginning.

Exadata Cloud Service X8M has substantial advanced security beyond just end-to-end encryption. It starts with a minimal attack surface and full stack non-disruptive patching. This is especially important when vulnerability patches are released. They are implemented quickly without causing database disruptions. In addition, the system stack is pre-scanned and fixed using STIG, Nessus, and Qualsys. The Oracle Exadata Cloud Service X8M is architected to fight malicious software and intrusion with Oracle Advanced Intrusion Detection Environment (AIDE)—similar to virus scanners and co-engineered with Oracle Database and Oracle Linux.

The security does not stop there. Exadata Cloud Service X8M integrates Intel Software Guard Extensions (SGX) into the Exadata Storage Servers. Identity and Access Management (IAM) is policy-based, while authentication on the operating system (OS) is not password-based.

And then there is the unique Oracle Data Safe, a unified control center for all of the customer's Oracle Databases. Deliberately priced very reasonably[8], it provides detailed insight into data sensitivity, data risks, and database security configurations. It also enables customers to discover sensitive data and where it's located; remove risk from non-production data sets by masking sensitive data; implement and monitor security controls; assess user security; configure audit policies; monitor user activity while identifying unusual behavior; and address data security compliance requirements by reviewing and mitigating risks bases on personally identifiable information (PII).



Conclusion: Security is intrinsic to and designed into Exadata Cloud Service X8M and Enterprise top-of-class. Security is automated with very little discretionary choices left to the customer. AWS Database Cloud Services' security on the other hand, is not outstanding but adequate. However, AWS' philosophy is that many aspects of security are the responsibility of the customer. Not all customers are equally proficient at securing their data, which has led to several well-publicized data hacks.

[8] Oracle Cloud Services Pricing

Availability

Database availability means being able to access and use the database. Mission/business-critical databases commonly run 24 hours a day, 7 days a week. Availability is crucial. Downtime is costly in terms of productivity, revenue, profits, reputation, and customers. The goal is to maximize availability as cost effectively as possible.

AWS Database Cloud Services

Amazon Database Cloud Services provide a service level agreement (SLA) of 99.9% uptime with several fine print caveats. It does not include any planned downtime or disruptions caused by features such as automated snapshots. Nor does it include any database software corruptions, errors, failures, malware, etc.

As previously discussed in the “non-disruptive” section, AWS Database Cloud Services have an optional HA Multi-AZ service. This service enables automatic failover without human intervention should there be a failure in primary availability zone, network connectivity, compute unit, or storage. Amazon specifically calls out that HA Multi-AZ “does not failover automatically in response to database operations such as long running queries, deadlocks or database corruption errors.” And because the HA replica is synchronously mirrored or replicated, any corruptions, human errors, maliciousness, or malware in the primary RDS database is by definition replicated to the HA database instance. There is no rollback to non-affected points in time.

To roll back to a non-affected database requires their automated database snapshot backups be stored in AWS S3 object storage. Automated backups occur daily during a user-configurable 30-minute backup window. If the backup requires more time than allotted to the backup window, the backup will continue to completion. They’re enabled by default for new database instances. The default window is selected at random during an 8-hour block of time within the region. The first backup is a full backup followed thereafter with incrementals.

Backup retention is 0 to 35 days configurable by the user. The default retention period is one day when the database instance is created via the RDS API or seven days when created via the AWS Console. It’s also important to note that the backup window can’t overlap the weekly maintenance window for a given database instance.

Manual snapshots are a different story. They’re limited to no more than 50 per instance per region. Whereas there are no limits to the automated backups, manual snapshots are kept until the customer explicitly deletes them. Manual snapshots are not free. They consume storage.

When a customer changes a backup retention period from 0 to non-zero value, an immediate outage occurs with the first backup. It also occurs if they turned off the automatic backups and delete all existing automated backups for that database instance.

RDS combines the periodic database snapshot backups with the transaction logs to enable restoration of the DB Instance to any second during the retention period (typically up to the last few minutes).

When the backups are in a single AZ, storage IO is briefly suspended while the backup process initializes (typically under a few seconds) and a brief period of elevated latency is likely to be experienced. When the database deployments are multi-AZ, there is no IO suspension because the backup is performed off the standby instance.

AWS claims database restores from snapshots typically occur within five minutes of customer initiation. However, the recovery point objectives (RPO), or the amount of data that can be lost, are generally quite large for non-RDS database instances with the default being 24 hours. In addition, any database patches, hot fixes, and updates that occurred after the snapshot that’s used for the restore, will have to be manually reinstalled (disruptively), adding more outage time.

Oracle Exadata Cloud Service X8M

Oracle provides unprecedented availability for the Exadata Cloud Service X8M. The Oracle SLA for availability is 99.95%, does not have any fine print, and includes any unplanned AND planned downtime.

This is because the Exadata Cloud Service X8M is built on the Oracle Maximum Availability Architecture[9] (MAA). Oracle MAA has evolved to be the industry de facto standard for database high availability.

Take the RAC example. RAC does not just provide application transparent scalability. It's also application transparent availability. Should an Exadata X8M RAC database server node fail, the entire RAC is reconfigured in ≤ 3 seconds with no impact on the applications. In addition, with Oracle Database 21c, PDBs within a CDB can be configured for standbys and failover PDBs individually.

Active Data Guard is another example. Active Data Guard is more than a standby database locally or in another availability domain (AD). Its far sync enables zero data loss protection at any distance, continuously updating the remote database as the local is updated on a block-level basis. This currently requires a DBA to configure.

Active Data Guard also automatically detects and repairs damaged blocks with user-transparent recovery of corrupted database blocks from either the primary or the standby database. And its standby block change tracking delivers incremental backups on the standby (manually configurable). In addition, the global data services load balance connection requests and provides integrated service management across replicated databases, considering placement of connections depending on read or read/write workload.

But most importantly, Active Data Guard delivers unrivaled application continuity. It makes outages transparent to applications.

Conclusion: Oracle is quite conscientious about database availability based on its decades as a Tier-1 database provider to the Enterprise. Oracle goes well beyond common public cloud service level agreements for availability. AWS is generally in line with typical public cloud service level agreements and does not consider planned downtime a part of its SLAs.

ACID

ACID (atomicity, consistency, isolation, and durability) is the cornerstone of any relational database. It's a set of properties that are intended to guarantee data validity despite errors, power failures, system outages, and other mishaps. ACID is mandatory for every relational database. However, for most NoSQL databases (a.k.a. Schemeless databases,) the consistency part is generally not instantaneous; instead, it's eventually consistent.

AWS Database Cloud Services

All of AWS Relational Database Services (RDS) meet the ACID properties. That includes Aurora, PostgreSQL, Oracle Database SE/SE2, MySQL, MariaDB, Redshift, Timestream, QLDB, and the ElastiCache in-memory services when caching for a relational database service.

AWS' non-relational Database Cloud Services do not meet the ACID properties because they are only eventually consistent, including DynamoDB, DocumentDB, Keyspaces, and Neptune.

Oracle Exadata Cloud Service X8M

Every database model, workload, and database data model in Oracle Database running on Exadata Cloud Service X8M meet the ACID properties, including the database models typically found in schema-less NoSQL databases such as JSON/document, XML/Object, key value, Graph, Spatial, etc.

Many eventually consistent database users get tired or exasperated at the lack of immediate consistency. Oracle has eliminated that problem.

Conclusion: Both AWS and Oracle meet ACID requirements for their relational databases. However, Oracle goes further by providing ACID for databases considered non-relational or schema optional databases such as document, graph, spatial, object, key value, and more.

Total Cost of Ownership (TCO)

TCO is much more than just how Database Cloud Services are licensed. Database Cloud Service costs obviously include any reserved or on-demand infrastructure, virtual CPU cores utilized, unit of

[9] MAA best practices are created and maintained by a team of Oracle developers that continually validate the integrated use of Oracle Database High Availability features. Real-world customer experience is also fed back into the validation performed by the MAA team, spreading lessons learned to other customers.

time granularity for on-demand resources (per hour, minute, or second), memory, storage, networking, other services such as those required for ETLs, etc. Other factors include cost of non-granular capacity increments and performance. This can be difficult to estimate ahead of time.

Most Database Cloud Service providers have a cost estimator that can help. However, it's only an estimate. Actual fees are based on what's actually used or consumed and paid in arrears. Pricing is often quite complicated.

What most estimators and TCO calculators leave out is the other larger costs. Costs that include downtime or outages, personnel time for labor-intensive tasks, lost time from troubleshooting, decrease in knowledge worker productivity[10], and the value of time-to-market[11].

AWS Database Cloud Services

AWS has different fees for each of their 16 Database Cloud Services. There are shape instance fees—memory, storage, and vCPUs. The fee depends on whether Amazon is providing the database software, or the customer is bringing their own license. The fees can be quite complicated depending on the shape, whether the vCPUs are spot or unlimited. It's important to note that vCPUs are actually half a physical core. Determining the number of vCPUs required can be an issue if this is not taken into account. Cost per vCPU per hour is on par with Oracle's OCPU except Oracle's OCPU is a full real core making AWS twice as much in reality per vCPU hour.

AWS has a slew of other fees including API, long-term retention, ETL services, data egress, and more. Performance is another issue. High latency (≥ 1ms) means longer times to complete queries and jobs in general. Low IOPS (80,000 or 256,000 max) means databases need to be replicated to meet performance requirements that exceed those IOPS. Both add significant out-of-pocket costs. Customers can be surprised at the cost when they get the bill.

What's notable is that the open-source AWS Database Cloud Services also increase the above-the-line costs of reduced productivity and longer times to market because of a severe lack of automation and intuitive tools. They tend to be highly disruptive and are single model databases with severely limited scalability and flexibility. This seriously delays time-to-market and time-to-actionable-insights. Those above-the-line costs can and often do, far exceed any out-of-pocket cost savings from utilizing open-source databases.

Oracle Exadata Cloud Service X8M

Oracle Exadata Cloud Service X8M also charges fees based on whether Oracle provides the Oracle Database EE, or the customer brings their own license (BYOL). However, how Oracle charges is very different from AWS.

There are charges for the underlying infrastructure starting with a base rack. The database fees start at just 2 OCPUs.[12] This is 95% less than the 48 licenses required for similar database servers on AWS RDS. And each OCPU delivers twice the performance of each AWS vCPU. That's a significant price/performance advantage for Oracle. That advantage remains constant as Exadata Cloud Service X8M scales; however, it scales more that 25X or 2,500% more than AWS Aurora or any other RDS database. It actually has even better price performance and scales to 73X more than AWS Aurora or any other RDS database because Exadata X8M utilizes the CPU cores in the storage servers to offload SQL, JSON, decryption, and more, but Oracle does not charge licensing fees for those CPU cores.

For BYOL customers, Oracle charges for the same underlying infrastructure. In addition, there is a PaaS fee that covers the Exadata software, TDE, and additional management packs.

Performance is a huge price advantage for Exadata Cloud Service X8M. Remember that time is money in cloud fees. Oracle OCPU scaling is dynamic, both up and down. Oracle charges for OCPUs with 1-second billing in 1-minute minimum increments. This is where the performance advantage shows. Queries, transactions, jobs, and workloads that complete faster cost less. They will complete anywhere from 2X to 50X faster versus various AWS RDS Database Cloud Services. That translates into a much lower charged fee.

---

[10] The value of sub-second response time is well known in its positive impact on productivity. Reducing application response time from 3 seconds to .3 seconds typically results in more than 2X productivity gains. https://jlelliotton.blogspot.com/p/the-economic-value-of-rapid-response.html
[11] Longer time to market results in lost revenues that can never be reclaimed whereas faster time to market can result in unanticipated revenues.
[12] Oracle Exadata Cloud Service X8M Pricing

Faster completion of all database jobs also translates into higher productivity and faster time-to-market. As a result, TCO for Exadata Cloud Service X8M is much lower than AWS Database Cloud Services.

Remember, Exadata Cloud Service X8M does not require ETLs for many of its databases. And those that do require fewer services and lower cost using Oracle ODI E-LT.

One more cost advantage for Exadata Cloud Service X8M comes from the Hybrid Columnar Compression (HCC) technology. HCC reduces storage costs by approximately 10X for cooler data, data warehouses, and big data analytics. A 10X reduction in storage costs is massively significant. HCC only works with the Oracle Database and Oracle engineered systems such as the Exadata X8M. It does not work on AWS.

Oracle customers that have switched from AWS to Oracle Cloud Infrastructure have reported cost savings ranging from 2X to 70X. This is why Oracle CTO and founder Larry Ellison has said most database cloud users will save at least half of what they're spending on AWS.

Comparative Oracle Database BYOL

The conventional wisdom says that in order to level the playing field, costs should be compared based on the Enterprise customer already having a license deal with Oracle and utilizing BYOL. This takes the cost of licensing the Oracle Database on AWS out of the equation.

Except it turns out the AWS cloud infrastructure costs at MSRP to run the BYOL Oracle Database are much more expensive than customers expect. And compared to Oracle Exadata Cloud Service X8M MSRP, it comes in 2-3X more expensive. A look at this actual customer example shows how.

The customer had approximately 90 Oracle Databases in a variety of EC2 shapes ranging from 1 to 32 cores (2-64 vCPUs). Keep in mind AWS does not allow database consolidation. Unused cores and memory in one EC2 instance cannot be shared with databases running in another EC2 shape. Here is what they were spending based on MSRP.

RDS Instances	90	$1,003,311.00
Provisioned IOPS Storage	59,330	$88,995.00
	Total Annual Cost	**$1,361,094.00**

Contrast that with MSRP cost when running in the Oracle Exadata Cloud Service X8M.

Exadata Cloud Service Components		
Exadata Database Machine Quarter Racks	2	$254,323.82
Additional Database Servers	0	$0.00
Additional Storage Servers	0	$0.00
Base OCPU (24x7x365)	64	$180,862.72
Burst OCPU (10% of time annually)	34	$9,608.40
Total Annual Cost		**$444,794.94**

AWS cost more than 3X more. Even assuming maximum discounts from AWS, they were still costing the customer much greater than 2.5X more.

Much of the cost differences can be traced to AWS not allowing database consolidation or the sharing of resources between Oracle Database instances. The customer received considerably less usable infrastructure from AWS than they would from OCI. AWS isolation leads to greater cost. Consolidation is required to more closely match utilization. Moving to Oracle Exadata Cloud Service X8M results in approximately a 67% decrease in annual cost. Put another way, AWS RDS cost 306% more than using Exadata Cloud Service X8M for the same workloads.

Consider that based on the customer's use, their average AWS infrastructure utilization equaled approximately 17%. This means that on average approximately 83% of those resources remained idle. The utilization is noticeably lower than provisioned, but the customer was charged for the entire 100% of the resources allocated. A side-by-side comparison of AWS cloud infrastructure versus Oracle Exadata Cloud Service X8M for Oracle Database BYOL reveals stark contrasts.

Amazon RDS Oracle BYOL		Exadata Cloud Service X8M Oracle BYOL	
Oracle Databases	90	45	Oracle Databases
RDS Instances	90	1	Exadata Quarter Racks
		2	Exadata Database Servers
RDS Cores Provisioned	576	200	Exadata Cores Provisioned
RDS Cores Used Peak (Isolated)	150	100	Exadata Cores Used Peak (consolidated)
RDS Cores Spare Capacity (not sharable)	74%	50%	Exadata Cores Spare Capacity (sharable)
RDS Memory Provisioned (GB)	6,754	560	Exadata Memory
RDS Memory Used (GB)	4,514		
EBS Storage Space GB (allocated)	81,930	243,712	Exadata DB Storage
EBS Storage Space GB (used)	42,348	157,286	Exadata FlashCache
Provisioned IOPS	198,000	8,820,000	Exadata Included IOPS

Oracle enables all of the resources to be shared by all 90 Oracle Databases. Resources are sized to combined workloads. And Oracle's elastic dynamic scaling up and down means the customer only pays for what they actually consume.

Conclusion: It does not matter whether the customer is looking at a complete Database-as-a-Service offering from AWS and Oracle, or just an Oracle Database BYOL, Oracle Exadata Cloud Service X8M has a much lower TCO.

Overall Conclusion

Database Cloud Services (a.k.a. database as a service or DBaaS) are designed to simplify the use of databases for most organizations by taking a significant amount of DBA labor out of the picture. Most of these Database Cloud Services are designed for medium to small environments or databases that are not mission/business-critical. They are generally not architected for the demands of the Enterprise, most large IT organizations, governments, or IT organizations where mission/business-critical applications and workloads are essential, or IT organizations where performance, elasticity, scalability, and availability are paramount.

Both AWS and Oracle promote their Database Cloud Services as architected to meet those pressing business demands. A deeper dive reveals some hard truths. The first is that Oracle has leveraged its more than four decades as an Enterprise database Tier-1 provider and more than 10 years of delivering co-engineered Exadata hardware and Oracle Database software into the class-leading Enterprise-grade Exadata Cloud Service X8M. It definitely meets the requirements of Enterprises and anyone else requiring a mission/business-critical database service in the cloud.

The second is that AWS has made tremendous strides in their Database Cloud Services, but they come up short in meeting Enterprise-specific requirements. The AWS databases upon which their services are based lack fundamental mission/business-critical competencies. Even the AWS Oracle Database offering is the lower-level Oracle Database Standard Edition 1 or 2, lacking many of the critical capabilities of the Enterprise Edition. Aurora and Redshift are anemic in terms of elasticity, performance, and scale when compared to Exadata Cloud Service X8M. In short, AWS Database Cloud Services have a long way to go to be truly Enterprise or just simply mission/business-critical.

Along the dimensions reviewed in this report—mission/business-critical, functionality loss, elastic scalability, non-disruptive operations, performance, ETL, automation, security, availability, ACID, and TCO—the advantages of Oracle Database on Exadata Cloud Service X8M are decisive. As documented, the AWS

Database Cloud Services delivers less functionality, operates slower, has more outages/downtime, and generally costs 2-3 times as much.

If Enterprise-class Database Cloud Services are required, the answer is not AWS Aurora, Redshift, or any other AWS Database Cloud Service.

Wikibon believes that business and government organization running larger-scale mission-critical applications based on Oracle workloads are best suited for the Oracle Cloud Infrastructure Exadata Cloud Service X8M. These organizations will be able to complete migration to a full function OCI database platform faster and operate at an appreciably lower cost. They will operate faster, decrease their time-to-market, decrease their time-to-actionable-insights, scale much bigger with fine grain elasticity, have fewer disruptions, and reduce their TCO. The facts make this crystal clear.

For More Information on these specific Database Cloud Services go to:

Oracle Exadata Cloud Service X8M

AWS Relational Database Service (RDS)

Appendix A: Comparison Summarization

Enterprise DB Cloud Services Comparison	AWS DB Cloud Services	Oracle Exadata Cloud Service X8M
Mission/Business Critical	ø	√
No Functionality Loss	ø	√
Non disruptive Elastic Scaling	ø	√
Max Uncompressed DB Size	64 TB	2.5 PB
Data Warehouse DB Size	192 TB	25 PB
Max DB Cores Per Workload	64	1,600
Max Cores Per SQL Offload	ø	3,072
Operationally Non disruptive	ø	√
Higher Performance	ø	√
SQL Read Latency	> 1 ms	≤ 19 µs
SQL Read IOPS	80 K	48 M
SQL Write IOPs	80 K	30 M
SQL Throughput	2.3 GB/s	1,600 GB/s
Simplified ETL	ø	√
Discreet DB cloud services	16	1 Converged DB
Significant Automation	ø	√
End to End Security	Yes w/caveats	√
Mandatory Encryption	ø	Yes: At rest/in-flight
Availability	99.9%	99.995%
ACID	Yes: RDS only	√
Conversion Costs	>	<
TCO	>	<
Complete DBaaS TCO	>	<
Complete BYOL TCO	>	<

