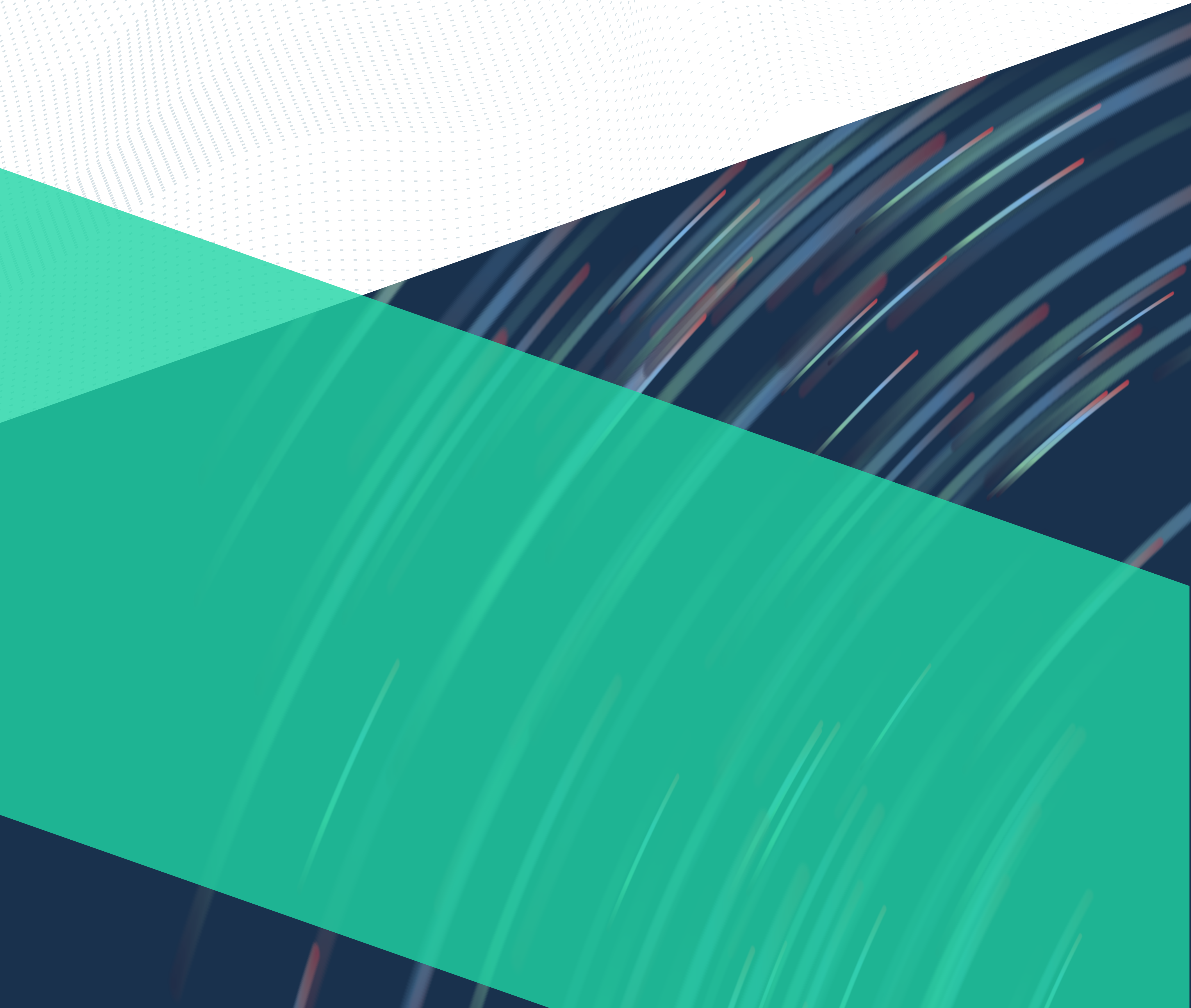
house of brick

YOUR COMPLETE GUIDE

Migrating Oracle to AWS



Migrating Oracle to AWS: Your Complete Guide

Introduction

The desire to run Oracle software in the AWS cloud brings arguably the most popular enterprise software vendor to the most popular public cloud platform. It makes a lot of sense to run Oracle software on AWS for the benefit of public, private, and public sector organizations worldwide. While it makes logical sense, figuring out the actual strategy for Oracle on AWS can be fraught with misleading information, incorrect assumptions, risky compliance traps, and costly misaligned budgets. Having a good understanding of these unique dynamics will provide for a trusted, secure, cost managed cloud environment for Oracle-based enterprise applications.

In discussing Oracle to cloud migration considerations, Gartner stated the following

> **"Migrating Oracle workloads to cloud has a strong licensing and support impact. Not all Oracle products are deployable on all cloud platforms, and their capabilities may limit migration initiatives. I&O leaders should evaluate the viability, benefits and drawbacks of migrating Oracle products on them. "**

This guide will provide the House of Brick real-world perspective on the critical considerations (several of which are also discussed in Gartner's analysis), and how to evaluate these criteria to develop your Oracle on AWS cloud strategy. The key topics that we cover are:

- Oracle Licensing
- Cloud Sizing and Architecture
- Cost Anticipation and Control
- Database Migration
- Cloud Governance

House of Brick has comprehensive solutions that aid in the development and implementation of a strategy to run Oracle on AWS. These solutions include a combination of innovative compliance monitoring software as well as professional and managed services:

- **Professional Services** – advanced consulting to assess cloud readiness, security posture, Oracle license compliance, cost optimization, architecture, and implementation of your business-critical application and database migration.
- **Managed Services** – operational support for operating Oracle in a public or private cloud environment, including managed license compliance.
- **OpsCompass Monitoring** – active monitoring and management of cloud inventory, configuration drift, security posture, and Oracle license compliance.

For more information on OpsCompass, visit opscompass.com

Migrating Oracle to AWS: Your Complete Guide

Licensing

House of Brick has seen many clients make costly missteps in database cloud migrations despite solid technical planning efforts. The wildcard that must be accounted for is always Oracle licensing. As a general rule, the annualized Oracle Database licensing ccosts for any given cloud deployment are higher than the combined cost of all the cloud infrastructure that the software runs on. Given that cost imbalance, mistakes in licensing can easily eliminate the return on investment (ROI) for any cloud migration initiative.

Know Your Current License Posture

The proper first step in any planned public cloud migration initiative should be to completely understand the licensing position of the current environment. To accomplish this a complete list of Oracle entitlement including product, metric, quantity, and any restrictions is a must when starting any cloud migration project. Many House of Brick clients find this far more challenging that they might initially assume. Particular challenges exist for clients that have had years as an Oracle customer with incremental purchases spanning years or have inherited multiple licensing portfolios after merger and acquisition activity.

Once a list of all owned Oracle licensing entitlement is assembled, reconciling it with actual usage can be another daunting challenge. While tallying a list of all installed servers or virtual machines with Oracle software isn't too tricky, there are additional challenges in verifying which separately licensed options or packs each system is utilizing beyond the basic database features.

What happens if your Oracle Licensing isn't ready?

- Significant financial liability
- Organizational disfunction
- Non-technical challenges

Care should be taken to compare the list of available and owned Oracle licenses to the in-use Oracle deployments both to determine if there is any unlicensed usage that might required remediation as well as to validate any excess or unused licenses that are available for migration projects.

Planning how to apply Oracle databases licenses during cloud migrations can be complicated by the existence of special licensing rules for public cloud environment. The Oracle document "Licensing Oracle Software in the Cloud Computing Environment," more commonly called the Cloud Policy, is an extra-contractual document that defines how Oracle licenses can be used in AWS and/or Azure. A thorough review of current entitlement and the Cloud Policy is necessary to properly plan a cloud migration that doesn't incur ballooning licensing liabilities.

Understand the migration's impact on usage

When migrating Oracle databases to the cloud, a very common issue is that it creates a temporary spike in Oracle database usage. As new cloud platforms are stood up and data is being migrated, there is a strong risk that these intermediate states will cause licensing violations by running more than databases than owned licenses can cover. Contrary to popular belief, there is no exception or carve-out in Oracle licensing agreements for migrations. Every environment must be fully licensed for the duration of its existence. Attempting to perform a migration without maintaining license compliance is a significant risk to an organization and can easily represent millions of dollars of liability if Oracle were to perform a license audit during migration.

Assess impact of SaaS applications

In addition to considering the number and type of licenses available, it is also critical to determine who the users of the application are. The standard Oracle license agreement is for “internal business purposes only”. If the application using the Oracle database is for the benefit of customers or other third-parties, it may require what Oracle calls “Proprietary Application Hosting” rights. This is an add-on grant of rights that allows a license for an Oracle database product to be used for the benefit of customers or other third-parties. These rights are usable in the cloud, but House of Brick frequently sees clients overlooking this and provisioning external-facing applications without the correct rights.

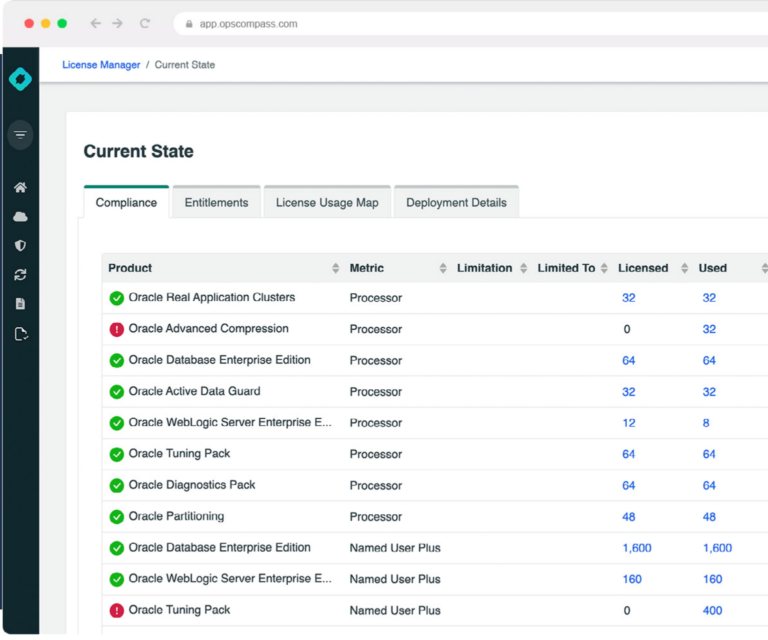
One valuable technique to remove licensing liability, for applications that may not have a hard reliance on Oracle Enterprise Edition features, is to utilize the Amazon RDS Oracle Standard Edition License Included option. This offering allows a way to use Oracle Standard Edition without requiring the usual multi-year licensing agreement with Oracle. Instead the licensing is provided by Amazon and the cost is bundled into the compute instance.

Peace of mind from agreement to deployment

House of Brick uses OpsCompass’ product every step of the way and using it’s License Management feature is designed to help you avoid unnecessary license costs by automatically monitoring your Oracle systems for license compliance issues.

- Visibility into Software Usage
- Continuous Software Analysis
- Audit Readiness and Defense

For more information on OpsCompass, visit opscompass.com



Current State

Product	Metric	Limitation	Limited To	Licensed	Used
✓ Oracle Real Application Clusters	Processor			32	32
! Oracle Advanced Compression	Processor			0	32
✓ Oracle Database Enterprise Edition	Processor			64	64
✓ Oracle Active Data Guard	Processor			32	32
✓ Oracle WebLogic Server Enterprise E...	Processor			12	8
✓ Oracle Tuning Pack	Processor			64	64
✓ Oracle Diagnostics Pack	Processor			64	64
✓ Oracle Partitioning	Processor			48	48
✓ Oracle Database Enterprise Edition	Named User Plus			1,600	1,600
✓ Oracle WebLogic Server Enterprise E...	Named User Plus			160	160
! Oracle Tuning Pack	Named User Plus			0	400

Migrating Oracle to AWS: Your Complete Guide

Sizing and Architecture

House of Brick frequently sees customers incur unnecessary costs or have business-affecting performance issues due to a poorly architected cloud migration that doesn’t take into account the true performance needs of each database workload. Doing proper analysis around performance and capacity needs for database workloads in advance of a cloud migration is critical for avoiding both. When moving to the cloud it’s necessary to adjust architectural approach to take into account the flexibility of cloud environments. Traditional on-premises sizing methods tend to be tied to multi-year hardware refresh cycles, and thus it’s traditional to overprovision CPU, RAM, and storage IO resources in order to leave headroom for growth. This kind of sizing mentality translates badly to the cloud. It leads to wasteful cloud spending and migrations that can’t realize any of the savings that result in adopting a flexible public cloud computing paradigm.

Right-sizing and capacity planning

The first and most obvious sizing metric to consider is database disk size.   Sizing information seems relatively simple to gather from existing database, but in practice House of Brick sees many clients make incorrect estimations of needed database size. Database sizing needs to take into account not only the disk space required for the database datafiles but also space required for transaction logs, backups, temporary files, and required software.

The next challenge is gathering reliable and accurate information on CPU and I/O metrics. This information is readily available from a variety of tools, but mixed workloads often complicate identifying how to interpret the data and isolate the portion that corresponds to individual databases. In House of Brick's experience, the most reliable source of performance metrics on Oracle database workloads is the internal AWR or Statspack data available (depending on licensing) directly from the database. This data, while sometimes hard to interpret, is the only source of metrics that is reflective only of the demands of the database itself.

This is an important consideration as database workloads in on-premises environment often exist in virtual machines that host one or more databases, so much of the available performance data is very mixed and reflects multiple workloads.

What are your Sizing and Architecture risks during a migration?

- Large avoidable excess costs
- Serious performance issues
- Perception of a failed migration

Consider service levels and availability

Another important piece of data to gather and analyze in preparation for a cloud migration is service level agreement (SLA) data or other commitments around service quality and availability. There are multiple options available in public cloud environments for architecting highly available (HA) database environments and provisioning disaster recovery (DR) capabilities. As is always the case in architecting appropriate HA/DR, it is extremely difficult to make appropriate decision about design and especially costs without having clear targets to design towards.

House of Brick has had to help many client organizations that have created public cloud architectures that didn't seem to meet expectations for performance, scalability, and reliability. In most cases these weren't actually caused by defective architectures per se, but rather architectures that didn't meet the goals of the organizations because those goals were undocumented or vague.

Understand your migration options

When planning to migrate Oracle database to the AWS public cloud environment, it's important to be aware of the options for migration. The AWS EC2 service offers traditional cloud compute virtual machines that be installed with a variety of operating systems and applications. Configuring an EC2 instance to act as an Oracle database is quite possible and the most familiar option for organizations used to managing Oracle databases in their on-premises environment. AWS also offers another option via their Relation Database Service (RDS).

The Amazon RDS for Oracle offering provides a managed database service that allows automated deployments and maintenance and patching for Oracle databases. In many cases this is a superior option to self-managing an Oracle workload on EC2, as RDS automates away much of the undifferentiated heavy-lifting tasks involved in provisioning and maintaining an Oracle database. There are however a few limitations on RDS support for Oracle features and configurations that should be carefully reviewed. It is critical to understand which database workloads in a migration are good candidates for RDS before an appropriate architecture can be devised.

The final piece of data, specific to the Oracle realm, that should be addressed while planning a loud migration is whether Oracle Real Applications Clusters (RAC) is in use. If it is in use, understanding specifically why it's being used is critical. Many organizations that House of Brick assists in planning cloud migrations has had a hard time formalizing what specific requirements are being met by deploying Oracle RAC. Oracle RAC does bring impressive features in terms of high availability and in some cases performance, but those features come at a heavy premium. Oracle RAC is very costly to operate in terms of software licenses, specialized infrastructure requirements, and skilled personnel requirements. During a major infrastructure change like a cloud migration is it very appropriate to consider whether Oracle RAC is required or if it is time to transition away from Oracle RAC and just use traditional Oracle databases or managed Oracle databases in the RDS service instead.

Once all of this data is gathered and analyzed, the last and perhaps trickiest step is translating all this into an effective cloud architecture. Be very cautious as comparing on-premises metrics to published AWS cloud metrics can often result in misleading comparisons. To list an obvious example, Oracle databases report I/O operations per seconds (IOPS) using the default database block size, usually 8 kilobytes, whereas AWS published IOPS metrics for Elastic Block Store SSD tiers are all based on 16 kilobyte block storage operations. Failure to appropriately compare metrics can lead to unsatisfactory performance and cost after a cloud migration.

Migrating Oracle to AWS: Your Complete Guide

Costing

Once licensing and architecture sizing and capacity are understood and accounted for, the topic of costing must be addressed. It's very rare for a cloud migration to occur in a "blank check" scenario. More typically there are requirements that firm projections of costs for the migration and for the ongoing cloud operating expenses should be prepared and approved in advance of a migration project being initiated. This is a complex topic, because of the strong relationship between licensing, cost, and architecture.

Don't treat right-sizing like a one-time thing

Assuming a detailed architecture has been prepared, translating that into projected costs is relatively straightforward. Where House of Brick sees many clients stumble is treating costing as a one-time exercise that can be completed and set aside early in migration planning. In practice, cloud configuration is a constantly moving target. During the preparation phase for a migration the architecture is typically in a state of flux as the plan solidifies.

This continues through the test and execution steps of a migration as migrated application stacks encounter performance or configuration

What happens if your cost controls aren't optimized?

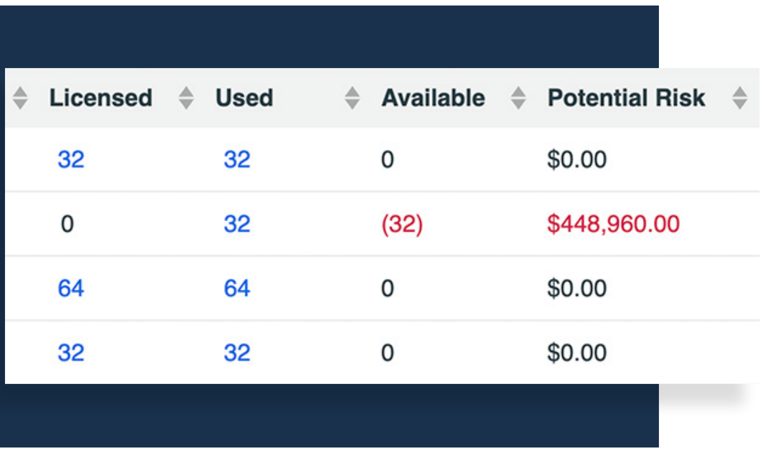
- A successful migration percieved as an organizational failure.
- High costs and significant variance from anticipated costs.

challenges. This continues post-migration as the flexibility of the AWS public cloud environment allows operators to quickly allocate additional capacity in order to meet performance challenges. While all this flexibility is excellent for operations, it makes controlling and accounting for cloud costs becomes a significant challenge. In practice, this problem is intractable until processes are evolved and tools are acquired or developed to manage costs. Anticipated and actual costs should be reviewed frequently, as it's very difficult to determine whether cost variances are justified or not. Increases in provisioned capacity, and thus cost, are sometimes justified as a fast response to challenging operational scenarios caused by welcome growth in business. In other cases such increases are due to human error or mistaken forecasts of necessary provisioned capacity. The drift detection capability of OpsCompass may help alert on cloud inventory and configuration variances that left unchecked could become costly mistakes or compounded problems..

Usage and Cost Checks

OpsCompass License Manager for Oracle automatically monitors your Oracle systems for license compliance, and alerts on non-compliant events.

For more information on OpsCompass, visit opscompass.com



Licensed	Used	Available	Potential Risk
32	32	0	$0.00
0	32	(32)	$448,960.00
64	64	0	$0.00
32	32	0	$0.00

Optimizing cost of database storage deserves extra focus

Beyond just implementing a process to control cloud costs, specific areas of cost impact should be specially considered when trying to forecast the costs of an Oracle database migration to AWS. The main area where House of Brick recommends special focus on costs is database storage. AWS offers a variety of storage tiers through the Elastic Block Store (EBS) service. These storage offerings are substantially the same regardless of whether a database is deployed on EC2 or on RDS. As a general rule databases should be provisioned on SSD tiers of storage of which there are two available in the EBS service. The first, knowns as general purpose SSD storage, is suitable for many workloads. The second, known as provisioned IOPS storage, is much costlier but offers reliable high-speed storage access. Databases, being traditionally performance bound mostly by disk I/O, seem like excellent candidates for the provisioned IOPS tier. This is generally true, especially in latency-sensitive high-performance applications.

The costs of provision IOPS storage can quickly get out of control though. In larger database environments with very high provisioned IOPS levels, the storage costs can be an entire order of magnitude higher for provisioned IOPS storage versus general purpose SSD storage. Having a good understanding of the implications and tradeoffs between better operational performance and higher storage costs should be well understood before costs start to spiral out of control.

Another area specific to the database realm that can have a significant impact on ongoing cloud operational costs is backup storage. Many organizations are used to storing backups in specialized appliances, or on file shares for easy storage and retrievability. Bringing this same paradigm to the cloud can lead to unnecessarily high storage costs. Very large databases that are backed up frequently and retained over long terms in accordance with operational best practices and data retention policies can lead to backup storage footprints many times higher than the actual database size.

In order to control these costs, it is important to utilize the most cost-effective cloud storage available for these backups. In AWS, the S3 object store service offers very low-cost storage suitable for long term retention of data such as database backups. While it sounds in simple in theory to transition backup storage to S3 storage, in practice many organizations struggle with it. Backup tools must be upgraded or reconfigured to support S3, operational procedure documents such as runbooks needs to be updated to account for a different storage media, and in many cases compliance officers or auditors have to be satisfied that the backup storage media is adequately secure and compliant.

It's often tempting to skip all those operational headaches and just attempt to take a lift-and-shift approach to implementing backups in the cloud in the exact same way they were performed in the on-premises environment. While this approach does speed cloud migration, it often leads to large unnecessary costs for backup storage that could have been avoided by better planning and a bit of work up-front to utilize the cloud resources available in the most cost effective manner.

Migrating Oracle to AWS: Your Complete Guide

Database Migration

Having a good plan for sizing, architecture, licensing, and costing is a pre-requisite to a successful cloud migration but doesn't guarantee it. Even with all of those boxes are appropriately checked off there's still a risk that a cloud migration will be perceived as a failure if the process of migration itself introduces too much operational pain. Thus it is important to not only have a good plan on the desired end state for a cloud migration, but also a solid and tested plan for how to execute the migration with only minimal disruptions in service.

For Oracle databases in particular the migration copy time is a prime consideration that tends to drive migration plans. For smaller databases this is relatively straightforward. A downtime is scheduled and the data is copied to the cloud database using traditional tools such as Oracle Datapump or RMAN. Multi-terabyte databases, on the other hand, cannot be copied to the cloud in just a few hours, and for 24/7 mission-critical databases it's often not acceptable to schedule multi-day downtime activities in order to migrate the database to the cloud.

This conflict between acceptable downtime windows and the time required to copy data to the cloud means that more intricate strategies have to be employed to synchronize the data to the intended cloud destination without interrupting service.

For Oracle databases, traditional replication tools such as GoldenGate can be utilized for this purpose. In addition, Amazon offers the Database Migration Service (DMS) which offers similar replication functionality in order to ease customer migrations into a cloud environment. The advantages to migrating using a replication tool are fairly striking in that the downtime to switch over from the on-premises database to the cloud database can be reduced to minutes instead of hours or days.   This convenience does come with an operational cost of course as these tools are somewhat tricky to configure and operate. Organizations that lack staff already trained in implementing these solutions may find they have a difficult learning curve and will require significant investments in staff time to take advantage of all their potential.

Be sure to understand database interdependencies

Another complicating factor is cloud migrations with databases is application adjacency. Mission-critical Oracle databases are often caught in a tangled web of interdependencies with multiple application stacks. This is a special challenge when planning public cloud migrations. It is rarely practical to take on the operational risk of attempting a big bang migration and moving entire datacenter estates into the cloud all at once. It's far more common and prudent to stagger migrations of application stacks so that each one can be given proper attention and validation during the migration process. This common phased approach can often lead to interim states wherein a database is still on premises but an application communicating with it is in the cloud, or vice versa. These interim states are very dangerous as many applications are sensitive to communication latencies.

Any congestion of the network link between the on-premises datacenter and the cloud environment can manifest as apparent database transaction failure which can then lead to degraded or failed application states. Migration plans must take into account the requirements of communication between applications and databases and the dangers of interim states that might stress those dependencies and lead to unwelcome outcomes.

Account for the challenges of running older versions

The final Oracle specific factor to take into account in planning a cloud migration is database versions. Many organizations still run older versions of Oracle database software that are no longer under support. This is surprisingly common as the validation exercises for an Oracle database upgrade can be fairly heavyweight and many organizations wish to defer such disruptions to fragile application stacks. When migration to the cloud these older unsupported versions of Oracle database pose special challenges. They are flatly unsupported by the Amazon RDS service which removes one migration path immediately. Putting older versions of Oracle into the AWS cloud on EC2 is still possible, but may be difficult as older versions of Oracle will often require older operating systems and pose some compatibility problems in a modern environment. Taking care to account for the special challenges of older versions of Oracle database during migration planning is important for avoiding surprises during the execution of the migration plan.

What happens if you don't have a database migration plan?

- Technical issues
- Unplanned interruptions of service

Given all the aforementioned concerns about older Oracle versions, it may be tempting to combine a cloud migration with an upgrade to the latest long-term support Oracle release. In general House of brick endorses keeping Oracle versions current as a best practice. It improves security and compliance posture and generally provides a smoother operational experience. That being said, it is generally not a good idea to attempt to combine a major database upgrade with a cloud migration. Both are complex tasks with many risk areas so combining them just compounds the risks. In some circumstances it may work, but if something goes wrong during a migration it can make it extremely challenging for technical staff to troubleshoot and identify issue root cause. With the entire operating environment migrated to the cloud, combined with a new version of Oracle, even skilled personnel will be challenged to quickly determine the cause of an issue and how to remediate it. House of Brick encourages all organizations to separate database major version upgrades and cloud migration activities.

Migrating Oracle to AWS: Your Complete Guide

Governance

Governance and compliance are important topics to consider carefully when planning a cloud migration. While a large topic in it's own right, in the specific realm of Oracle databases there are several governance and compliance considerations that should be given careful thought.

With any mission-critical enterprise database platform, data security against infiltration or accidental loss is a primary consideration. Generally these are accomplished with encryption and backup strategies to safeguard the data. Both of these areas are often accomplished very differently in the cloud. While using traditional Oracle methods to accomplish both are still supported, in an AWS environment there are often cost friendlier alternatives available. For example, instead of using Oracle's Advanced Security option for database encryption or backup encryption, it's often more cost effective to use storage level encryption in EBS with keys managed by the AWS Key Management Service (KMS).

What are your database governance risks?

- Getting compliance sign-off for your cloud environment
- Licensing liability issued in a future Oracle software audit
- Unplanned interruptions of service

While House of Brick doesn't recommend any one particular implementation as superior, there have been notable issues with clients attempting to translate their data retention and protection requirements to a public cloud environment and then running into difficulties with adequately proving compliance during later audits.

Have a process to ensure license compliance

In addition to data protection concerns, Oracle license compliance is another area where House of Brick sees clients facing significant challenges. It's very easy in a public cloud environment to clone entire Oracle environments or alter their sizing very quickly. While this can streamline operational scenarios and improve team velocity, it also is fraught with the risk of licensing violations. House of Brick has seen many clients inadvertently create licensing liabilities by not having adequate controls around where Oracle is installed, how many CPU licenses are available, and what features are in use.

The nature of Oracle software, which allows almost any feature or option to be utilized regardless of whether it's licensed or not, frequently leads to very agile teams that are iterating rapidly and unintentionally causing Oracle licensing violations. Oracle is historically unsympathetic to claims of accidental violation during software audit, so even unintentional mistakes can lead to large financial liability during audit.

Monitor for database security compliance

We will be releasing the Oracle CIS security benchmark in OpsCompass very soon.

For more information on OpsCompass, visit opscompass.com

Full circle solutions to prevent cloud challenges

300+
Audits Defended

$ 4.2 Billion+
Saved in Oracle License Claims

2,500+
Clients World-Wide

Questions?

CONTACT US NOW!

house of brick

www.houseofbrick.com | +1 877-780-7038