

Best Practices for Running Oracle Database on Amazon Web Services

Abdul Sathar Sait

December 2014



Contents

Abstract	3
Introduction	3
Oracle Licensing Considerations	4
Amazon RDS License Included	4
Bring Your Own License (BYOL)	5
Choosing Between Amazon RDS and Amazon EC2 for Your Oracle Database	6
Architecting for Security and Performance	7
Network Configuration	7
Amazon EC2 Instance Type	9
Database Storage	10
Oracle AMIs	12
Conclusion	12
Further Reading	13

Abstract

Amazon Web Services (AWS) offers you the ability to run your Oracle Database in a hosted IaaS environment. Running Oracle Database on AWS is very similar to running Oracle Database in your data center. To a database administrator or developer, there are no differences between the two environments. However, there are a number of AWS platform considerations relating to security, storage, compute configurations, management, and monitoring that will help you get the best out of your Oracle Database implementation on AWS. This whitepaper provides the best practices for achieving optimal performance, availability, and reliability, and lower total cost of ownership (TCO) while running Oracle Database on AWS. The target audience for this whitepaper includes database administrators, enterprise architects, systems administrators, and developers who would like to run their Oracle Database on AWS.

Introduction

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying Oracle Database on the reliable and secure AWS cloud infrastructure. AWS offers its customers two options for running Oracle Database on AWS:

- Using [Amazon Relational Database Service \(Amazon RDS\) for Oracle](#), which is a managed database service that helps simplify the provisioning and management of Oracle databases. Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud by automating installation, disk provisioning and management, patching, minor version upgrades, failed instance replacement, as well as backup and recovery tasks.

The Multi-AZ feature of Amazon RDS operates two databases in multiple Availability Zones with synchronous replication, thus creating a highly available environment with automatic failover. The push-button scaling feature of Amazon RDS allows you to scale the database instance up and down easily for better cost management and performance. Amazon RDS also comes with a [license-included option](#), which allows pay per use by the hour.

- Running a self-managed Oracle Database directly on Amazon Elastic Compute Cloud (Amazon EC2). This option gives you full control over the setup of the infrastructure and database environment. Running the database on Amazon EC2 is very similar to running the database on your own server. You have full control of the database and have operating system-level access, so you can run monitoring and management agents and use your choice of tools for data replication, backup, and restoration. Furthermore, you have the ability to use every optional module available in Oracle Database. However, this option requires you to set up, configure, manage,

and tune all the components, including Amazon EC2 instances, storage volumes, scalability, networking, and security, based on AWS architecture best practices.

Whether you choose to run a self-managed Oracle Database on Amazon EC2 or the fully managed Amazon RDS for Oracle, following the best practices discussed in this whitepaper will help you get the most out of your Oracle Database implementation on AWS. We'll discuss Oracle licensing options, considerations for choosing Amazon EC2 or Amazon RDS for your Oracle Database implementation, and how to optimize network configuration, instance type, and database storage in your implementation.

Oracle Licensing Considerations

As stated in the [Oracle documentation](#), Oracle fully supports deploying Oracle Database on AWS. Oracle Database licensing on AWS is based on the size of the instance on which the database is installed. For information about Oracle Database licensing, go to [Licensing Oracle Software in the Cloud Computing Environment](#) on the Oracle website.

A few key points:

- The virtual core count of Amazon EC2 instances is considered equal to the physical core count for licensing purposes. To find out the virtual core count of each Amazon EC2 instance type, see [Virtual Cores by Amazon EC2 and RDS DB Instance Type](#).
- Oracle Database Standard Edition may only be licensed on Amazon EC2 instances that have up to 16 virtual cores.
- Oracle Standard Edition One may only be licensed on Amazon EC2 instances that have up to 8 virtual cores.
- For Oracle Database Standard Edition One or Standard Edition, Amazon EC2 instances with 4 or fewer virtual cores are counted as one socket.
- For Oracle Database Enterprise Edition, Amazon EC2 instances with 2 or fewer virtual cores are counted as one socket.

Any discussion of Oracle licensing policies and costs in this whitepaper is for informational purposes only and is based on the information available at the time of publication. Users should consult their own Oracle license agreements for more specific information.

Amazon RDS License Included

You have the option to include the cost of the Oracle Database license in the hourly price of the Amazon RDS service if you use the License Included service model. In this case, you do not need to purchase Oracle licenses separately; the Oracle Database software has been licensed by AWS. License Included per-hour pricing is inclusive of software, underlying hardware resources, and Amazon RDS management capabilities. This is a great option for optimizing license costs, and gives you flexibility when scaling your Amazon RDS instances up or down. You can take advantage of hourly pricing with

no upfront fees or long-term commitments. In addition, you can purchase Amazon RDS Reserved Instances under one-year or three-year reservation terms. With Reserved Instances, you can make a low, one-time, upfront payment for each database instance, and then pay a significantly discounted hourly usage rate.

Note: The hourly license for the License Included model in Amazon RDS is available only for Oracle Standard Edition One. For other editions of Oracle Database on Amazon RDS and any edition of Oracle Database on Amazon EC2, you need to bring your own license (that is, acquire a license from Oracle), as discussed in the following section.

Since you are paying for the Oracle license only for the hours in which you use the Amazon RDS service, the License Included option may help you reduce overall licensing costs for development and testing environments that are active only during business hours. For most businesses, the total business hours per week ($10 \times 5 = 50$ hours) are only about 30% of the total hours in a week ($24 \times 7 = 168$ hours), so this service model could result in considerable savings.

This service model also gives you the flexibility to resize the instance based on your needs, because the license is included in the instance cost. In cases where your regular capacity requirements are much smaller than periodic, predictable spikes, this service model allows you to scale up to absorb the additional capacity needed, and scale down to save on cost. For example, you might have databases that require the performance of a db.m3.large instance for most days of the month except for the last three days. During the last three days of the month, your database might be heavily used due to payroll processing and month-end closing. In this scenario, you can use an Oracle Database on Amazon RDS based on the db.m3.large instance type throughout the month, scale up to db.m3.2xlarge for the last three days, and then scale down again. This could translate to 65% or more cost savings compared to using the db.m3.2xlarge instance for the whole month.

Bring Your Own License (BYOL)

If you already own Oracle Database licenses, you can use the BYOL service model to run your Oracle databases on Amazon RDS. This will result in a lower cost for the Amazon RDS instance, because the cost of the Oracle license isn't included. The BYOL model is designed for customers who prefer to use their existing Oracle Database licenses or purchase new licenses directly from Oracle.

If you want to use Oracle Database Enterprise Edition or Standard Edition with Amazon RDS, or run your own self-managed Oracle Database on Amazon EC2, BYOL is the only supported option.

Oracle License Portability to AWS

Subject to the terms and conditions of the specific license agreement, Oracle licenses may be portable to AWS. In other words, your existing licenses can be transferred for use on AWS. These include:

- Server-based licenses (based on CPUs used)
- Enterprise License Agreements (ELA)
- Unlimited License Agreements (ULA)
- Business Process Outsourcing (BPO) licenses
- Oracle PartnerNetwork (OPN) licenses
- Named User Plus licenses

Additional conditions or limitations (including possible costs) may be applicable for licenses that are ported to AWS. Please check your specific license agreement for additional details and limitations.

Oracle licensing applies similarly to Oracle Database on Amazon RDS and on Amazon EC2 with the exception that hourly licensing is available only on Amazon RDS.

Choosing Between Amazon RDS and Amazon EC2 for Your Oracle Database

Both Amazon RDS and Amazon EC2 offer different advantages for running Oracle Database. Amazon RDS is easier to set up, manage, and maintain than running Oracle Database in Amazon EC2, and lets you focus on other tasks rather than the day-to-day administration of Oracle Database. Alternatively, running Oracle Database in Amazon EC2 gives you more control, flexibility, and choice. Depending on your application and your requirements, you might prefer one over the other.

If you are migrating multiple Oracle databases to AWS, you will find that some of them are a great fit for Amazon RDS while others are better suited to run directly on Amazon EC2. Many AWS customers use a combination of Amazon RDS and Amazon EC2 for their Oracle Database workloads.

Amazon RDS might be a better choice for you if:

- You want to focus on your business and applications, and have AWS take care of the undifferentiated heavy lifting tasks such as provisioning of the database, management of backup and recovery tasks, management of security patches, minor Oracle version upgrades, and storage management.
- You need a highly available database solution and want to take advantage of the push-button, synchronous Multi-AZ replication offered by Amazon RDS, without having to manually set up and maintain a standby database.
- You would like to have synchronous replication to a standby instance for high availability for Oracle Database Standard or Standard One editions.
- You want to pay for the Oracle license as part of the instance cost on an hourly basis instead of making a large upfront investment.

- Your database size is less than 2.5 TB and your maximum IOPS needs are less than 30,000.
- You do not want to manage backups and, most importantly, point-in-time recoveries of your database.
- You would rather focus on high-level tasks, such as performance tuning and schema optimization, rather than the daily administration of the database.
- You want to scale the instance type up and down based on your workload patterns without being concerned about licensing and the complexity involved.

Amazon EC2 might be a better choice for you if:

- You need full control over the database, including Sys/System user access, or you need access at the operating system level.
- Your database size exceeds the 80% of current maximum database size possible in Amazon RDS, which is 3 TB at the time of publication.
- You need to manage the database instances using Oracle Enterprise Manager or other third-party tools.
- You need to use Oracle features or options that are not currently supported by Amazon RDS.
- Your database IOPS needs are higher than 30,000.
- You need a specific Oracle Database version that is not supported by Amazon RDS.

Architecting for Security and Performance

Whether you choose to run Oracle Database on Amazon RDS or Amazon EC2, optimizing every component of the infrastructure will enhance security, performance, and reliability. In the following sections, we'll discuss best practices for optimizing network configuration, instance type, and database storage in an Oracle Database implementation on AWS.

Network Configuration

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS cloud that is dedicated to your account. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways, and security settings.

A *subnet* is a range of IP addresses in your Amazon VPC. You can launch AWS resources into a subnet that you select. Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet.

To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

The following table describes the basic differences between security groups and network ACLs.

Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
Evaluate all rules before deciding whether to allow traffic	Process rules in numerical order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group)

Amazon VPC provides isolation, additional security, and the ability to separate Amazon EC2 instances into subnets, and allows the use of private IP addresses. All of these are important in database implementation. Deploy the Oracle Database instance in a private subnet and allow only application servers within the Amazon VPC, or a Bastion host within the Amazon VPC, to access the database instance. Create appropriate security groups that allow access only to specific IP addresses through the designated ports. These recommendations apply to Oracle Database regardless of whether you're using Amazon RDS or Amazon EC2.

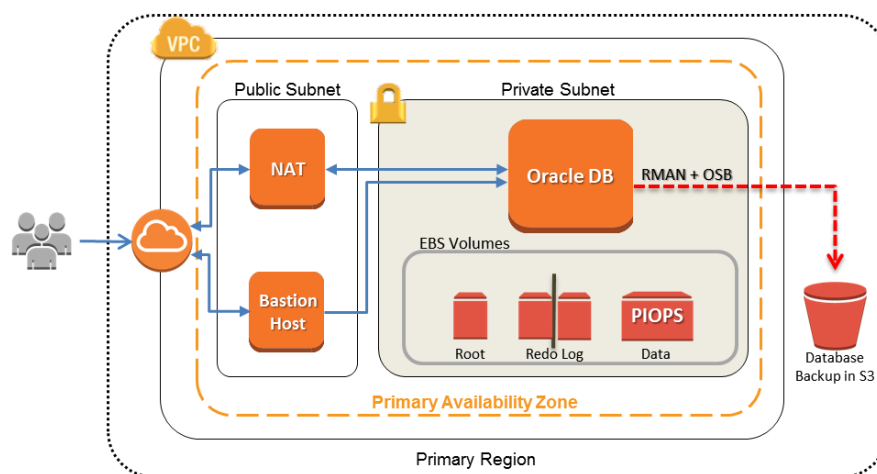


Figure 1: Oracle Database in Private Subnet of an Amazon VPC

Amazon EC2 Instance Type

AWS has a large number of Amazon EC2 instance types available, so you can choose the instance type that best fits your workload. However, not all the available instance types are best suited for running Oracle Database.

If you use Amazon RDS for your Oracle Database, AWS filters out some of the instance types based on best practices, and gives you the following options:

- db.t1.micro
- db.m1.small
- db.m3.medium
- db.m3.large
- db.m3.xlarge
- db.m3.2xlarge

We recommend that you choose db.m3-based Amazon RDS instances for any enterprise database workloads. Your choice of the Amazon RDS instance type should be based on the database workload and the Oracle Database licenses available.

If you're running your self-managed database on Amazon EC2, you have many more choices available for the Amazon EC2 instance type. This is often one of the reasons users opt to run Oracle Database on Amazon EC2 instead of using Amazon RDS. Very small instance types are not suitable, because Oracle Database is resource-intensive when it comes to CPU usage. Instances with a larger memory footprint help improve database performance by providing better caching and a bigger system global area (SGA). We recommend that you choose instances that have a good balance of memory and CPU. Choose the instance type that matches the Oracle Database licenses you are planning to use and the architecture you are planning to implement. Refer to the whitepaper [Advanced Architectures of Oracle Database on AWS](#) for architectures best suited for your business needs.

Oracle Database uses disk storage heavily for read/write operations, so we highly recommend that you use only instances optimized for Amazon Elastic Block Store (Amazon EBS). Amazon EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS. Bandwidth and throughput to the storage subsystem is crucial for good database performance; choose instances with higher network performance for better database performance.

The following instance types are best suited for running Oracle Database on Amazon EC2. Please refer to the [Virtual Core table](#) to determine the virtual core count for each instance type for licensing purposes.

Instance Type	vCPU	Memory	Instance Storage
m3.xlarge	4	15	2 x 40 SSD
m3.2xlarge	8	30	2 x 80 SSD
c3.4xlarge	16	55	2 x 160 SSD
r3.large	2	15.25	1 x 32 SSD
r3.xlarge	4	30.5	1 x 80 SSD
r3.2xlarge	8	61	1 x 160 SSD
r3.4xlarge	16	122	1 x 320 SSD
r3.8xlarge	32	244	2 x 320 SSD
i2.xlarge	4	30.5	1 x 800 SSD
i2.2xlarge	8	61	2 x 800 SSD
i2.4xlarge	16	122	4 x 800 SSD
i2.8xlarge	32	244	8 x 800 SSD

Database Storage

For database storage, you will normally use Amazon EBS. For some very high-performance architectures, you can use instance storage SSDs, but they should be augmented with Amazon EBS storage for reliable persistence; see the [Advanced Architectures for Oracle Database on Amazon EC2](#) whitepaper for details of this architecture.

For high and consistent IOPS and database performance, we highly recommend using General Purpose (GP2) volumes or Provisioned IOPS (PIOPS) volumes. GP2 and PIOPS volumes are available for both Amazon EC2 and Amazon RDS. GP2 can provide up to 3,000 IOPS per volume, and PIOPS can provide up to 4,000 IOPS per volume. GP2 volumes provide an excellent balance of price and performance for most database needs. When your database requires higher IOPS than what GP2 can provide, PIOPS volumes are the right choice.

For PIOPS volumes, you specify an IOPS rate when you create the volume, and Amazon EBS delivers within 10% of the provisioned IOPS performance 99.9% of the time over a given year. A Provisioned IOPS (SSD) volume can range in size from 4 GB to 1 TB, and you can provision up to 4,000 IOPS per volume. The ratio of IOPS provisioned to the volume size requested can be a maximum of 30; for example, to get 3,000 IOPS your volume size should be at least 100 GB.

GP2 volumes are also SSD-based, similar to PIOPS volumes, but the IOPS you get from GP2 volumes can vary from a baseline IOPS up to a maximum burstable 3,000 IOPS

per volume. This works very well for most database workloads, because the IOPS performance needed from the database varies many times during a period of time based on the amount of load and queries being executed.

General Purpose (SSD) volume performance is governed by volume size, which dictates the base performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher base performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your General Purpose (SSD) volume can use to burst large amounts of I/O when more than the base performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its base performance level and the better it performs when more performance is needed.

A good approach is to estimate the amount of IOPS consistently needed for your database, and allocate enough GP2 storage to obtain that many IOPS. Any additional IOPS needed for periodic spikes should be covered by the burst performance based on the available credits. See the whitepaper [Determining the IOPS Needs for Oracle Database on AWS](#) for estimation methods for IOPS needs of your Oracle Database.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. If you notice that your volume performance is frequently limited to the base level (due to an empty I/O credit balance), you should consider using a larger General Purpose (SSD) volume (with a higher base performance level) or switching to a Provisioned IOPS (SSD) volume for workloads that require sustained IOPS performance greater than 3,000 IOPS. See the [Amazon EBS User Guide](#) for additional details on GP2 volumes.

For Amazon RDS, General Purpose (SSD) storage delivers a consistent baseline of 3 IOPS per provisioned GB and provides the ability to burst up to 3,000 IOPS. If you are already using Magnetic Storage for Amazon RDS, you can convert to General Purpose (SSD) storage, but you will encounter a short availability impact when doing so. Using Provisioned IOPS, you can provision up to 3 TB of storage and 30,000 IOPS per database instance. For a workload with 50% writes and 50% reads running on an m2.4xlarge instance, you can realize up to 25,000 IOPS for Oracle Database. However, by provisioning more than this limit, you may be able to achieve lower latency and higher throughput. Your actual realized IOPS may vary from the amount you provisioned based on your database workload, instance type, and database engine choice. Refer to [Factors That Affect Realized IOPS Rates](#) in the *Amazon RDS User Guide*.

For Oracle Database on Amazon EC2, stripe multiple volumes together for more IOPS and larger capacity. You can use multiple Amazon EBS volumes individually for different data files, but striping them together allows better balancing and scalability. Oracle Automatic Storage Management (ASM) can be used for striping. Keep data files, log files, and binaries on separate Amazon EBS volumes, and take snapshots of log file volumes on a regular basis. Choosing an instance type with local SSD storage allows

you to boost the database performance by using Smart Flash Cache (if the operating system is Oracle Linux) and by using local storage for temporary files and table spaces.

Most Oracle Database users take regular hot and cold backups. Cold backups are taken while the database is shut down, whereas hot backups are taken while the database is active. Store your hot and cold backups in Amazon Simple Storage Service (Amazon S3) for high durability and easy access. You can use AWS Storage Gateway or Oracle Secure Backup Cloud Module to directly back up the database to Amazon S3. You can use lifecycle policies in Amazon S3 to move older backups to Amazon Glacier for archiving.

Oracle AMIs

An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need.

Oracle periodically provides official AMIs for some Oracle products, including Oracle Database, on AWS. However, Oracle-provided database AMIs that are available might not always be the latest version. Furthermore, Oracle-supplied AMIs are based on the Oracle Linux operating system, so depending on your operating system of choice, this might not be the best option.

You are not required to use an Oracle-provided AMI to install and use Oracle Database on Amazon EC2. You can start an Amazon EC2 instance with an operating system AMI, and then download and install Oracle Database software from the Oracle website, just as you would do in the case of a physical server. See the whitepaper [Choosing the Operating System for Oracle Workloads on Amazon EC2](#) for recommended operating systems for Oracle workloads on AWS.

Once you have the first environment set up with all the necessary Oracle software, you can create your own custom AMI for subsequent installations. You can also directly launch AMIs from [AWS Marketplace](#). You should closely scrutinize any community AMIs provided by third parties for security and reliability before using them. AWS is not responsible or liable for their security or reliability.

Conclusion

Depending on your usage scenario, you can use Amazon RDS for Oracle Database or run a self-managed Oracle Database on Amazon EC2. Regardless of your choice, following the best practices provided in this paper will help you get the best out of your Oracle Database implementation on AWS.

Further Reading

For additional information about running Oracle workloads on AWS, consult the following resources:

Oracle Database on AWS:

- Advanced Architectures for Oracle Database on Amazon EC2
<http://d0.awsstatic.com/enterprise-marketing/Oracle/AWSAdvancedArchitecturesforOracleDBonEC2.pdf>
- Strategies for Migrating Oracle Database to AWS
<http://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf>
- Choosing the Operating System for Oracle Workloads on Amazon EC2
<http://d0.awsstatic.com/whitepapers/choosing-os-for-oracle-workloads-on-ec2.pdf>
- Determining the IOPS Needs for Oracle Database on AWS
<http://d0.awsstatic.com/whitepapers/determining-iops-needs-for-oracle-database-on-aws.pdf>
- AWS Case Study: Amazon.com Oracle DB Backup to Amazon S3
<http://aws.amazon.com/solutions/case-studies/amazon-oracle/>

Oracle on AWS

- <http://aws.amazon.com/oracle/>
- <http://aws.amazon.com/rds/oracle/>

Oracle on AWS FAQ

- <http://www.oracle.com/technetwork/topics/cloud/faq-098970.html>

Oracle on AWS Test Drives

- <http://aws.amazon.com/solutions/global-solution-providers/oracle/labs/>

Oracle licensing on AWS

- <http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf>

Getting started with Oracle RMAN backups and Amazon S3:

- <http://aws.amazon.com/backup-recovery/getting-started/>

AWS service details

- <http://aws.amazon.com/products/>
- <http://aws.amazon.com/documentation/>
- <http://aws.amazon.com/whitepapers/>

AWS pricing information:

- <http://aws.amazon.com/pricing/>
- <http://calculator.s3.amazonaws.com/index.html>

© 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.