Best Practices for Running Oracle Database on Amazon Web Services

AWS Whitepaper

Best Practices for Running Oracle Database on Amazon Web Services: AWS Whitepaper

Table of Contents

Best Practices for Running Oracle Database on Amazon Web Services 1
Abstract 1
Introduction 1
Oracle Licensing Considerations 3
Amazon RDS License Included 3
Bring Your Own License (BYOL) 4
Oracle License Portability to AWS 4
Choosing Between Amazon RDS and Amazon EC2 for Your Oracle Database 5
Architecting for Security and Performance 6
Network Configuration 6
Amazon EC2 Instance Type 7
Database Storage 8
Backup Storage 9
Amazon S3 10
Amazon Glacier 10
Amazon EFS 10
Amazon EBS Snapshots 10
Management 11
Automation 11
Oracle AMIs 11
AWS EC2 Systems Manager 11
Conclusion 12
Resources 13
Document Details 14
Contributors 14
..... 14
AWS Glossary 15

Best Practices for Running Oracle Database on Amazon Web Services

Publication date: **January 2018** (*Document Details* (p. 14))

Abstract

Amazon Web Services (AWS) offers you the ability to run your Oracle Database in a cloud environment. Running Oracle Database in the AWS Cloud 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 best practices for achieving optimal performance, availability, and reliability, and lowering the 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 easily scale the database instance up or down for better cost management and performance. Amazon RDS also comes with a License Included service model, which allows you to 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. In the fully managed Amazon RDS service, this is all taken care of for you.

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

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, see Licensing Oracle Software in the Cloud Computing Environment on the Oracle website. A few key points to consider are:

- As stated in Amazon EC2 Instance Types page, each vCPU is a hyperthread of an Intel Xeon core except for T2 and m3.medium. For Oracle Enterprise Edition every two vCPUs of hyperthreaded instance equate to a licensing requirement of one Oracle processor license. For non-hyperthreaded instances, each vCPU equate to one Oracle processor license.
- Oracle Database Standard Edition may only be licensed on instances that have up to 16 Amazon vCPUs.
- Oracle Standard Edition One and Standard Edition Two may only be licensed on instances up to 8 Amazon vCPUs.
- For Oracle Database Standard Edition, Standard Edition One, or Standard Edition Two, every four Amazon vCPUs used (rounded up to the nearest multiple of four) equate to a licensing requirement of one socket, which is considered equivalent to an Oracle processor license.

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. For more specific information, users should consult their own Oracle license agreements.

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 includes software, underlying hardware resources, and Amazon RDS management capabilities. This service model optimizes 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 payment up front 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 and Standard Edition Two. For other editions of Oracle Database on Amazon RDS and any edition of Oracle Database on Amazon EC2, you need to use 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 Amazon RDS, 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 x 5 = 50 hours) is only about 30% of the total hours in a week (24 x 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 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 Edition One or Standard Edition Two.
- 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 and IOPS needs are less than the RDS Oracle limits. See the documentation for the current maximum.
- You don't 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 or 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 in Amazon RDS.
- You need to use Oracle features or options that are not currently supported by Amazon RDS. See the documentation for currently supported options in Amazon RDS Oracle
- Your database IOPS needs are higher than the current IOPS limit. See the documentation for the current IOPS limit.
- You need a specific Oracle Database version that is not supported by Amazon RDS. For more information, see Oracle Database Editions.

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

With Amazon Virtual Private Cloud (Amazon VPC), you can 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, security settings, and configuration of route tables and network gateways.

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.

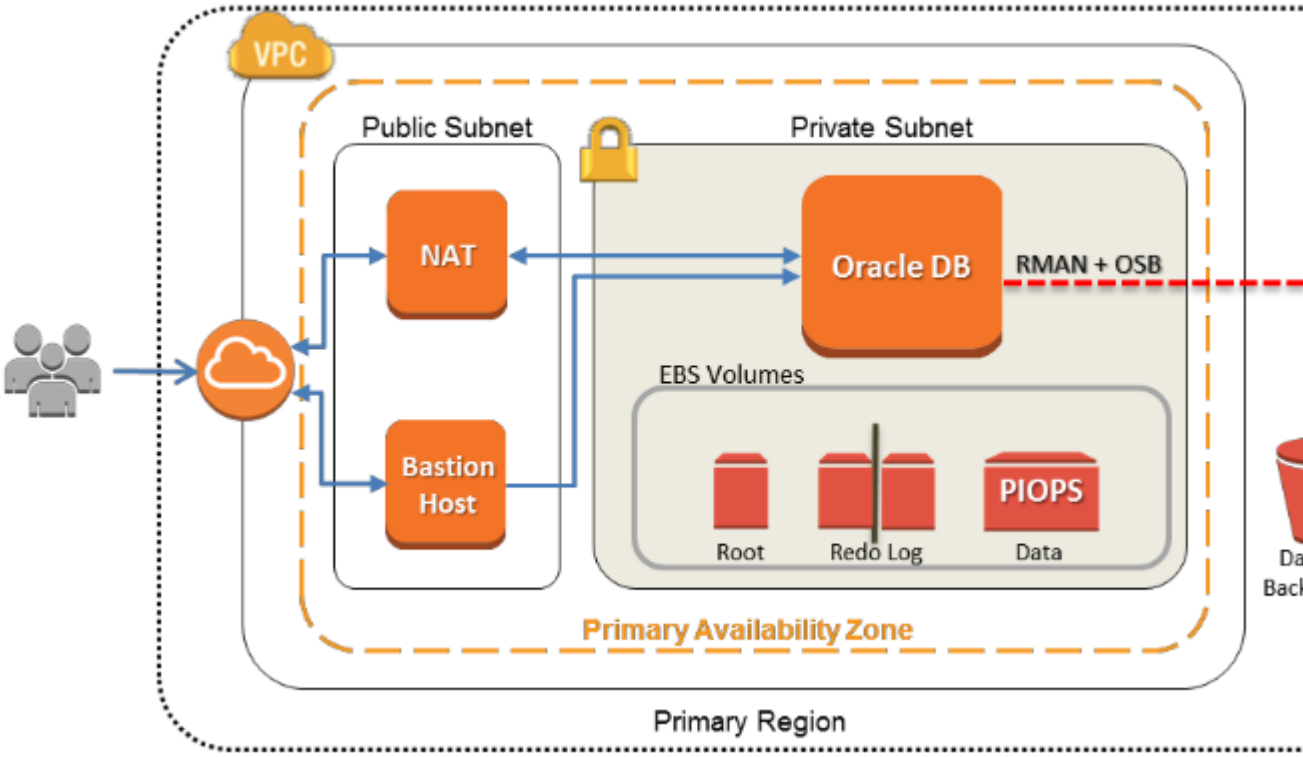
To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACLs).

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
Stateful: Return traffic is automatically allowed, regardless of any rules	Stateless: Return traffic must be explicitly allowed by rules
Evaluates all rules before deciding whether to allow traffic	Processes 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.



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 various options in T-class, M-class and R-class instances. We recommend that you choose db.m-based or r-based Amazon RDS instances for any enterprise database workloads. For the latest information about RDS instances, see Amazon RDS for Oracle Database Pricing. 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. For architectures best suited for your business needs, see the whitepaper Advanced Architectures for Oracle Database on Amazon EC2.

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 families are best suited for running Oracle Database on Amazon EC2.

Instance Family	Features
M family	• EBS-optimized by default at no additional cost • Support for Enhanced Networking • Balance of compute, memory, and network resources
X family	• Lowest price per GiB of RAM • SSD Storage and EBS-optimized by default and at no additional cost • Ability to control processor C-state and P-state configuration
R family	• Optimized for memory-intensive applications • High-frequency Intel Xeon E5-2686 v4 (Broadwell) Processors • DDR4 Memory • Support for Enhanced Networking
I family	• Optimized for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS at a low cost • NVMe SSD ephemeral storage • Support for TRIM • Support for Enhanced Networking

Database Storage

Most users typically use Amazon EBS for database storage. For some very high-performance architectures, you can use instance storage SSDs, but they should be augmented with Amazon EBS storage for reliable persistence. For details about this architecture, see the Advanced Architectures for Oracle Database on Amazon EC2 whitepaper.

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. See the documentation for the latest limits of IOPS per volume for both GP2 and PIOPS volume types. 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. 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.

Similar to PIOPS volumes, GP2 volumes are also SSD-based, 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 load size and the number of 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.

Throughput optimized HDD volumes (st1) offers low-cost HDD volume designed for intensive workloads which require less IOPS but high throughput. Oracle databases used for data warehouses and data analytics purposes can leverage st1 volumes. Any log processing or data staging areas like Oracle external tables or external BLOB storage which require high throughput can leverage st1 volumes. Throughput optimized (st1) volumes can handle max 500 IOPS per volume.

Cold HDD volumes (sc1) are suitable for handling legacy systems which are kept around for the purposes of occasional reference or archive purposes. These systems are accessed less frequently and a few scans are performed per day on the volume.

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. For information about estimation methods you can use to determine the IOPS needs of your Oracle Database, see the Determining the IOPS Needs for Oracle Database on AWS whitepaper.

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 10,000 IOPS. For additional details about GP2 volumes, see the Amazon EBS User Guide.

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 the current maximum storage limit and the maximum IOPS per database instance. Your actual realized IOPS may vary from the amount you provisioned based on your database workload, instance type, and database engine. For more information, see 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 (that is, if the operating system is Oracle Linux) and by using local storage for temporary files and table spaces.

Backup Storage

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. AWS native storage services offer a choice of solutions for your needs.

Amazon S3

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 file interface to directly back up the database to Amazon S3. AWS Storage Gateway file interface provides an NFS mount for S3 buckets. Oracle RMAN backups written into the NFS mount is automatically copied to S3 buckets by the AWS Storage Gateway instance.

Amazon Glacier

Amazon Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup. You can use lifecycle policies in Amazon S3 to move older backups to Amazon Glacier for long-term archiving. Amazon Glacier offers three options for data retrieval with varying access times and costs: Expedited, Standard, and Bulk retrievals. For more information about these options, see Amazon Glacier FAQs.

Amazon EFS

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it. Backups stored in EFS can be shared with NFS options (read/write, read-only) to other EC2 instances. Amazon EFS uses bursting model for EFS performance. Accumulated burst credits give the file system permission to drive throughput above its baseline rate. A file system can drive throughput continuously at its baseline rate. Whenever it's inactive or driving throughput below its baseline rate, the file system accumulates burst credits. Amazon EFS is useful when you have to refresh dev and test databases from production database RMAN backups regularly. Amazon EFS can also be mounted in on-premises data centers when connected to your Amazon VPC with AWS Direct Connect. This option is useful when the source Oracle database is in AWS and other refreshed Oracle databases are in on-premises data centers. Backups stored in EFS can be copied to an S3 bucket using AWS CLI commands. See the documentation for getting started on EFS.

Amazon EBS Snapshots

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data lazily in the background so that you can begin using it immediately. If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. See the documentation on creating and managing EBS snapshots.

Management

Topics

- Automation (p. 11)
- Oracle AMIs (p. 11)
- AWS EC2 Systems Manager (p. 11)

Automation

Oracle database creation and deployment can be automated using AWS CloudFormation templates. For step-by-step instructions deploying an Oracle database environment, see Oracle Database on AWS Quick Start.

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. Oracle-supplied AMIs are based on the Oracle Linux operating system.

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 with a physical server. For recommended operating systems for Oracle workloads on AWS, see the Choosing the Operating System for Oracle Workloads on Amazon EC2.

After 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.

AWS EC2 Systems Manager

AWS EC2 Systems Manager is a collection of capabilities that helps you automate management tasks such as systems inventory, applying operational patches, automatic creation of AMIs, and configuring operating systems and applications at scale. EC2 Systems Manager uses an SSM (System State Management) Agent to collect inventory, state information within the EC2 instance, and run patch commands. Patch Manager integrates with AWS Identity and Access Management (IAM), AWS CloudTrail, and Amazon CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage.

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, by following the best practices provided in this paper you can get the best out of your Oracle Database implementation on AWS.

Resources

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

Oracle Database on AWS

- Oracle and Amazon Web Services at https://aws.amazon.com/oracle/
- Amazon RDS for Oracle Database at https://aws.amazon.com/rds/oracle/
- Advanced Architectures for Oracle Database on Amazon EC2 whitepaper at https://d0.awsstatic.com/enterprise-marketing/Oracle/AWSAdvancedArchitecturesforOracleDBonEC2.pdf
- Strategies for Migrating Oracle Databases to AWS whitepaper at https://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf
- Choosing the Operating System for Oracle Workloads on Amazon EC2 whitepaper at http://d0.awsstatic.com/whitepapers/choosing-os-for-oracle-workloads-on-ec2.pdf
- Determining the IOPS Needs for Oracle Database on AWS whitepaper at https://d0.awsstatic.com/whitepapers/determining-iops-needs-for-oracle-database-on-aws.pdf
- Oracle Database on AWS Quick Start at https://aws.amazon.com/quickstart/architecture/oracle-database/
- Oracle on AWS Test Drive at https://aws.amazon.com/solutions/global-solution-providers/oracle/labs/
- Getting Started: Backup Oracle databases directly to AWS with Oracle RMAN at https://aws.amazon.com/backup-recovery/getting-started/

Oracle Documentation

- Licensing at http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf
- Support at https://support.oracle.com/epmos/faces/DocContentDisplay?id=2174134.1

AWS Service and Pricing Details

- AWS Cloud Products at https://aws.amazon.com/products/
- AWS Documentation at https://aws.amazon.com/documentation/
- AWS Whitepapers at https://aws.amazon.com/whitepapers/
- AWS Architecture Monthly at https://aws.amazon.com/whitepapers/kindle/
- AWS Pricing https://aws.amazon.com/pricing/
- AWS Simple Monthly Calculator http://calculator.s3.amazonaws.com/index.html

Document Details

Contributors

The following individuals and organizations contributed to this document:

- Jayaraman Vellore Sampathkumar, AWS Oracle Solution Architect, Amazon Web Services
- Jinyoung Jung, Product Manager, Amazon Web Services
- Abdul Sathar Sait, Amazon Web Services

Date	Description
January 2018	Updated with new EC2 instance types, AWS EFS, and EC2 Systems Manager service.
December 2014	First publication

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.