Choosing the Operating System for Oracle Workloads on Amazon EC2

Published June 2014

Updated July 19, 2021

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. 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.

Contents

Abstract

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying enterprise applications in a highly secure, reliable, available, and cost-effective manner. The AWS Cloud is an excellent platform to run business-critical Oracle workloads in an efficient way. This whitepaper discusses the operating system choices that are best suited for running Oracle workloads on AWS. The target audience for this whitepaper includes enterprise architects, database administrators, IT managers, and developers who want to migrate Oracle workloads to AWS.

Introduction

Oracle software works well on Amazon Web Services (AWS), and many enterprises run their critical Oracle workloads on AWS for both production systems and non-production systems. These applications can benefit from the many features of the AWS Cloud, like scriptable infrastructure, instant provisioning and de-provisioning, scalability, elasticity, usage-based billing, and the ability to support a wide variety of operating systems. Whether you are migrating your existing Oracle environments to AWS or implementing new Oracle applications on AWS, choosing the operating system on which these applications will run is a crucial decision. We highly recommend that you choose an Oracle-certified operating system to run Oracle software on AWS, whether you are running Oracle Database, Oracle enterprise applications, or Oracle middleware. You can use the following Oracle-certified operating systems on AWS:

- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server
- Oracle Linux
- Microsoft Windows Server

> **Note:** Only Oracle can make definitive statements about what products are considered certified. For details, see Oracle's My Oracle Support website or ask your Oracle sales representative.

You can use any one of the four operating systems for all of your Oracle workloads, or you can use a combination of them as needed. For example, to implement Oracle Siebel, you can run Oracle Database on RHEL while running web servers and application servers on Microsoft Windows Server.

All four of these operating systems are well suited for enterprise workloads, but each of them has features and capabilities that the others do not have. Knowing the differences will help you make the right decision about what is best for your environments.

If you migrate an existing Oracle environment on Intel platform to AWS, and if that environment currently uses one of the four operating systems in the preceding list, then it might be best to choose the same operating system on AWS, to keep any compatibility risks to a minimum. However, it also might be worthwhile to evaluate other options. If you migrate from a non-Intel platform or implement a completely new

environment on AWS, then you should carefully evaluate the operating systems before you choose the one that is best for your environment.

Oracle AMIs

An Amazon Machine Image (AMI) is a special type of pre-configured operating system and virtual application software that is used to create a virtual machine on Amazon Elastic Compute Cloud (Amazon EC2). The AMI serves as the basic unit of deployment for services delivered using Amazon EC2. The 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.

An AMI includes the following:

- A template for the root volume for the instance
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

There are no official AMIs available for most Oracle products. In addition, the AMIs that are available might not always be the latest version. Even when there are latest versions of the AMIs available, they will be based on the Oracle Linux operating system, so depending on your operating system of choice, this might not be the best option.

You do not need an Oracle-provided AMI to install and use Oracle products on AWS. You can start an Amazon EC2 instance with an operating system AMI, and then download and install Oracle software from the Oracle website, just as you would do in the case of a physical server. You can use any one of the four operating systems discussed in the preceding section for this purpose.

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.

AMIs use one of two types of virtualization:

- Paravirtual (PV)

- Hardware Virtual Machine (HVM)

The main difference between PV and HVM AMIs is the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

Note: For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch new instances. For more information on current generation instance types, see Amazon EC2 Instances.

Operating systems and Oracle licensing

On AWS, the operating system you use does not affect Oracle licensing. The number of Oracle licenses you need to run your Oracle workloads on AWS will be the same no matter which operating system you choose.

As currently advised by Oracle, the key factor that affects Oracle licensing on AWS for Amazon EC2 and RDS is that you should count two vCPUs as equivalent to one Oracle Processor license if hyper-threading is enabled, and one vCPU as equivalent to one Oracle Processor license if hyper-threading is not enabled. When counting Oracle Processor license requirements in AWS Cloud Environments, the Oracle Processor Core Factor Table is not applicable. You can consult Oracle's Licensing Oracle Software in the Cloud Computing Environment to understand how Oracle licensing applies to AWS. To find out the physical core count of each Amazon EC2 instance type, see Physical Cores by Amazon EC2 Instance Type.

Oracle-certified operating systems

This section provides information about the four operating systems that are certified by Oracle and recommended for use with AWS.

> **Note:** It is possible to run Oracle products on non-certified operating systems, but for the best performance and supportability, we recommend that you use an Oracle-certified operating system for use on AWS.

Red Hat Enterprise Linux

A large number of enterprises of all sizes use Red Hat Enterprise Linux (RHEL) to deploy Oracle workloads. RHEL is a great choice for any Oracle workloads on AWS.

AWS and Red Hat have teamed to offer RHEL on Amazon EC2, providing a complete, enterprise-class computing environment with the simplicity and scalability of AWS. Red Hat maintains the base RHEL images for Amazon EC2. As an AWS customer, you will receive updates at the same time that updates are made available from Red Hat, so your computing environment remains reliable and your RHEL-certified applications maintain their supportability. For additional information about RHEL on AWS, see Red Hat on AWS.

RHEL is available for all Amazon EC2 instance types on AWS, including HVM instances. On HVM instances, RHEL supports HugePages, which can especially enhance the performance of Oracle Database. HugePages is a Linux feature that makes it possible for the operating system to support very large memory pages. On AWS, you can use HugePages only on HVM instances. For more information about HVM instances on AWS, see Linux AMI virtualization types.

> **Important:** A special feature in RHEL named Transparent HugePages (THP) is not compatible with Oracle Database and should be disabled for best performance.

RHEL on AWS Pricing

AWS customers can quickly deploy and scale compute resources, according to their business needs, with flexible purchase options for RHEL and RHEL with High availability:

- **Pay-as-you-go.** Provision resources on-demand, as computing needs grow, without long-term commitments or upfront costs.
- **Reserved Instances.** Lower cost further by purchasing compute resources with a one-time, upfront payment.
- **Bring existing subscription**. Customers with Red Hat Enterprise Linux Premium subscriptions can use Red Hat Cloud Access to move subscriptions to Amazon EC2.

SUSE Linux Enterprise Server

SUSE Linux Enterprise Server (SLES) is an operating system of choice for Oracle workloads in many large Oracle deployments. SLES is a great choice to run Oracle workloads on AWS as well. SUSE maintains the base SLES images for Amazon EC2,

and as an AWS customer you will receive updates at the same time that updates are made available from SUSE.

SLES also is available for all Amazon EC2 instance types on AWS, including HVM instances. On HVM instances, SLES supports HugePages, which can especially enhance the performance of Oracle Database.

You can launch an SLES-based Amazon EC2 instance directly from the AWS console or from the AWS Marketplace. For additional information about SLES on AWS, see SUSE and AWS.

SUSE on AWS Pricing

SUSE on AWS is available with the on-demand and Bring Your Own Subscription (BYOS) subscription model. AWS on-demand SUSE subscriptions are offered at either a flat, hourly rate with no commitment or through a one-time, upfront payment. Both purchase options include Amazon EC2 compute charges and SUSE subscription charges. Amazon tracks and bills customers who purchase SUSE Linux Enterprise Server (SLES) or SUSE Linux Enterprise Server for SAP Applications (SLES for SAP) subscriptions through AWS. In BYOS image customers use existing products purchased from SUSE on a BYOS basis with images available as a Community AMI.

Oracle Linux

As the operating system Oracle uses to build and test their products, Oracle Linux is an excellent choice for running Oracle workloads on AWS. Oracle Linux EC2 instances can be launched using an Amazon Machine Image (AMI) available in the AWS Marketplace or as a Community AMI. You can also bring your own Oracle Linux AMI or existing Oracle Linux license to AWS.

Unlike the other three Linux operating systems discussed here, Oracle Linux has no cost for licensing, making it the lowest cost option. You can purchase support directly from Oracle, but support is not necessary to get updates and patches. Oracle provides public **yum** repositories to download updates and patches, even for customers who have not subscribed to support. For customers who have subscribed to support, Oracle Linux allows zero-downtime updates, which can be useful for mission-critical applications.

Oracle Linux has a special feature named Database Smart Flash Cache that is not available in any of the other operating systems discussed here. Database Smart Flash Cache allows the database buffer cache to expand beyond the system global area

(SGA) in main memory to a second-level cache on flash memory. Making use of Database Smart Flash Cache for Oracle Database can substantially increase the database performance. This is a good feature to use with Amazon EC2 instances that have a large amount of SSD instance storage.

Microsoft Windows Server

Microsoft Windows Server versions 2012, 2012 R2, 2016, and 2019 are available on Amazon EC2 as Oracle-certified operating systems to run Oracle workloads. Microsoft Windows Server is an excellent choice for many Oracle workloads, especially for running enterprise applications like PeopleSoft, Siebel, and JD Edwards. Microsoft Windows is available on all types of Amazon EC2 instances including HVM, making it a good choice for Amazon EC2 instance types with large memory configurations.

To access and launch all Microsoft Windows AMIs, see Windows AMIs. Microsoft Windows on Amazon EC2 is available with the managed service model, where AWS takes on all the burdens of acquiring Microsoft Windows licenses to use in the Amazon EC2 service. The Microsoft Windows license tends to be more expensive than the other three Oracle-certified operating systems for the same instance type.

Conclusion

We recommend that you choose one of the four operating systems discussed in this whitepaper for any of your Oracle environments on AWS so that your Oracle workloads run on an Oracle-certified operating system. You can use any one of the four operating systems for all your Oracle workloads, or you can use a combination of them as needed. Your choice typically will depend on familiarity, type of workload, instance choice, and cost preference.

Contributors

Contributors to this document include:

- Vuyisa Maswana, Solutions Architect, Amazon Web Services
- Abdul Sathar Sait, Amazon Web Services

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
- Strategies for Migrating Oracle Database to AWS
- Determining the IOPS Needs for Oracle Database on AWS
- Best Practices for Running Oracle Database on AWS

Oracle on AWS

- Oracle and Amazon Web Services
- Amazon RDS for Oracle

AWS service details

- AWS Cloud Products
- AWS Documentation
- AWS Whitepapers & Guides

AWS pricing information:

- AWS Pricing
- AWS Pricing Calculator

Document history

Date	Description
July 19, 2021	Updated for latest service changes and technologies.
December 2014	First publication