

Advanced Architectures for Oracle Database on Amazon EC2

*Abdul Sathar Sait
Jinyoung Jung
Amazon Web Services*

*November 2014
Last update: April 2016*



Contents

Abstract	2
Introduction	3
Oracle Database Editions and License Implications for AWS	3
Oracle License Portability to AWS	4
Oracle Database AMIs	5
Architecting for Security and Performance	5
Network Deployment	5
Amazon EC2 Instance Type Selection	5
Storage	6
Advanced Architectures	7
Standard Architecture	8
Enterprise-Class Architecture	8
Large Enterprise-Class Architecture	10
High-Performance Architecture	11
Oracle RAC Architecture	12
Further Reading	14

Abstract

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying Oracle Database on the reliable and secure AWS cloud infrastructure. You can use the inherent qualities of the AWS cloud, such as its instant provisioning, scalability, elasticity, and geographic footprint, to architect a highly resilient and performant database environment for demanding business applications. This whitepaper provides information on how to build advanced Oracle Database architectures in the AWS cloud.

Introduction

AWS customers can run Oracle Database on Amazon Relational Database Service (Amazon RDS), the fully managed database service in the cloud, and also directly on Amazon Elastic Compute Cloud (Amazon EC2). Each of these options offers distinct advantages.

Amazon RDS is a fully managed service that makes provisioning and management of Oracle databases very easy. Amazon RDS makes it simple 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, backup, and recovery. The Multi-AZ (Availability Zone) feature of Amazon RDS operates two databases with synchronous replication, creating a highly available environment with automatic failover. The push-button scaling feature of Amazon RDS allows you to easily scale up and scale down an instance for better cost management and performance. Amazon RDS also comes with a license-included option, allowing you to pay for use by the hour. Amazon RDS for Oracle is a great option for many enterprise workloads, from development and test environments, to production systems that use enterprise applications (such as PeopleSoft, JD Edwards, and Seibel), to custom-built applications that use Oracle Database on the back end.

Amazon EC2, in contrast, allows you full control over setup of the entire infrastructure and database environment when you run Oracle Database directly on it. Running a database on Amazon EC2 is very similar to running a database on your own server. You can have full control of the database and access at the operating system level. This control and access allows you to run monitoring and management agents and use your choice of tools for data replication, backup, and restoration. It also lets you use additional options available with Oracle Database Enterprise Edition.

Oracle Database Editions and License Implications for AWS

Oracle Database is available in four editions. As of the date of this whitepaper, Oracle licensing policies allow all four editions to be used on AWS, but users should confirm the specifics of their license agreement. The Express, Standard One, and Standard editions each have certain limitations compared to Enterprise Edition. There can also be a substantial price difference among the licenses for the different editions.

- Express Edition is not usually used by businesses for production purposes, due to its size and processor limitations.
- Standard One Edition and Standard Edition both have a limitation on the number of processors that can be used and the additional options and features available,

such as partitioning, online index rebuild, parallel query/transaction, advanced security, advanced compression, and so on. These editions are equivalent to Enterprise Edition in all other respects.

- Enterprise Edition does not have the limitations of the other editions, has all Oracle Database options available to it, and is widely used in enterprise production systems.

For information on Oracle Database licensing, go to the [Licensing Oracle Software in the Cloud Computing Environment](#) document 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, refer to the [Amazon EC2 virtual core web page](#).
- Oracle Database Standard Edition may only be licensed on Amazon EC2 instances up to 16 virtual cores.
- Oracle Standard Edition One and Standard Edition Two may only be licensed on Amazon EC2 instances up to 8 virtual cores.
- For Standard Edition, Standard Edition One, or Standard Edition Two, Amazon EC2 instances with 4 or less virtual cores are counted as one socket.
- For Enterprise Edition, Amazon EC2 instances with 2 or less 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 writing. Users should consult their own Oracle license agreement for more specific information.

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. Users should check their specific license agreement for additional details and limitations.

Oracle Database AMIs

Oracle has periodically provided official Amazon Machine Images (AMIs) for Oracle Database on AWS. However, the AMIs available might not be for the version of Oracle Database that you need. Oracle AMIs are provided only for your convenience; you actually don't need an Oracle AMI to install and use Oracle Database on AWS. If you don't find an AMI with the latest database version, 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. An AMI of any one of the four Oracle-certified operating systems can be used for this purpose. Oracle-certified operating systems available on AWS are Red Hat Enterprise Linux, Oracle Linux, SUSE Linux Enterprise Server, and Microsoft Windows Server.

Once you have the first environment set up with all the Oracle binary files, you can create your own custom AMI for subsequent installations. Oracle Database AMIs created by third parties are also available on AWS Marketplace, which can be directly launched as needed. Any community AMIs provided by third parties should be closely scrutinized for security and reliability before use, and AWS is not responsible or liable for their security or reliability.

Architecting for Security and Performance

Optimizing every component of your database infrastructure will enhance security, performance, and reliability. Let's look at each of the components that will be used in an Oracle Database implementation on Amazon EC2 from that point of view.

Network Deployment

AWS users are encouraged to use an Amazon Virtual Private Cloud (Amazon VPC) for the database instances. Using an Amazon VPC provides isolation, additional security, the ability to separate Amazon EC2 instances into subnets, and the ability to use private IP addresses. All of these are important when it comes to database implementation. Deploy the Oracle Database instance in a private subnet and allow only application servers within the VPC or a Bastion host within the VPC to access the database instance. In addition, create appropriate security groups that allow access only to specific IP addresses through the designated ports.

Amazon EC2 Instance Type Selection

It is very important to choose the right Amazon EC2 instance type for your database. AWS makes many different instance types available to allow customers to choose the

right type of instance for their workload. Very small instance types are not suitable for Oracle databases, because an Oracle database is resource-intensive when it comes to CPU usage.

Instances with a larger memory footprint improve database performance by providing better caching and bigger system global area (SGA). Thus, it's a good idea to choose instances that have a good balance of memory and CPU.

Oracle Database heavily uses disk storage for read/write operations, so we highly recommend using only Amazon EC2 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.

Storage

For database storage, AWS users are encouraged to use Amazon EBS. For high and consistent IOPS, we highly recommend using General Purpose (GP2) volumes or Provisioned IOPS (PIOPS) volumes. GP2 can provide up to 10,000 IOPS per volume, and PIOPS can provide up to 20,000 IOPS per volume. GP2 volumes provide an excellent balance of price and performance for most database needs. When very high IOPS is required, PIOPS volumes are the right choice.

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 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 done 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. Amazon Storage Gateway or Oracle Secure Backup Cloud Module can be used to directly back up the database to Amazon S3. Lifecycle policies can be applied to the backups in Amazon S3 to move older backups to Amazon Glacier for archiving.

Advanced Architectures

How organizations use Oracle Database varies heavily from implementation to implementation. In some cases, a database is used for online transaction processing (OLTP) loads with very high performance requirements. In other cases, a database is used as part of business-decision support systems that use very large databases with high reliability requirements and less stringent performance expectations. In another scenario, a moderate-sized database runs business applications where high availability and load distribution are the most critical criteria. A fourth scenario combines all the requirements mentioned and includes reliability, high availability, and high performance, as well as scalability and load distribution. Then there are small databases used throughout enterprises for various repositories, custom applications, and reporting. These scenarios constitute the vast majority of use cases for Oracle Database in most businesses. In the rest of the whitepaper, we will discuss five architectures for Oracle Database implementation on AWS that will fit very well with the five scenarios discussed above.

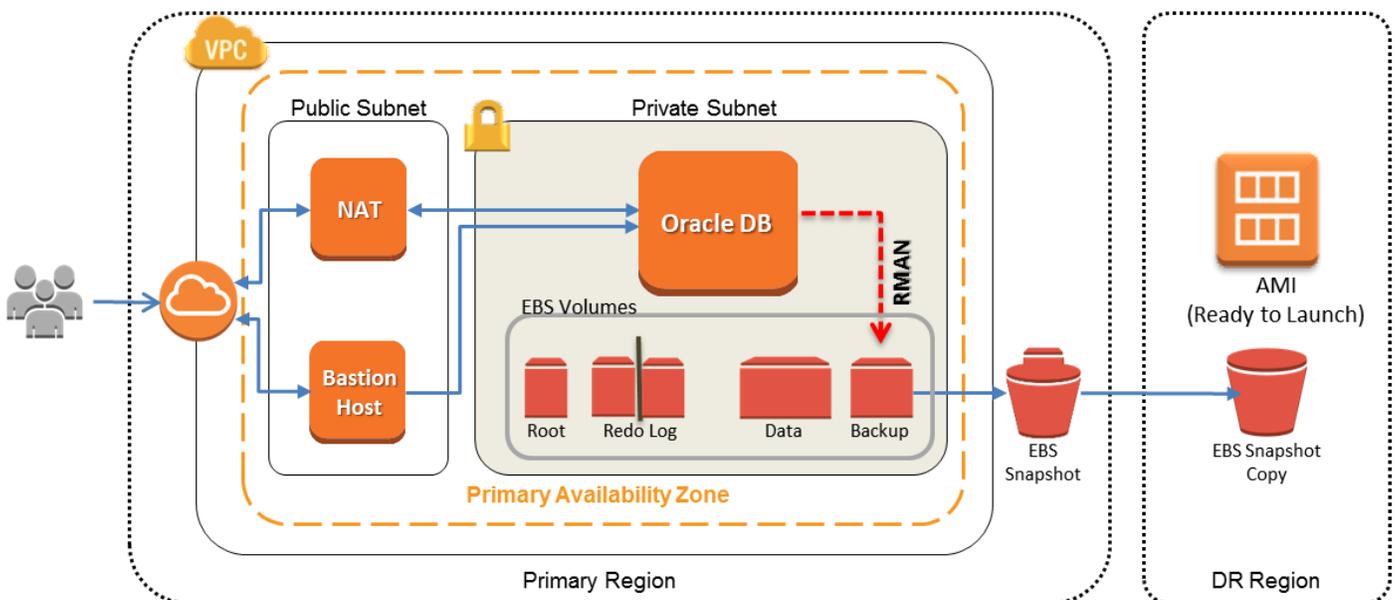
All the architectures discussed here include a disaster recovery (DR) component. Because a database is a crucial piece of any enterprise business system, it is important to have protection to make sure data is safe and recoverable, especially when unexpected things happen. Each architecture discussed here uses a distinct disaster recovery pattern based on the theme of the overall architecture. Depending on your objectives for recovery time and recovery point, you can mix and match the primary database and disaster recovery architectures to fit your needs. For details on these disaster recovery patterns, see the whitepaper [Using Amazon Web Services for Disaster Recovery](#).

The following discussion puts the architectures in order of increasing levels of capability, complexity, and cost. With each of the following architectures, AWS users may employ additional tools such as reserved instances and volume discounts to optimize the cost of your implementation.

Standard Architecture

The standard architecture shown below is a general-purpose database deployment with a good balance of cost and performance, optimized for lower overall cost. This architecture is adequate for many software development and test environments, small application deployments in production environments, custom projects, and reporting systems. You can perform data backup either by using EBS snapshots to Amazon S3 or using Oracle Recovery Manager (RMAN) and Oracle Secure Backup Cloud Module (OSB). Although this architecture is deployed as a simple, small design, it is highly scalable and can be migrated to another of the architectures described below as requirements change.

We recommend this architecture for database sizes up to 1 TB with up to a maximum of 3000 IOPS. Although this architecture is the simplest one, a very large percentage of Oracle databases fall into this category, because the number of databases used for development and test and small projects in any enterprise comes to a sizeable number.



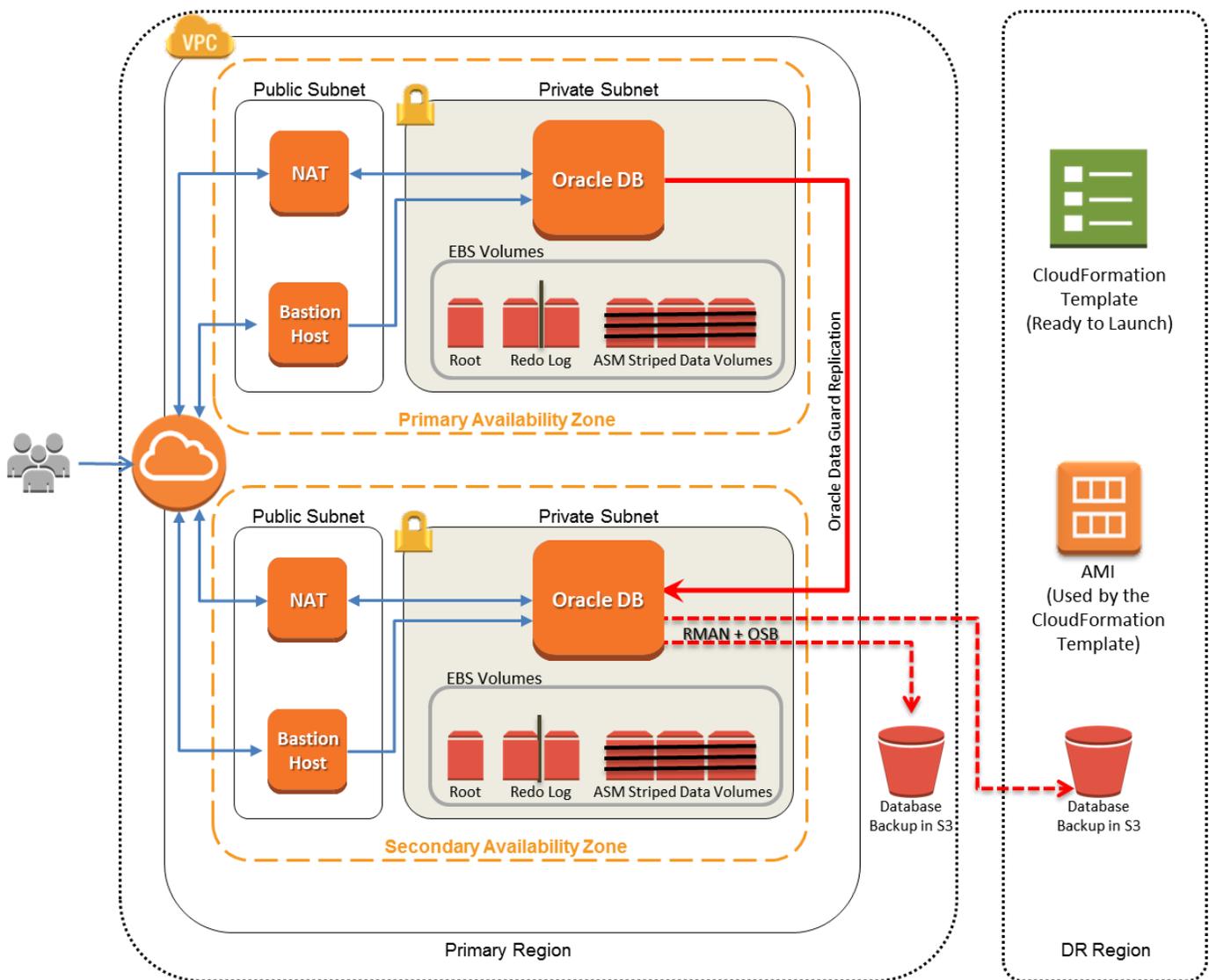
Enterprise-Class Architecture

For many enterprises, availability and reliability of the database on which their business applications run is crucial, because database downtime can result in lost business and in compliance and policy violations. The enterprise-class architecture shown below puts emphasis on high availability and reliability. It has an excellent balance of performance, high availability, and fault tolerance. This architecture can be used for large databases for enterprise applications that need enterprise-class performance and reliability. Based on your needs, the size of the Amazon EC2 instance chosen to run this architecture can vary, but bigger instances are recommended. Because Oracle Data Guard is used for replication, the Enterprise Edition license is required. However, this architecture can also

be modified to use third-party tools that work with Standard Edition and Standard One Edition to provide the same functionality as Oracle Data Guard.

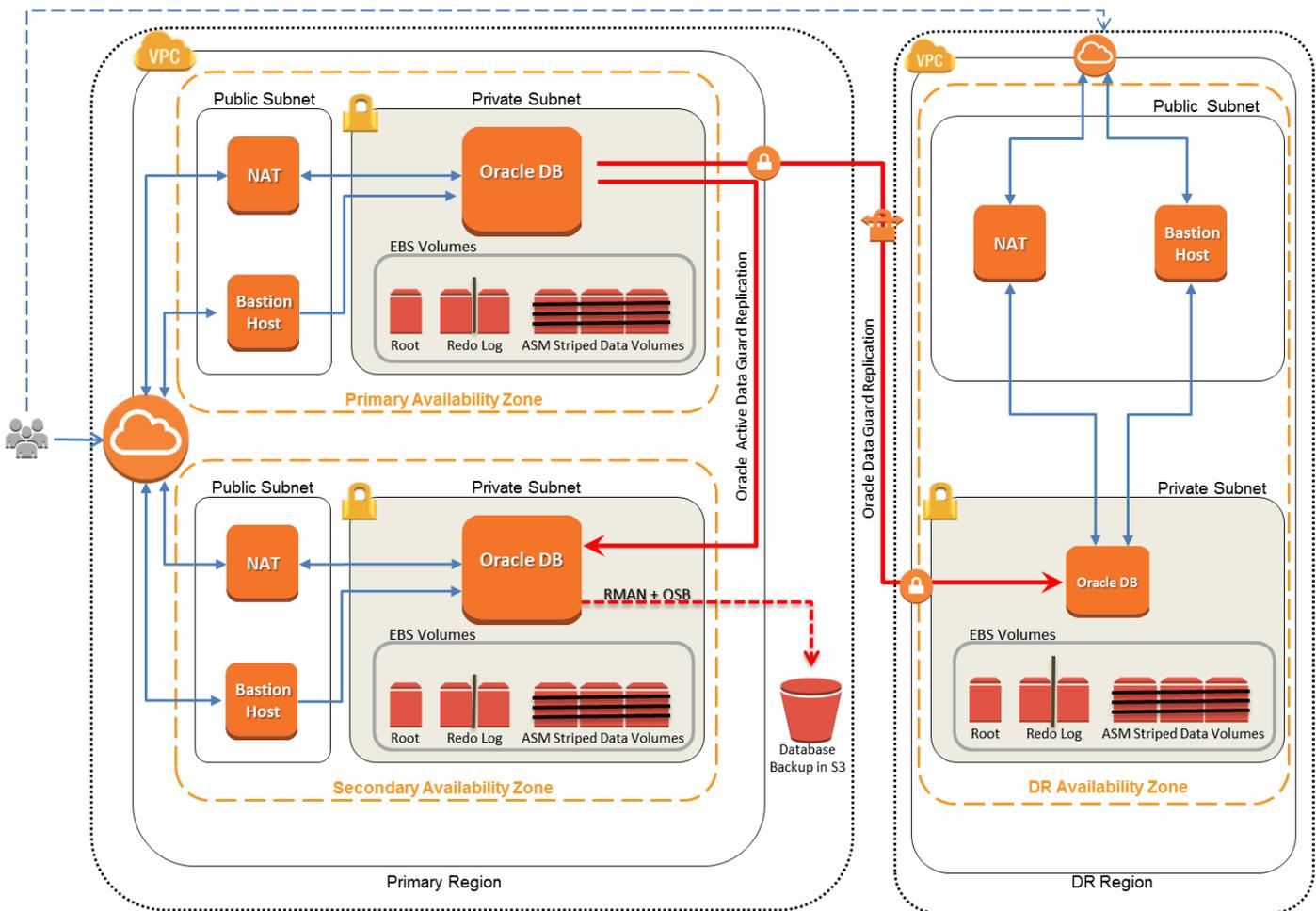
This architecture differs in many key ways from the one we discussed previously. One very important difference is that it uses multiple GP2 EBS volumes for the data files. Though it is possible to add and mount multiple EBS volumes separately to use individually for data files, we recommend striping across the EBS volumes for much higher performance through cumulative IOPS.

Here, the database is backed up to Amazon S3 using RMAN and OSB. DR is implemented using a “backup and restore” pattern to keep the cost lower.



Large Enterprise-Class Architecture

The large enterprise-class architecture builds on the previous enterprise-class one, distributing the load across multiple Availability Zones by using Oracle Active Data Guard. The emphasis in this design is on enterprise-level reliability, availability, and performance. This architecture is targeted at larger enterprise workloads with big databases and heavier processing requirements. In this case, we highly recommend that you use larger Amazon EC2 instances and striped Amazon EBS PIOPS volumes for better performance. Replicating with Active Data Guard means you can use the replicated standby instances for read-only loads, allowing some load distribution. This architecture also uses a “pilot light” disaster recovery pattern, ready to scale up at a moment’s notice, for quicker recovery in a disaster recovery event.

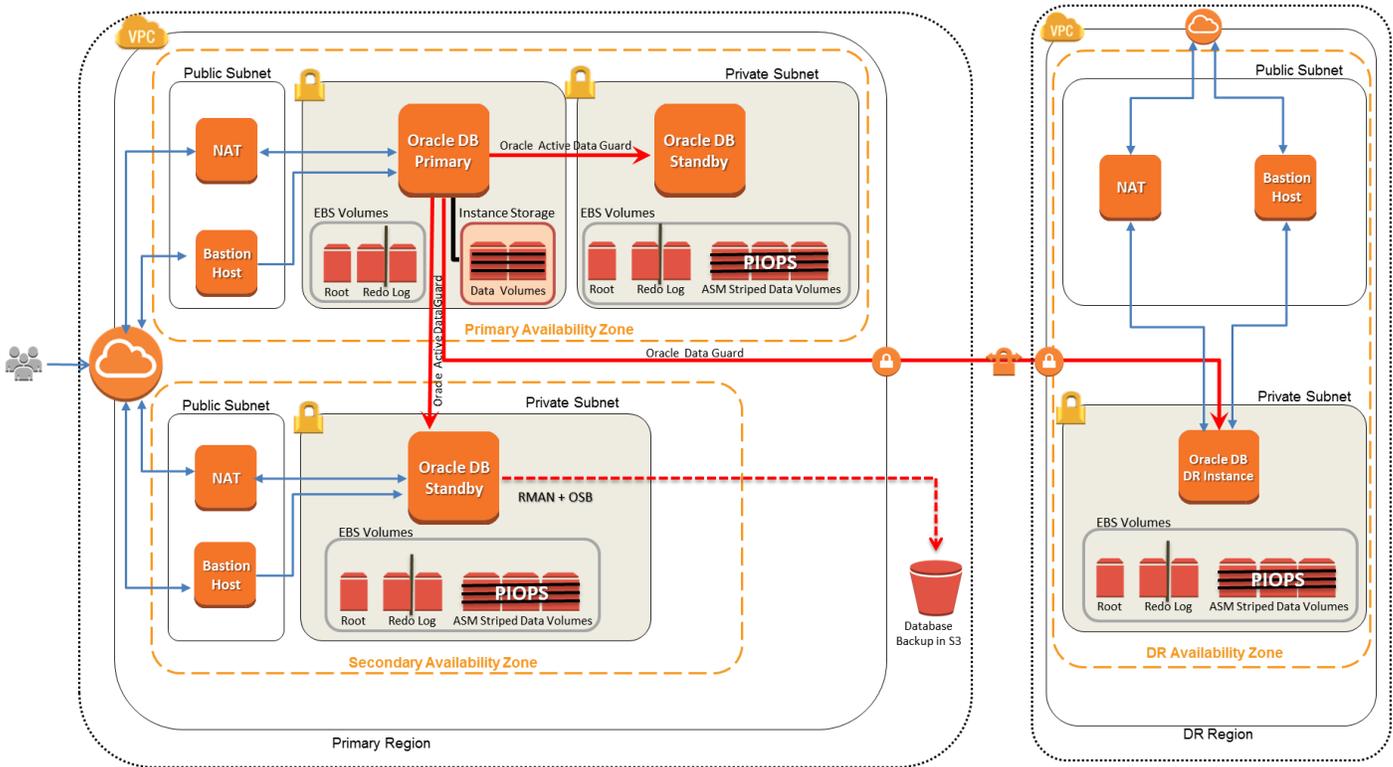


High-Performance Architecture

The high-performance architecture is designed primarily for very high I/O performance. It leverages local (ephemeral) SSD drives to take advantage of very high IOPS and avoid network latency. Under certain conditions, AWS users may be able to achieve up to 340,000 IOPS and very high throughput using this architecture.

This architecture uses ephemeral instance storage for the database; standby instance storage is based on EBS volumes for data safety and reliability. This architecture also performs replication to another instance in the same or a different Availability Zone using Oracle Active Data Guard. If the primary instance fails, this architecture rapidly switches over to the second instance in the same Availability Zone, making the database continuously available and preventing data loss. In a case where either the primary Availability Zone fails or both the primary and secondary instances fail, then this architecture switches over to a third instance in a secondary Availability Zone, thus keeping the database available and preventing data loss. Use of Active Data Guard for replication allows the second and third instances to be simultaneously used for read-only workloads, facilitating load distribution.

The maximum size of the database possible using this architecture is limited to the maximum size of SSD-based instance storage available for that particular instance type. When you consider database size for this architecture, also consider storage for log files and for a flash recovery area. Any instance type with a good balance of CPU power and memory that has local SSDs can be used to implement this architecture. However, the instance type best suited for this architecture is the I2 group of Amazon EC2 instances.

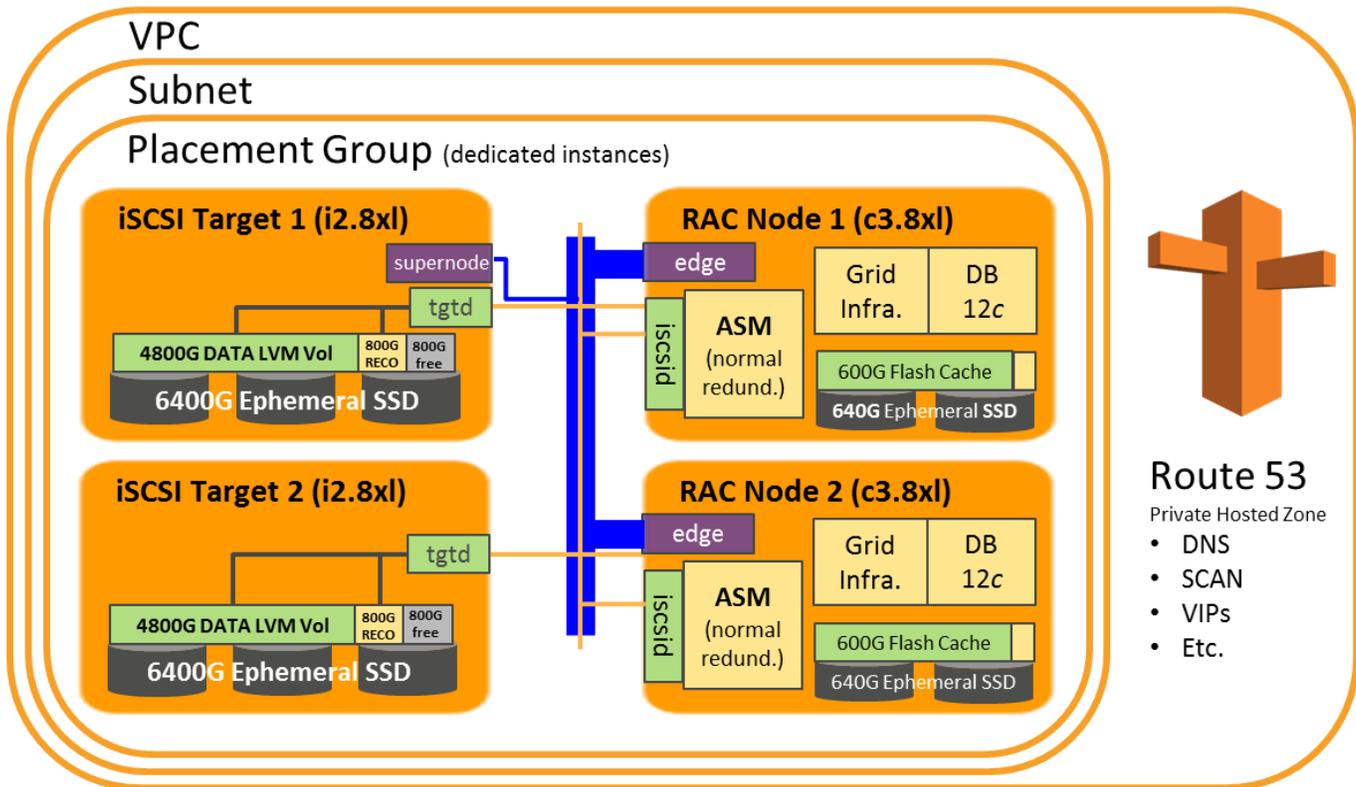


Oracle RAC Architecture

The Oracle Real Application Clusters (RAC) architecture is a shared-everything database cluster architecture. Oracle RAC allows a single database (a set of data files) to be concurrently accessed and served by one or many database server instances.

For details on how to build Oracle RAC on Amazon EC2, see the article [Deploying Scalable Oracle RAC on Amazon EC2](#). The design discussed in the article consists of two iSCSI target NAS devices serving their local SSD storage, and two RAC nodes. To minimize latency and maximize throughput, the architecture deploys all components on cluster compute EC2 instances with 10 GbE network capabilities. To further enhance performance, the architecture deploys all components in a single EC2 placement group, which minimizes network latencies among the instances. To provide DNS resolution for the RAC-related network names, such as the SCAN address and the VIPs, the architecture deploys an Amazon Route 53 private hosted zone.

The AMIs used in the article are available at no cost through AWS Marketplace. The architecture discussed in the article is just one of many methods to deploy Oracle RAC.



Further Reading

For additional information on data migration with AWS services, consult the following resources.

Oracle Database on AWS:

- 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>
- Best Practices for Running Oracle Database on AWS
<http://d0.awsstatic.com/whitepapers/best-practices-for-running-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/>
- AWS Article: Deploying Scalable Oracle RAC on Amazon EC2
<https://aws.amazon.com/articles/7455908317389540>

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 OSB and RMAN with Amazon S3:

- <http://aws.amazon.com/backup-storage/gsg-oracle-rman/>

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>

© 2016, 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.