

AWS Database Blog

Scaling Your Amazon RDS Instance Vertically and Horizontally

by Marie Yap and Neha Gupta | on 10 OCT 2016 | in [Amazon RDS](#), [RDS for MySQL](#) | [Permalink](#) | [Comments](#) | [Share](#)

This post was reviewed and updated May,2022.

As a managed service, [Amazon Relational Database Service](#) (Amazon RDS) takes care scaling your relational database so your database can keep up with the increasing demands of your applications.

In this post, we look into how we can vertically and horizontally scale your RDS instance. *Vertical scaling* refers to adding more capacity on your storage and compute of your current RDS instance. In contrast, *horizontal scaling* refers to adding additional RDS instances for reads and writes.

Vertical scaling

Vertical scaling is the most straightforward approach to adding more capacity in your database. Vertical scaling is suitable if you can't change your application and database connectivity configuration. You can vertically scale up your RDS instance with a click of a button. Several [instance sizes](#) are available, from general purpose to CPU and memory optimized, when resizing in [Amazon RDS for MySQL](#), [Amazon RDS for PostgreSQL](#), [Amazon RDS for Maria DB](#), [Amazon RDS for Oracle](#), or [Amazon RDS for SQL Server](#). Instance types have combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your database. In addition, each instance type includes several instance sizes, which allows you to scale your database to the requirements of your target workload.

The following are some things to consider when scaling up an RDS instance:

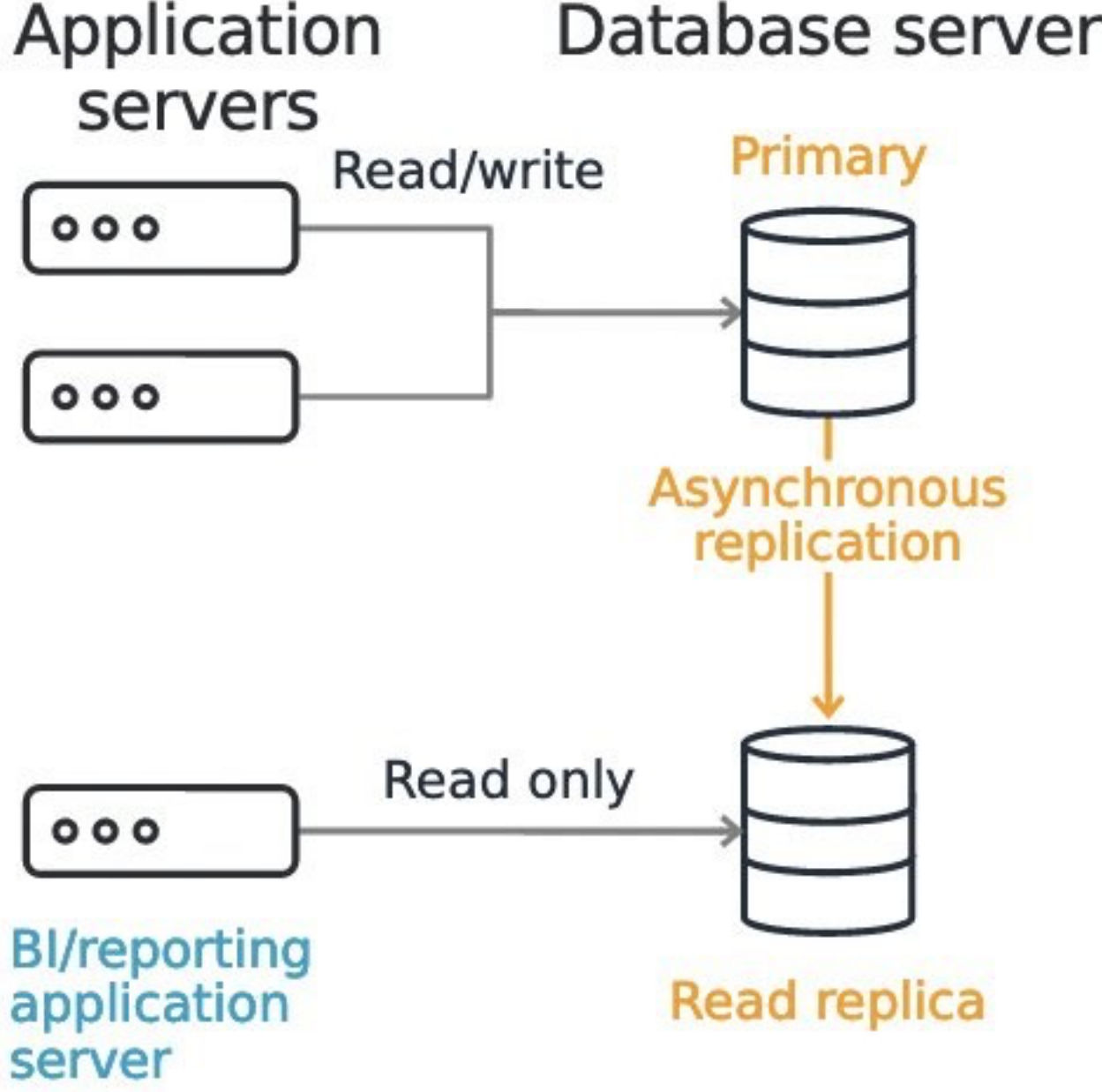
- Before you scale, make sure you have the correct licensing in place for your commercial engine such as Oracle, especially if you Bring Your Own License (BYOL). You can use License Manager to centrally track usage of your Oracle database licenses based on your license agreement terms.
- Database instance class support varies by database engine and AWS Region.
- Determine when you want to apply the change. You can apply the change immediately or during the maintenance window specified for the instance.
- Storage and instance type are decoupled. Also, modifying your storage doesn't incur any downtime. If your workload is predictable, you can separately modify your DB instance to increase the allocated storage space or improve performance by changing the storage type (such as to General Purpose SSD to Provisioned IOPS SSD).
- If you have an unpredictable workload, you can manage your storage capacity by enabling [Amazon RDS storage autoscaling](#).
- There is minimal downtime when you're scaling your RDS instance up on a Multi-AZ environment because the standby database gets upgraded first, then a failover occurs to the newly sized database. A Single-AZ RDS instance is unavailable during the scaling operation.

Horizontal scaling

Horizontal scaling increases performance by extending the database operations to additional nodes. You can choose this option if you need to scale beyond the capacity of a single DB instance. An advantage of horizontally scaling in Amazon RDS is that AWS handles the infrastructure management, provisioning, and configuration of additional nodes. You can easily create additional nodes from the Amazon RDS console or API.

To scale your read operations, you horizontally scale your database through read replicas. When you create a read replica, Amazon RDS creates read-only copies of your database and manages the asynchronous replication from the primary database. Amazon RDS DB engines such as MySQL, MariaDB, Oracle, PostgreSQL, and SQL Server all have the read replica feature.

The following figure illustrates how a read replica synchronizes from the primary database and how an application accesses it.

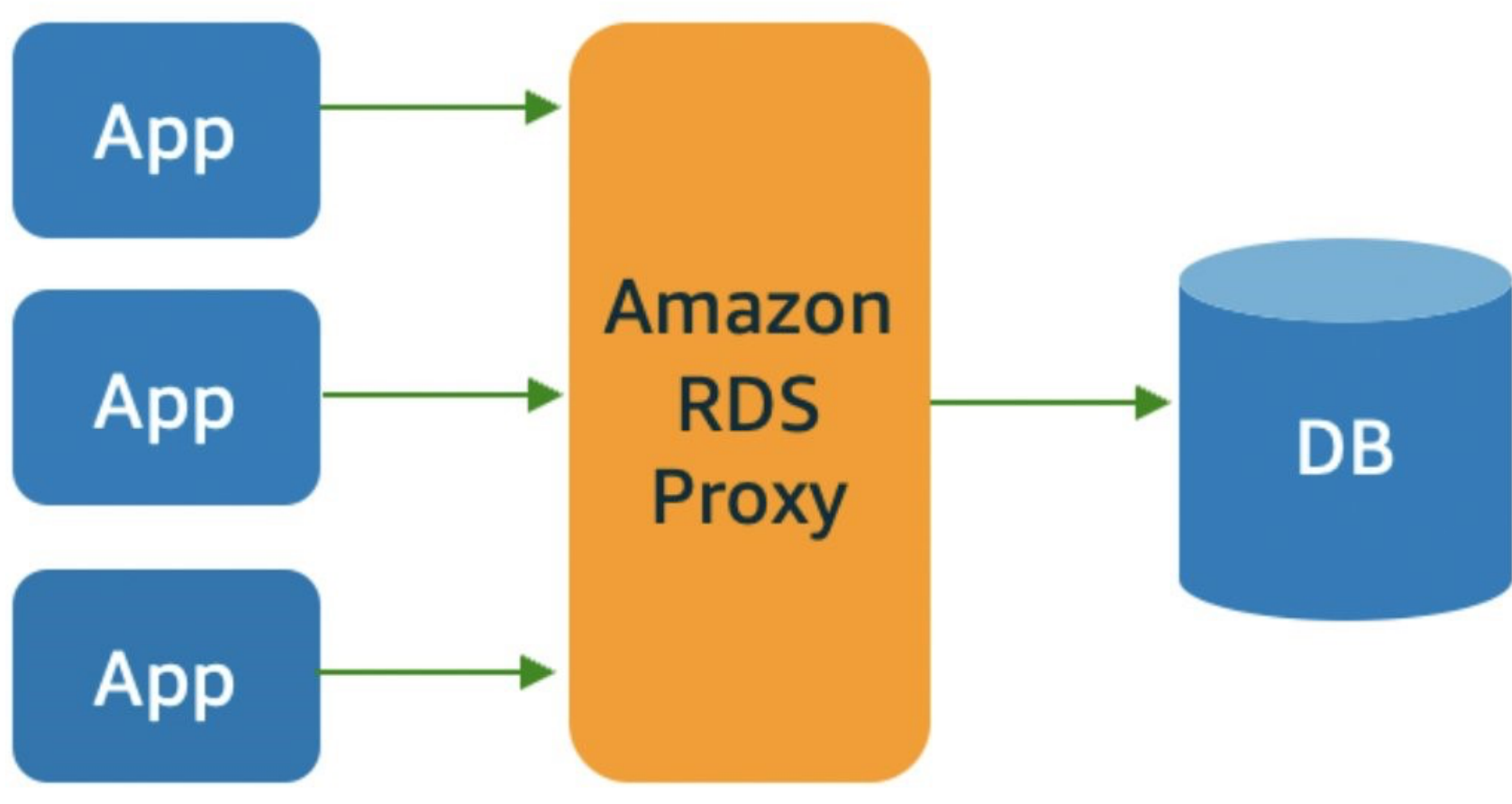


You can use read replicas to increase performance for read options, for example redirecting the read traffic for business reporting and read queries from applications. Another use case is for disaster recovery—you can promote the read replica to the primary database if it becomes unavailable. However, it's important to note that read replicas are not a replacement for the high availability and automatic failover capabilities that Multi-AZ provides.

Before you create a replica, it's important to understand several factors, such as support for multi-Region deployment and storage types, plus how each database engine performs the replication. For more information about read replicas, refer to [Overview of Amazon RDS read replicas](#).

Each read replica has a unique Domain Name Service (DNS) endpoint. To distribute the read request across multiple read replicas, you use the [Amazon Route 53](#) weighted record sets. For more information on how to set this up, refer to [How can I distribute read requests across multiple Amazon RDS read replicas](#).

Another way to scale your database is using [Amazon RDS Proxy](#) to support hundreds of thousands of connections. RDS Proxy is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. It sits between your application and Amazon RDS to manage connections to the database.



You can enable RDS Proxy for most applications with no code changes, and you don't need to provision or manage any additional infrastructure. All you have to do is point your application to the RDS Proxy endpoint instead of the database endpoint.

An RDS Proxy instance maintains a pool of established connections to your DB instances, reducing the stress on database compute and memory resources that typically occur during connections. RDS Proxy also shares infrequently used database connections so that fewer connections access the database. This connection pooling enables your database to efficiently support a large number and frequency of application connections so that your application can scale without compromising performance, which improves database efficiency and application scalability.

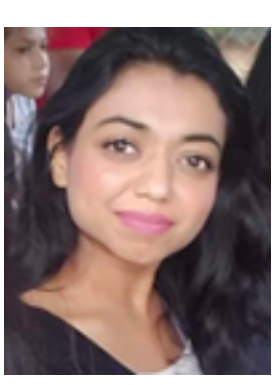
Conclusion

In summary, you can scale your RDS configuration up or out to meet the growing needs of your applications. RDS takes care of the heavy lifting in scaling your database so you can focus more on your application or applications.

About the Author



Marie Yap is a Principal Solutions Architect for Amazon Web Services based in Hawaii. In this role, she helps various organizations begin their journey to the cloud. She also specializes in analytics and modern data architectures.



Neha Gupta is a Solutions Architect at AWS and have 16 years of experience as a Database architect/DBA. Apart from work, she's outdoorsy and loves to dance.

TAGS: [Amazon RDS](#), [RDS](#), [RDS instance](#), [RDS MySQL](#), [scaling](#)

Comments

1 Comment

[Login](#)

G

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

D

f

X

G

4

Share

Best

Newest

Oldest

Matthew

a month ago

A little obsessed with scaling up? Is scaling down not possible?

0

0

Reply

Share