

# Using Amazon RDS DBaaS To Meet Business Growth

This solution brief outlines how setting up MySQL Amazon RDS instances can meet your business growth requirements. Amazon RDS is suitable for production workloads, and its ease of initial setup accommodates rapid deployment and application development.

## **This is a great option for:**

- Companies with unpredictable traffic patterns and modest operational budgets
- Enterprises with variable to high usage applications, with spiking traffic peaks
- Scaling environments to meet growing workload demands
- Databases with some level of automation

## Summary

**Amazon Relational Database Service (RDS)** removes much of the complexity traditionally associated with deploying and running a production MySQL environment. It offloads many common operational tasks such as failover and backups. The trade-off is that it also removes some low-level visibility.

This reduction in visibility can be offset when deployed in parallel with **Percona Monitoring and Management (PMM)**. This architecture is a good fit for variable, or high-traffic, cloud-based applications. The ability to dynamically add or remove replicas allows you to be more cost-effective, while still supporting high levels of traffic.

This document describes a proven, standard, RDS architecture that is easy to set up, and includes automated failover and basic business continuity components. This solution is based on MySQL Community Server 5.7 and ProxySQL, and is an ideal setup for AWS environments.

## Use case

Setting up a single MySQL Amazon RDS instance is simple. Due to the ease of initial setup, RDS is suitable for production workloads and can accommodate rapid deployment and application development.

Once the application passes a point where traffic cannot be served by a single instance, RDS allows you to add capacity on-demand through read replicas. This lets you horizontally spread read traffic across more nodes once you can no longer vertically scale your main instance.

Combining read replicas with ProxySQL adds transparent read/write splitting and load balancing across your replicas.

The ease of adding and removing replicas allows you to cost-effectively manage and add capacity during your peak traffic, and not pay for idle capacity during lower traffic times.

This environment is ideal for variable to high-traffic web applications and frameworks. From a cost perspective, capacity planning can be challenging when you have unpredictable, or even known traffic spikes, as you want to avoid paying for idle capacity. Amazon RDS also provides uptime SLAs as high as 99.99%, as this environment is meant for applications that require a guarantee that no data is lost during failover.

This solution is tailor-made for companies with variable or unknown traffic patterns, modest operational budgets, and ideally some level of existing automation. Although it can be done manually from the AWS console, you would gain the most benefit by managing your infrastructure via the API.



### Cloud Architecture Solution



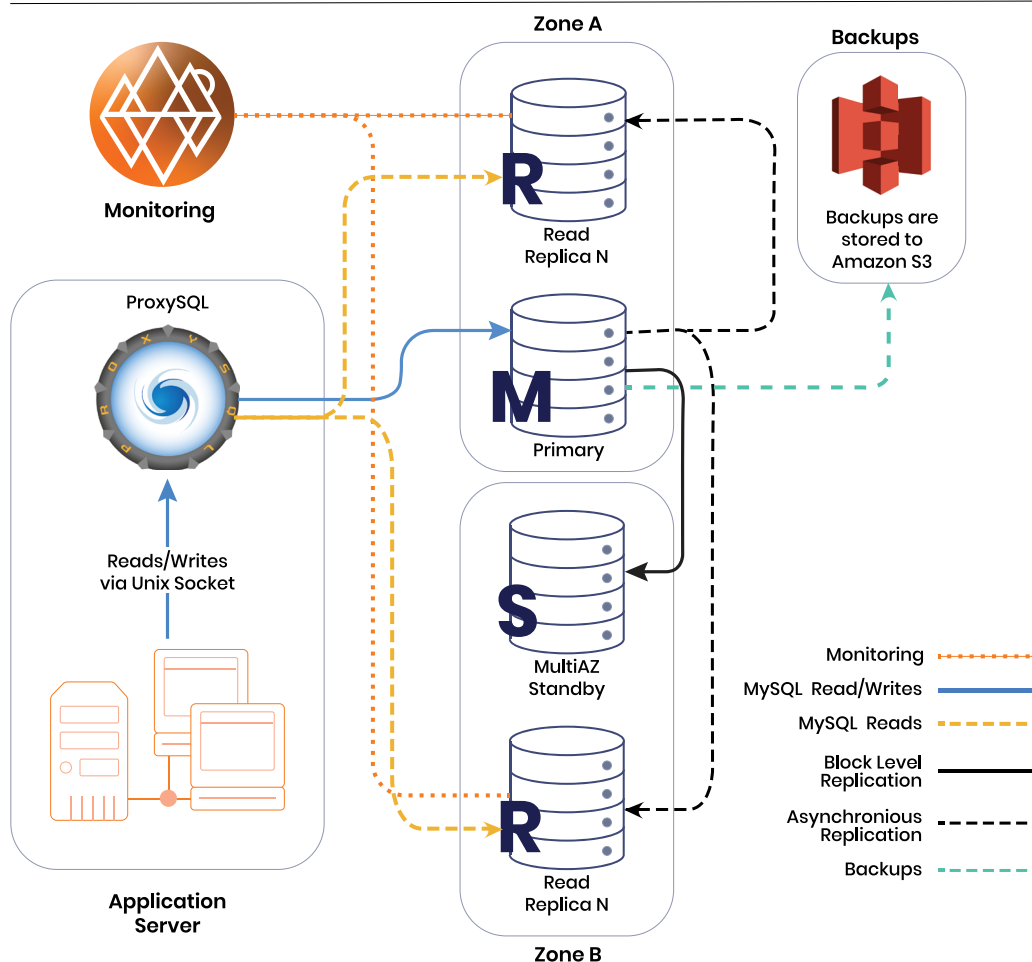
## Benefits of this approach

- The environment is easy to set up.
- This solution incorporates tightly integrated and automated backups.
- It includes automated maintenance and patching.
- Failover is transparent to applications and doesn't affect application performance.
- You can scale reads on demand.
- You pay only for the infrastructure you need.
- Allows you to scale read and writes as needed.
- Allows you to distribute application workload across your entire system, minimizing updates and work on applications while maximizing flexibility.

## Downsides of this approach

- Auto-scaling is not an option, and you have to manually add or remove replicas.
- The solution relies on the Amazon process for maintenance and updates.
- Amazon provides only limited automatic failover; you must manually failover down primaries to a replica. The whole datacenter must go down for failover to occur (zone outage).
- You must use software versions supported by Amazon RDS.
- Failover happens within minutes rather than seconds (due to manual process).
- RDS provides limited access and control of your operating system, infrastructure variables, and configuration.

## ARCHITECTURE DIAGRAM



## Architecture

This architecture is relatively simple. Rather than pointing the application to a single RDS endpoint, database traffic is sent to a local instance of ProxySQL on each application server, which routes to the appropriate primary or read replica. By leveraging read/write splitting in ProxySQL, queries are routed based on the defined rules and query text.

The primary RDS instance should be launched in Multi A/Z mode to maintain the fault tolerance of the primary. Read replicas can be added or removed on demand, and added to the pool of servers available to ProxySQL. We recommended you launch an additional EC2 instance running Percona Monitoring and Management (PMM) via the Amazon Marketplace, to allow for better workload analysis.

## Components

- RDS instance configured with Multi A/Z for fault tolerance\*
- Read replicas (in multiple A/Zs) added/removed on demand as needed\*
- ProxySQL running on each application node
- EC2 instance running PMM launched from Amazon Marketplace for monitoring/analytics\*
- S3 bucket for backups automatically created

\* Enabling encryption on the storage layer is recommended when creating the instance in AWS.

## ProxySQL

ProxySQL is a high-performance MySQL protocol-aware proxy that can transparently provide the application with read/write splitting and load balancing. This allows read scaling by adding a proxy level in front of the database. The recommended implementation is co-locating of ProxySQL on the application servers; however, it can be implemented as its own independent layer. For environments with many application servers, ProxySQL's management overhead should be taken into consideration.

## Failover

Failover is handled automatically by RDS by moving the endpoint from the active instance to the standby instance. This requires no changes to the application. As replication is handled at the physical level, no data loss occurs when failing over to the standby instance. Failovers can also be manually triggered as required.

## Read replicas

To scale reads, RDS offers the provisioning of read replicas. The read replicas are used to serve read traffic that is tolerant of stale reads, where writes may take some time to become available on the replica. No additional configuration is necessary for read replicas in RDS: All configuration is copied from the primary instance. ProxySQL can account for unavailable read replicas and redirect traffic accordingly.

## Backups

Backups are handled automatically by RDS on a user-defined schedule. Snapshots are taken from the standby instance to minimize the impact on the running application. Additional manual snapshots can be triggered on a scheduled or ad-hoc basis. Note: Automated snapshots can be restored only through RDS and are not portable.

Backups can be restored to any point-in-time during the specified retention period. This is generally to within 5 minutes of the current time.

## Monitoring

By default, you can enable basic or enhanced monitoring with CloudWatch. This covers only limited OS and MySQL metrics. Percona Monitoring and Management provides a comprehensive view of your RDS instance by integrating with CloudWatch, while also providing in-depth MySQL metrics with Query Analytics. The health of read replicas is monitored and handled automatically by ProxySQL, while the same metrics are available within PMM. Note: The resources available to the monitoring server are dependent on the number of instances and replicas being monitored.

## Percona can help

Managing your organization's database operations, on-premises or in the cloud, requires in-depth knowledge of potential issues, and diligent, dedicated, database software experts.

When setting up a high availability database environment, being aware of the issues we have highlighted will help protect your organization's data-based applications. It will also significantly enhance performance and scalability, enabling you to deliver a better end-user experience.

**Percona Support** services are accessible 24x7x365 online or by phone to provide assistance with tricky issues and help you keep your database running optimally.

We can provide onsite or remote **Percona Consulting** for current or planned projects, migrations, or emergency situations. Every engagement is unique, and we work alongside you to plan and create the most effective solutions for your business.

Finally, **Percona Managed Services** can support and help you manage your existing database infrastructure; whether hosted on-premises or at a co-location facility, or if you purchase services from a cloud provider or database-as-a-service provider.

To learn more about how Percona can help, and for pricing information, please contact us at **+1-888-316-9775 (USA), +44 203 608 6727 (Europe)**, or email us at [sales@percona.com](mailto:sales@percona.com).