

Using Relational Databases with AWS Lambda

Easy **connection pooling** with RDS Proxy

George Mao

Principal Serverless Specialist

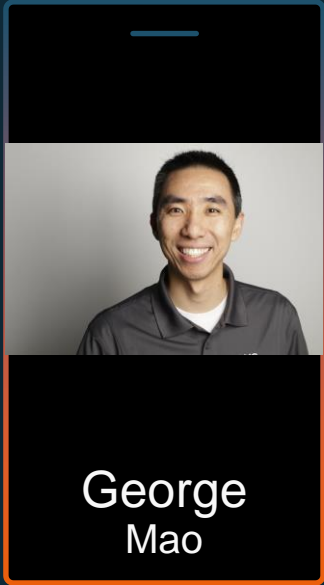
December 20, 2019

Amazon RDS Proxy

Make applications more scalable
with a fully managed database proxy



About me



George Mao

Serverless Specialist

georgmao@amazon.com

@georgemao (Twitter)

@georgemao (Slack #awsdevelopers)

Principal
Serverless
Specialist

Why are we here?

Amazon Relational Database Service (Amazon RDS)

Managed relational database service with a choice of popular database engines



Microsoft SQL Server

ORACLE



Easy to administer

Easily deploy and maintain hardware, OS, and DB software; built-in monitoring



Performant & scalable

Scale compute and storage with a few clicks; minimal downtime for your application



Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover



Secure & compliant

Data encryption at rest and in transit; industry compliance and assurance programs

Modern apps need a large number of open DB connections and high connection open/close rate



Modern apps can have 1000s of DB connections, exhausting DB resources

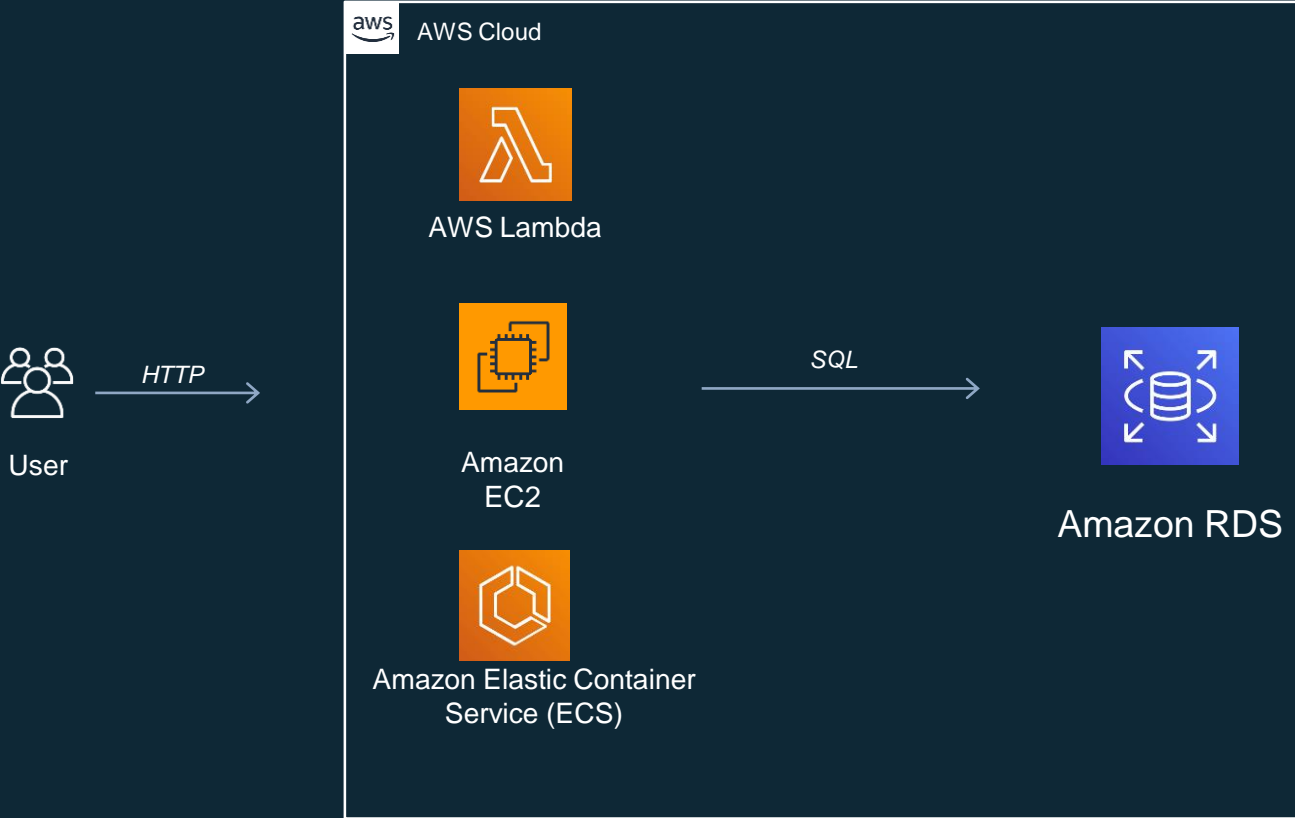


Custom failure handling: Code can contain security risks like DB credentials

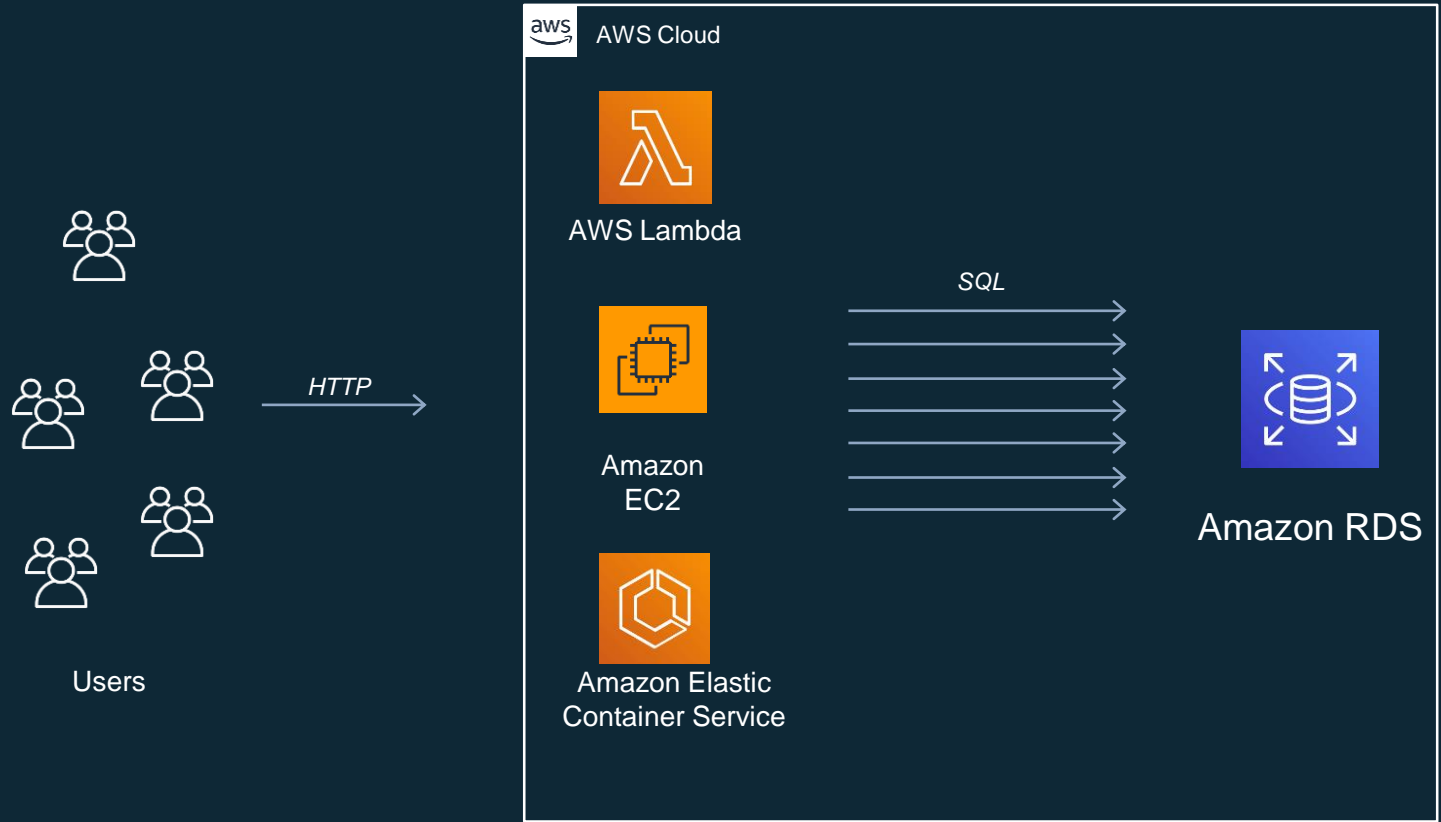


Self-managed proxy servers help manage DB load, but are difficult to deploy

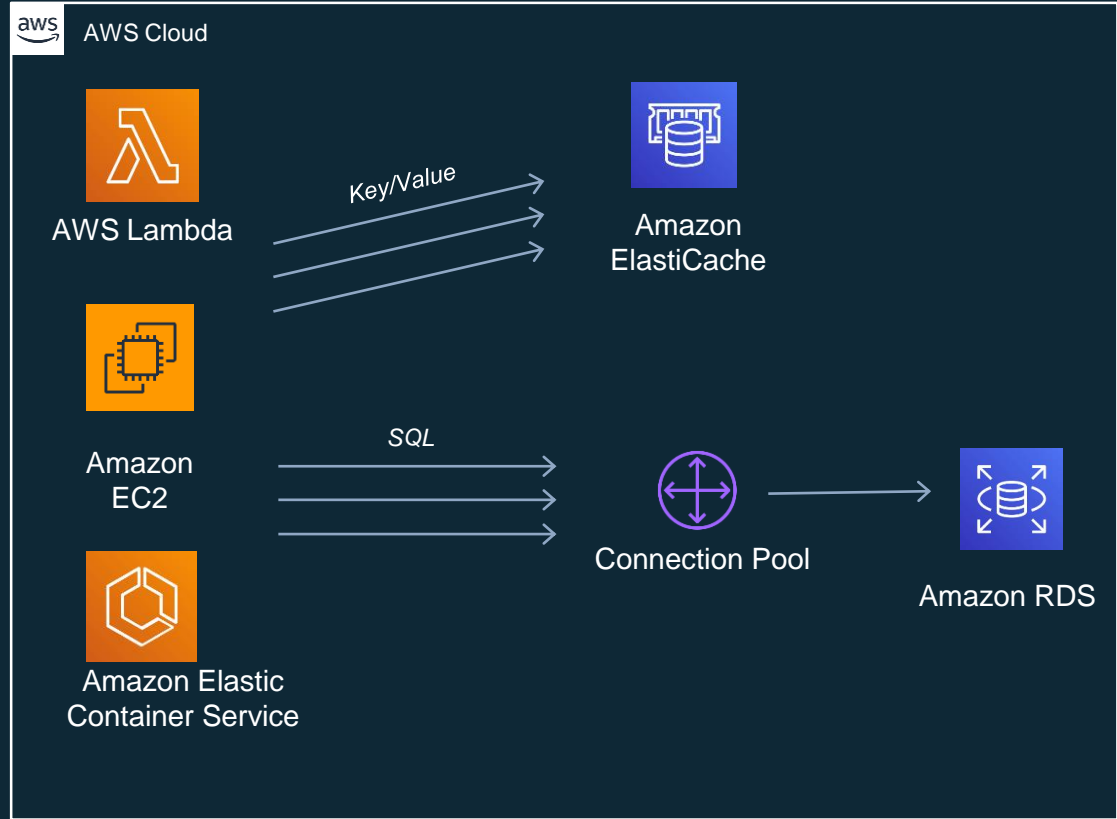
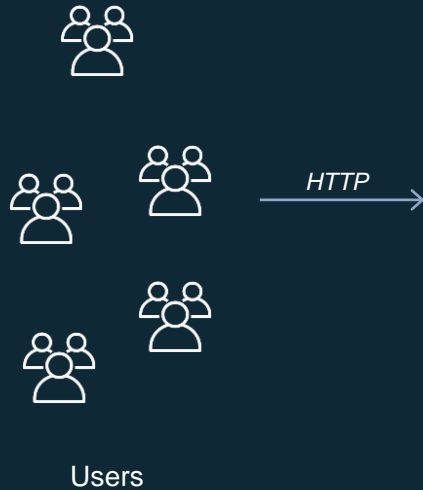
Database connections!



More database connections!



A common solution



Introducing: RDS Proxy (Preview)

Fully managed, highly available database proxy feature for Amazon RDS. Pools and shares DB connections to make applications more scalable, resilient to database failures, and secure.



Pool and share DB connections for improved app scaling



Increase app availability and reduce DB failover times



Manage app data security with DB access controls

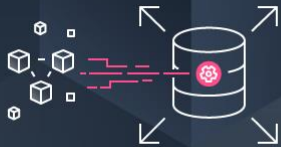


Fully managed DB proxy, compatible with your database

RDS Proxy (preview) regions and engines

Amazon RDS Proxy

Make applications more scalable
with a fully managed database proxy



RDS Proxy is available in:

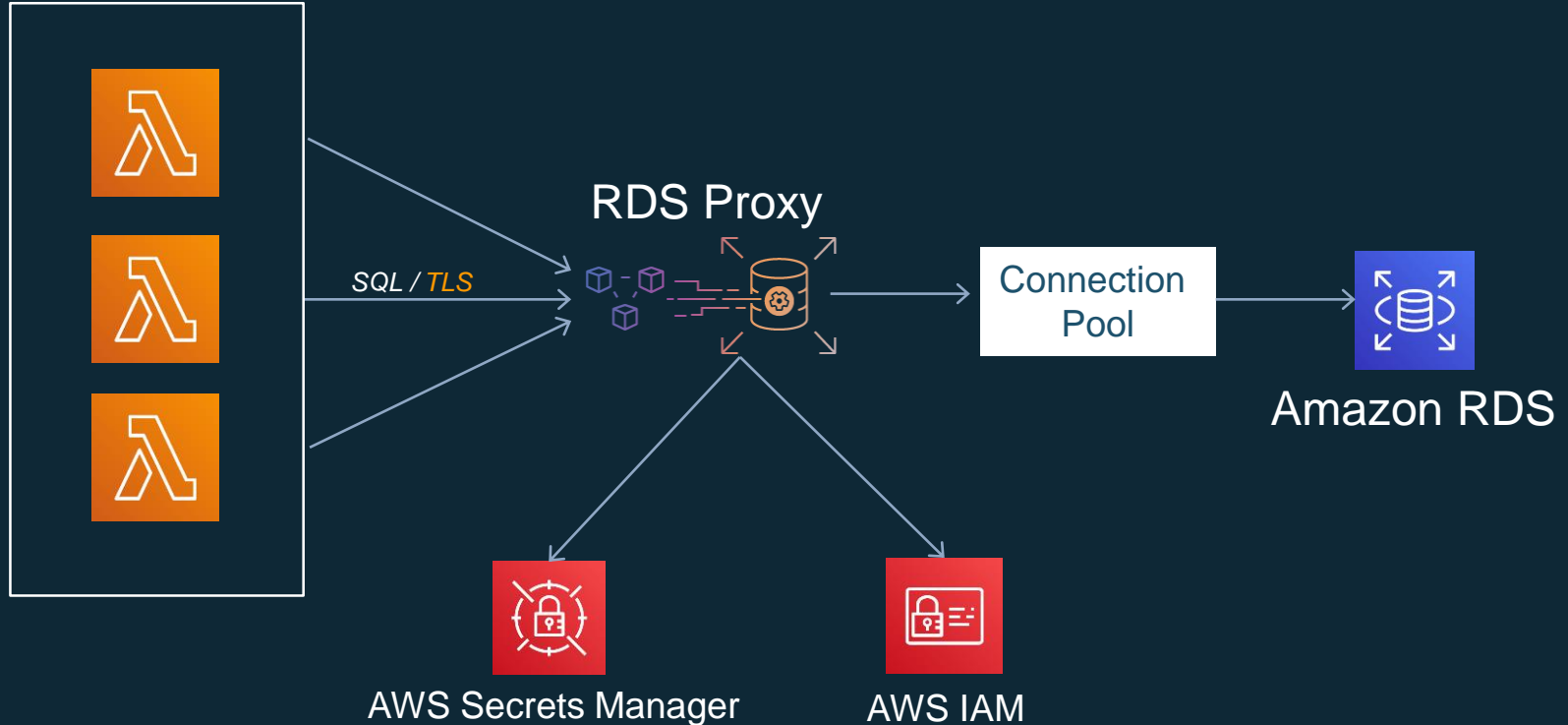
- Asia Pacific (Tokyo)
- EU West (Ireland)
- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)

RDS Proxy supports the following engines:

- RDS MySQL 5.6 and 5.7
- Aurora MySQL 5.6 and 5.7
- RDS and Aurora PostgreSQL **coming soon!**

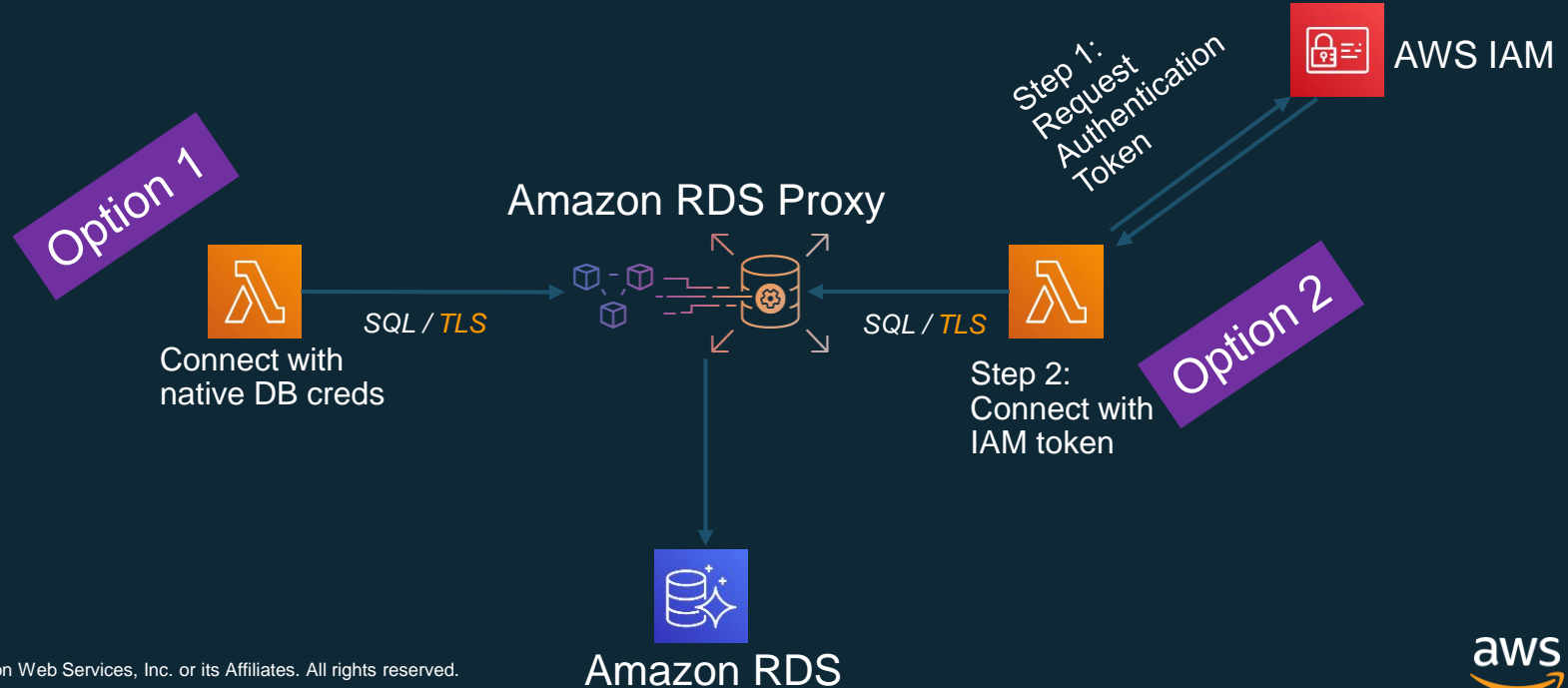
Scaling Serverless apps with Amazon RDS Proxy

Lambda functions



Two options for security: Control is in your hands

Developers can use either IAM or native DB credentials to connect





Demo

Monitoring a Proxy



CloudWatch Metrics & Logs

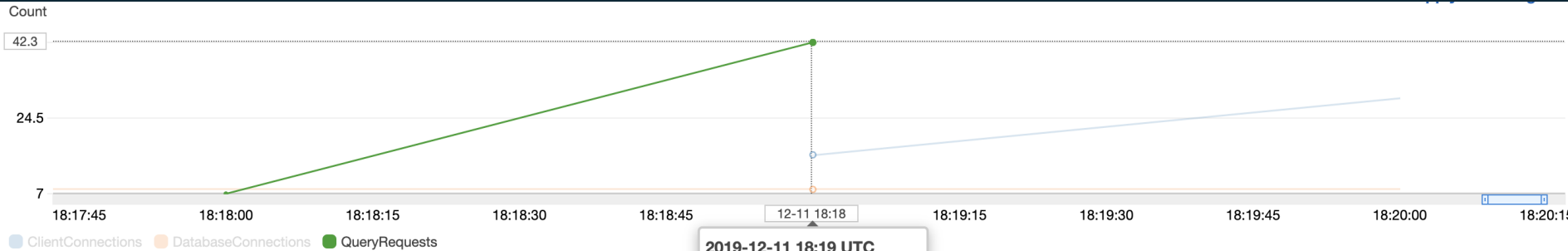
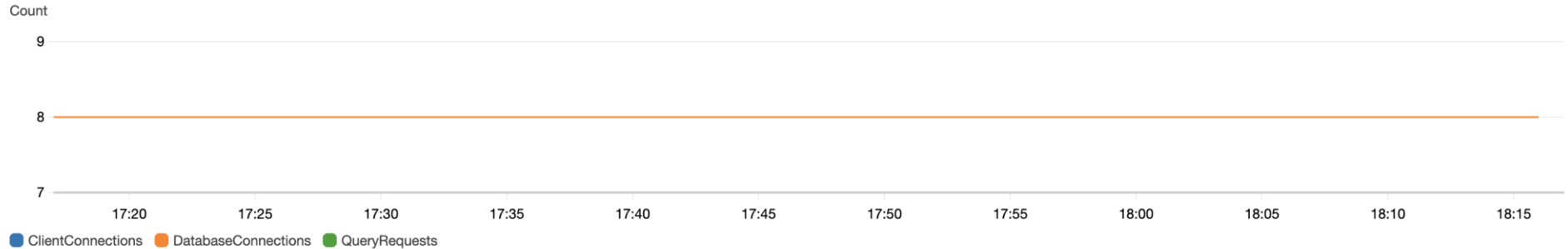
Useful metrics:

- ClientConnections
- QueryRequests
- DatabaseConnections

Connection logging in Log Group:

- /aws/rds/proxy/[proxy-name]

Sample CloudWatch Metrics



2019-12-11 18:19 UTC

1. QueryRequests	42
2. ClientConnections	16
3. DatabaseConnections	8

All metrics | Graphed metrics (3) | Graph options | Source

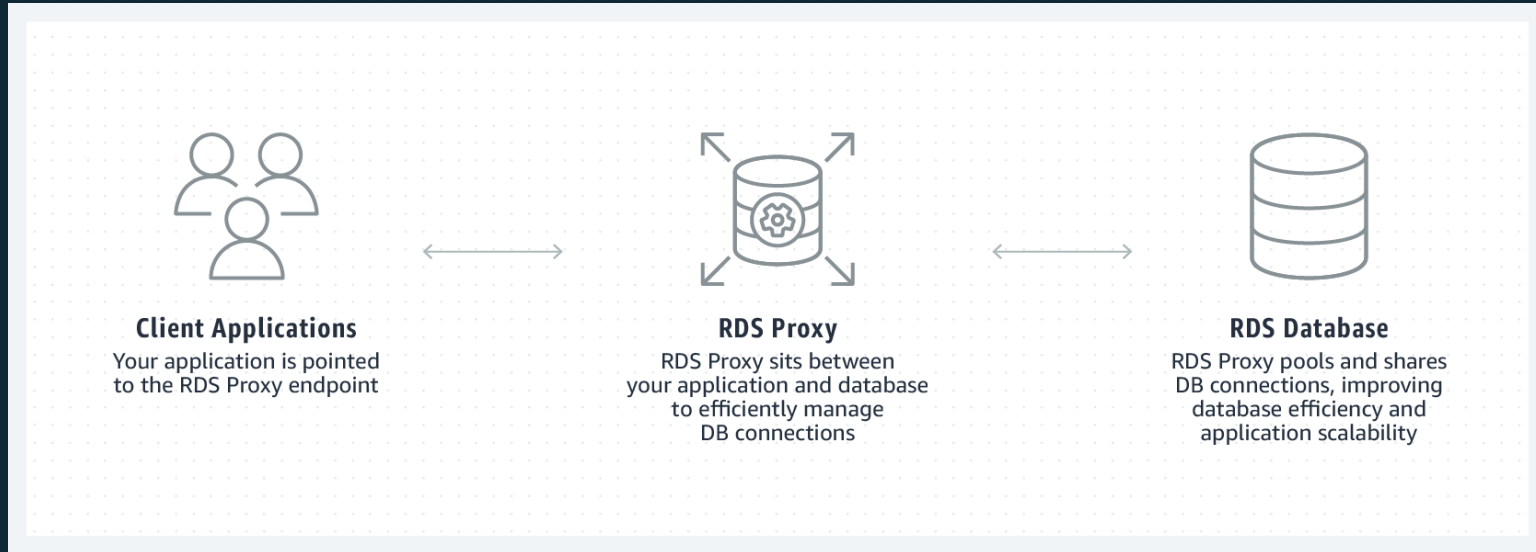
Math expression ? | Dynamic labels

Statistic: Sum | Period: 1 Minute | Remove



Getting Started

<https://aws.amazon.com/blogs/compute/using-amazon-rds-proxy-with-aws-lambda/>

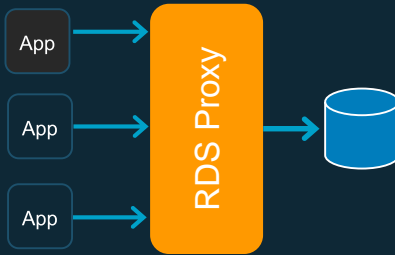


<https://aws.amazon.com/rds/proxy>

Thank you!

georgmao@amazon.com

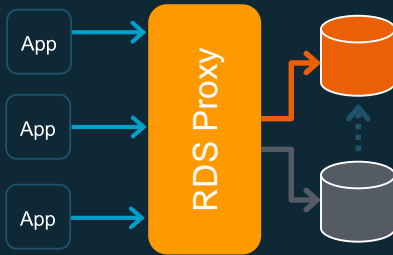
Connection pooling



Connection pooling

- Share database connections between transactions
- Detection of session state altering operations causes pinning
 - Sets of system variables, sets of user-defined variables, calls of locking functions, tables locks, creates of temporary tables, prepares statement prepare call
- Exclude specific variable sets from pinning

Seamless and faster failovers



Fast, seamless failover

Application connections preserved during failovers

Detects failovers and connects to standby quicker, bypassing DNS caches

Up to 66% faster failover times