Which technology to choose in AWS ?

RDS / Aurora / Roll-your-own

April 17, 2018

Daniel Kowalewski

Senior Technical Operations Engineer
Percona

AWS MySQL options

RDS for MySQL

MySQL M

Aurora MySQL

Amazon Aurora

EC2
(Roll-your-own)

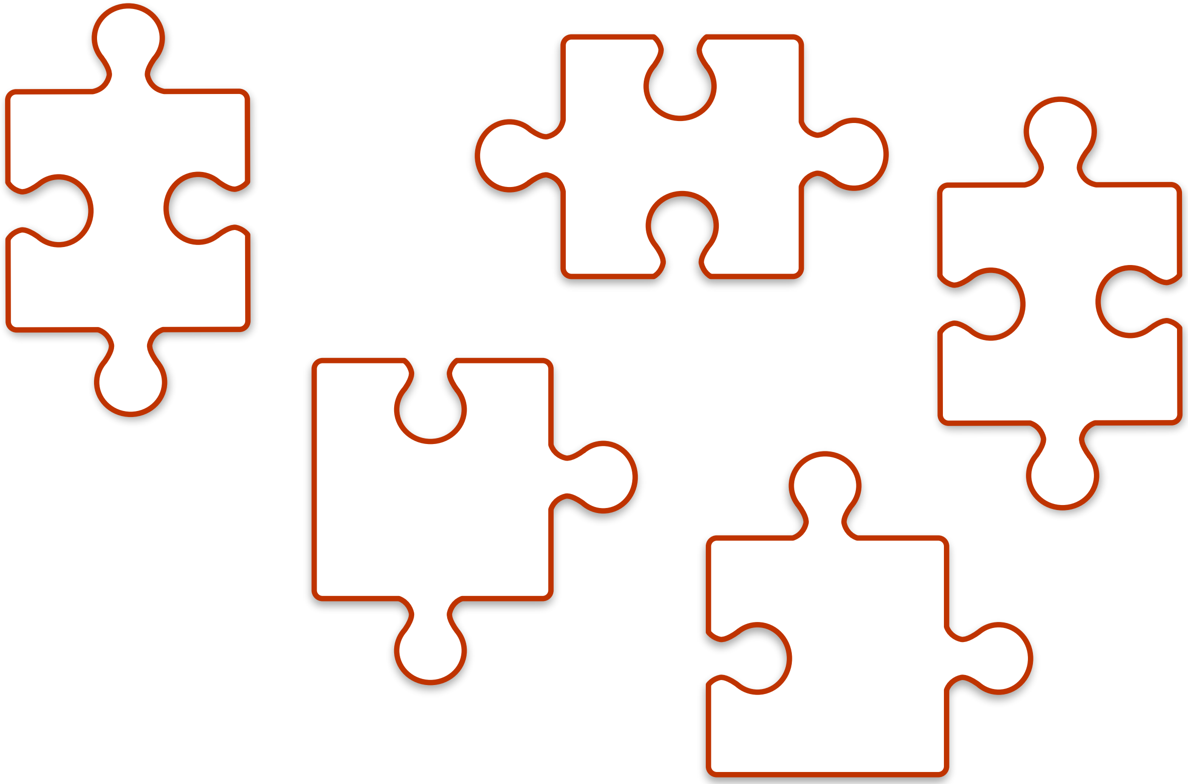
MySQL

PERCONA Server for MySQL

MariaDB

EC2 Roll-your-own

- EC2 Instances
- MySQL
- Galera Cluster (PXC)
- ProxySQL
- Async/semisync replication
- Orchestrator
- MHA
- Xtrabackup
- Percona Monitoring and Management



Agenda

- Overview of three options
  - RDS for MySQL
  - Aurora MySQL
  - EC2 Roll-your-own
- Scaling
- High Availability
- Backups
- Performance
- Pricing

RDS for MySQL

Amazon RDS for MySQL

- MySQL DBaaS
- Conceptually familiar
- Offload some management overhead

RDS Benefits

- Easy to set up
  - No OS setup or configuration
  - No database/package installation
  - Few clicks to running instance
  - Easy to create read replicas

RDS Benefits

- Less management overhead
  - Automating time-consuming administration
  - Hardware provisioning
  - Patching
  - Backups
  - High availability (Multi-AZ)
  - Security (not related to mysql user privileges)
  - Built-in encryption

RDS Benefits

- Scale
  - Vertical: Resizable computing capacity
  - Horizontal: Read replicas

RDS Limitations

- No Super privilege
- No OS access
- No direct access to logs
- 70 parameters you can’t change
- Very limited plugin support
- Storage limit: 16TB
- No access to backups
  - Vendor lock-in

Aurora MySQL

Amazon Aurora MySQL

- NOT MySQL
  - But "MySQL Compatible"
- Solution for high availability
- Sometimes a solution for scaling

Aurora MySQL Benefits

- Everything RDS for MySQL provides, plus...
- Very low latency replication
- Data size up to 64 TB
- Performance gains for some workloads
  - Especially high concurrency, large instances
- Minimal downtime for patches

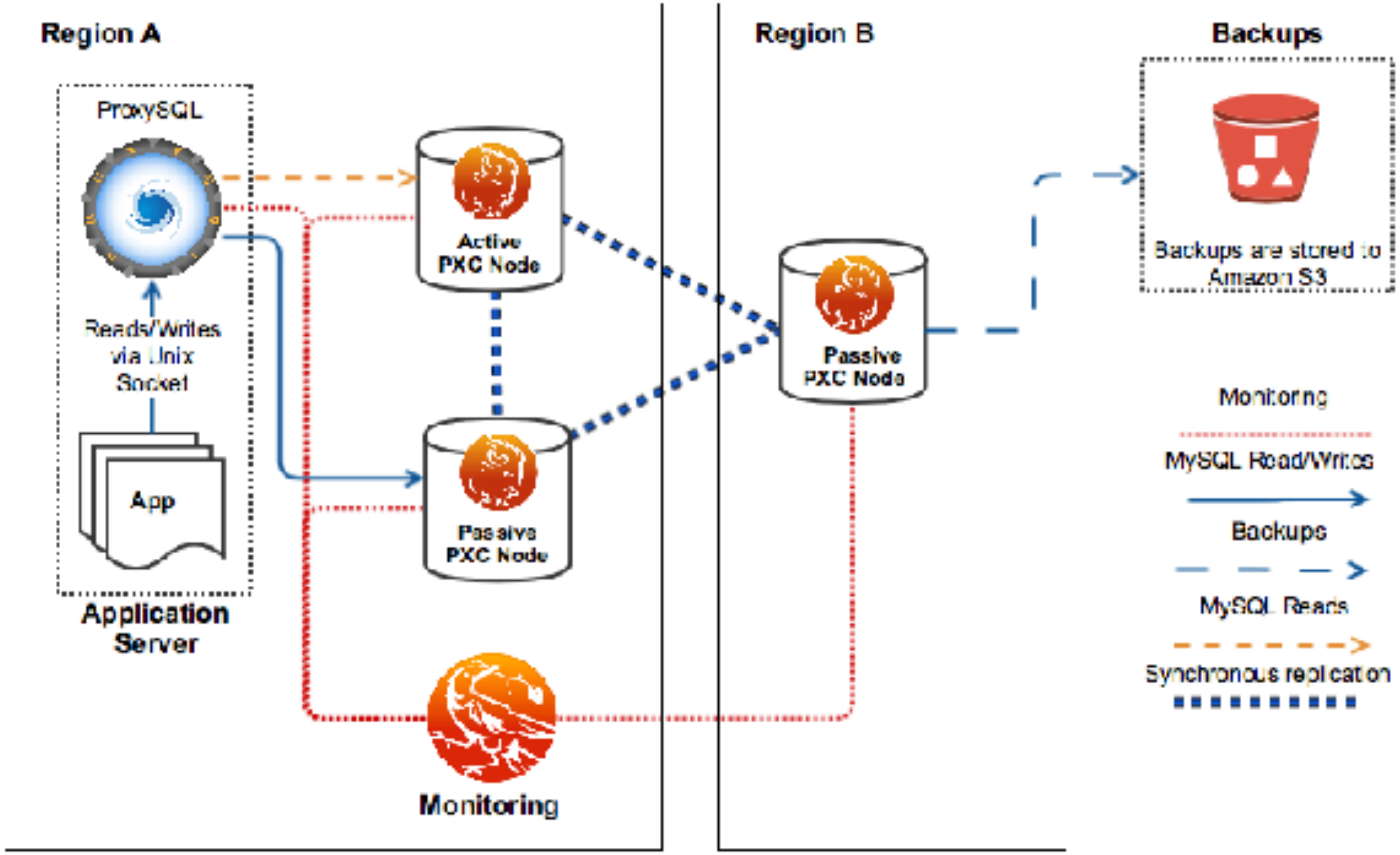
Aurora Limitations

- Mostly the same limitations as RDS for MySQL
- Storage limit: 64TB
- Slower (write latency) for small workloads
- Only InnoDB
- Only one logical copy of your data
    - Replicated 6 times
- No performance_schema (for 5.7, at the moment)
- It's not MySQL, so don't assume it acts like MySQL

One approach to roll-your-own

PXC + ProxySQL on EC2

PXC + ProxySQL on EC2



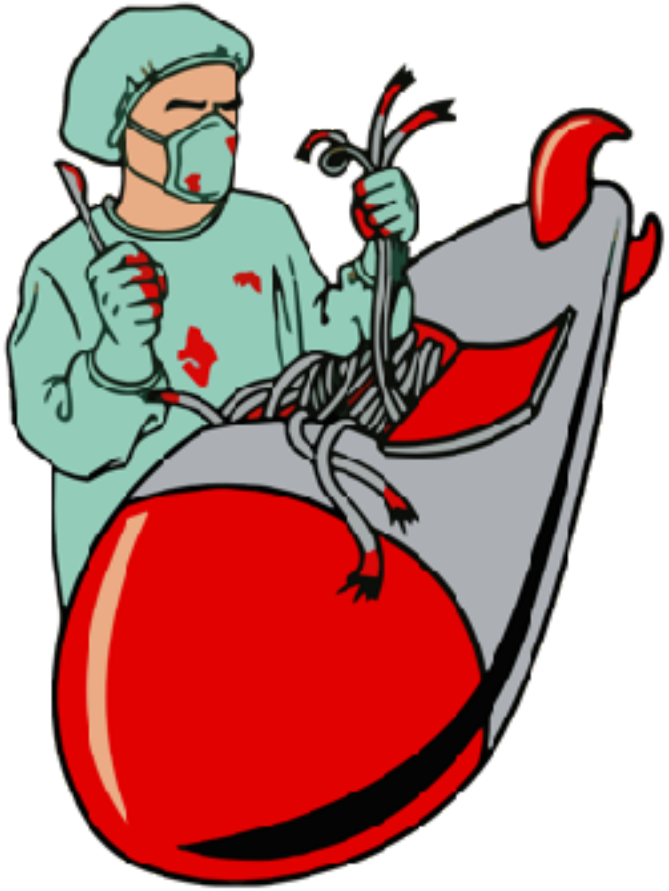
PXC on EC2

- Auto-provisioning with node AMI (SST)
- ProxySQL on app nodes
  - automatic failover
  - read/write splitting
  - auto scaling
- PMM
- It's not hard to automate backups

Benefits of roll-your-own

- More control over environment
    - Patching
    - MySQL parameters
    - Fine-tune OS
    - Which software and which versions
- More visibility
    - OS metrics

Limitations of roll-your-own

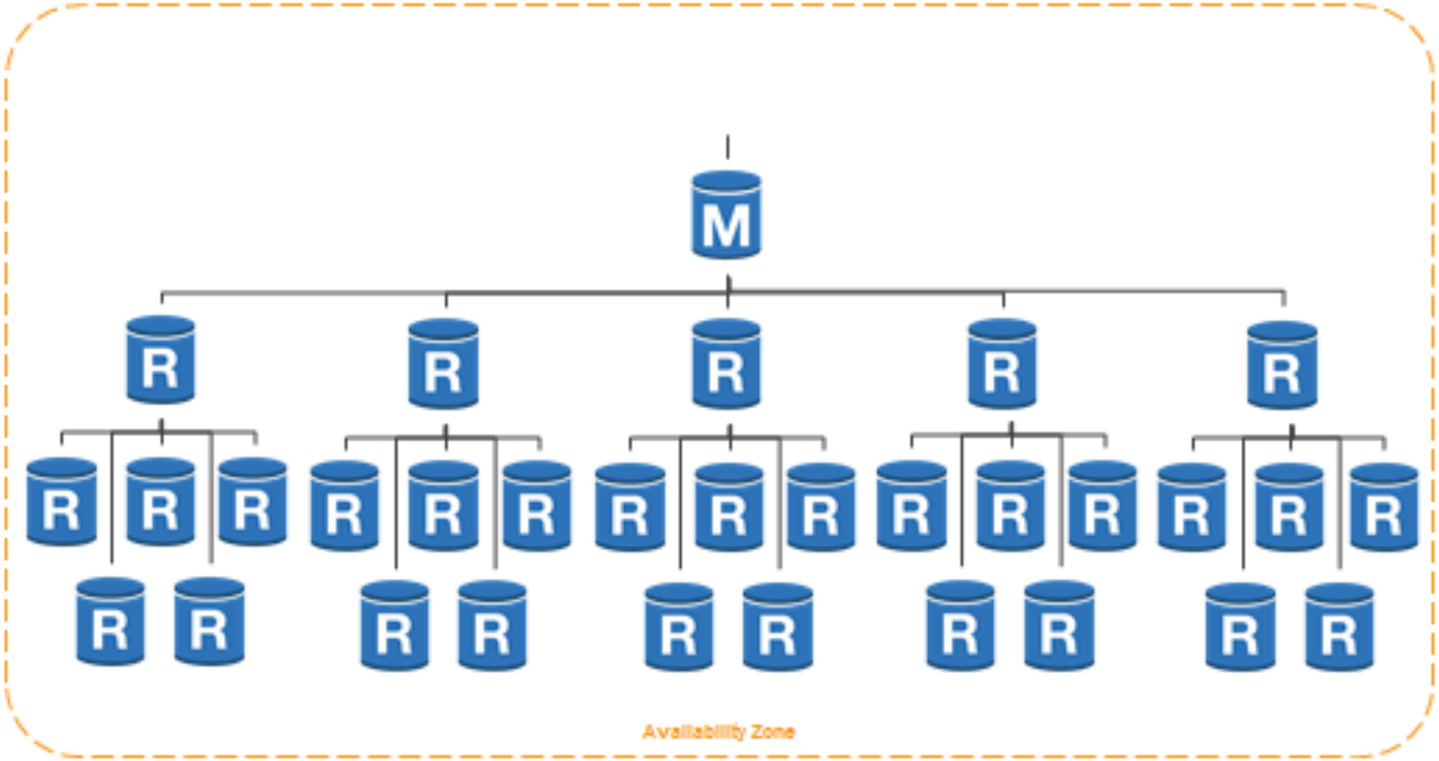


- Harder than RDS or Aurora to set up and maintain
- Depends on your chosen architecture

Scaling Reads

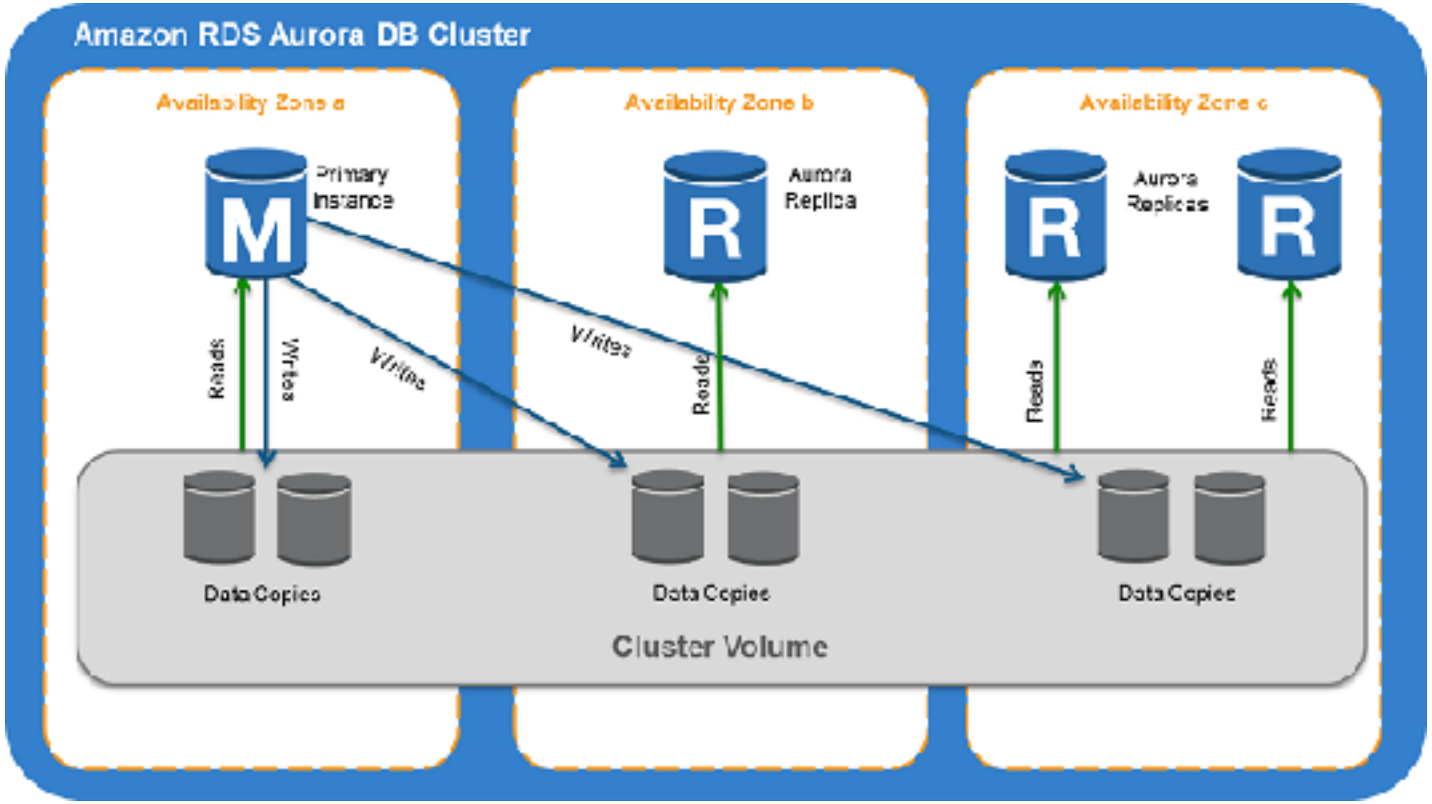
Scale – RDS for MySQL

- Up to 5 Read Replicas
- 2 tiers (30 RR)
- Normal Async Replication
- No r/w splitting
- No load balancing



Scale – Amazon Aurora MySQL

- Up to 15 instances
- Load balancing
  - Cluster / reader endpoints
- Shared storage
- Reduced lag
- Auto scaling



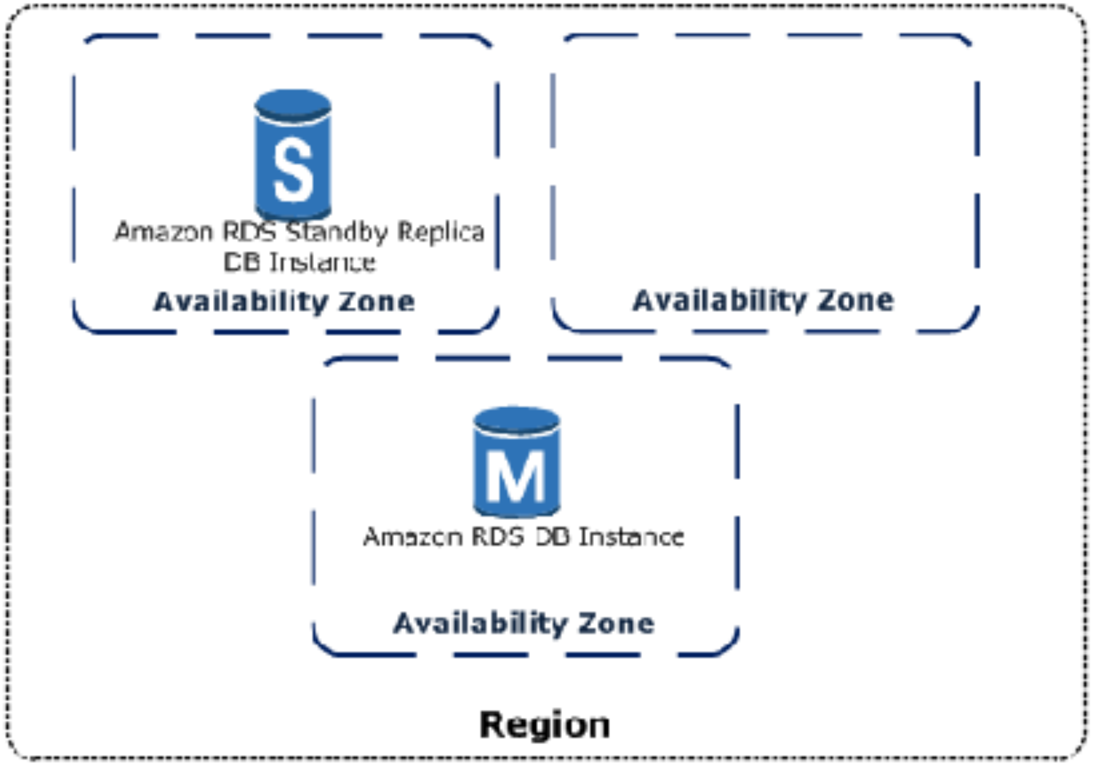
Scale - EC2 Roll-your-own

- Scaling depends on your implementation
- As many replicas and levels as you want
- Relatively simple to add Galera nodes
  - But that's not really scaling
- ProxySQL for R/W splitting, load balancing

High Availability

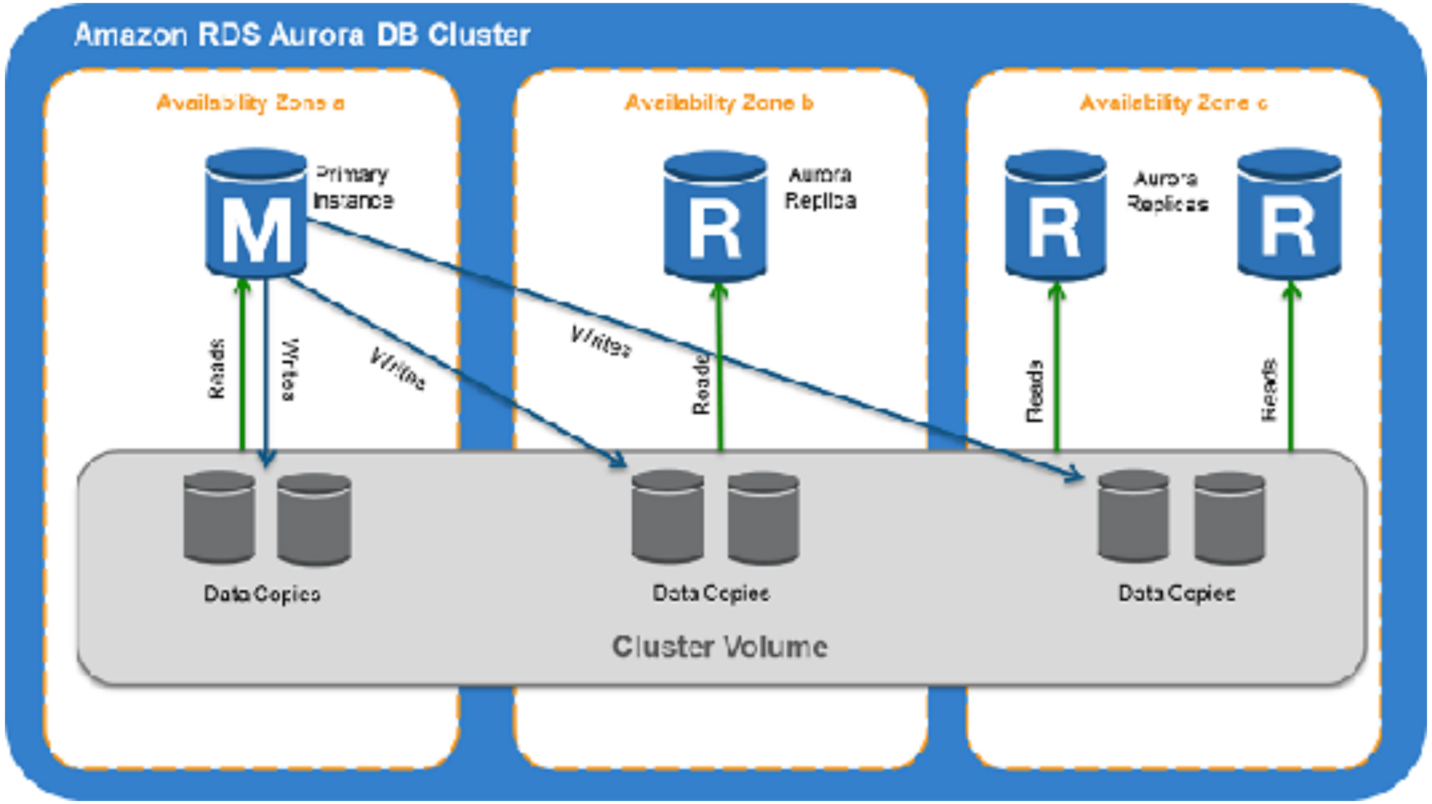
High Availability – RDS for MySQL

- Multi AZ (Standby DB Instance)
  - Synchronous replication at storage level
- Read Replica Promotion
  - Data integrity not ensured
- Failover can take 1 minute or more



High Availability – Aurora MySQL

- Connection Endpoints
- Storage: 6 copies, 3 AZs
- Automatic failure detection
- Replica promoted
- Failover takes a few seconds



High Availability - Roll-your-own

- With PXC (Galera cluster)
  - Synchronous replication to other nodes
  - No data loss for node failure
  - Automatic failure detection with ProxySQL
  - Possible to combine with semi-sync replication
- Other possibilities
  - All async/semisync
    - Orchestrator or MHA
    - ProxySQL

Backups

Backups

	EC2	RDS for MySQL	Aurora MySQL
Type/s	Physical / Logical / Binlog	Automated Snapshot + binlog archiving	continuous and incremental
Latest Recoverable Time	~1 sec	up to 5 mins	up to 5 mins
PITR resolution	Transaction	Second	Second
Storage	EBS	S3	S3
Retention	No limits	1 to 35 days	1 to 35 days
Automated restore	Your own	Yes	Yes

Performance

RDS MySQL Performance

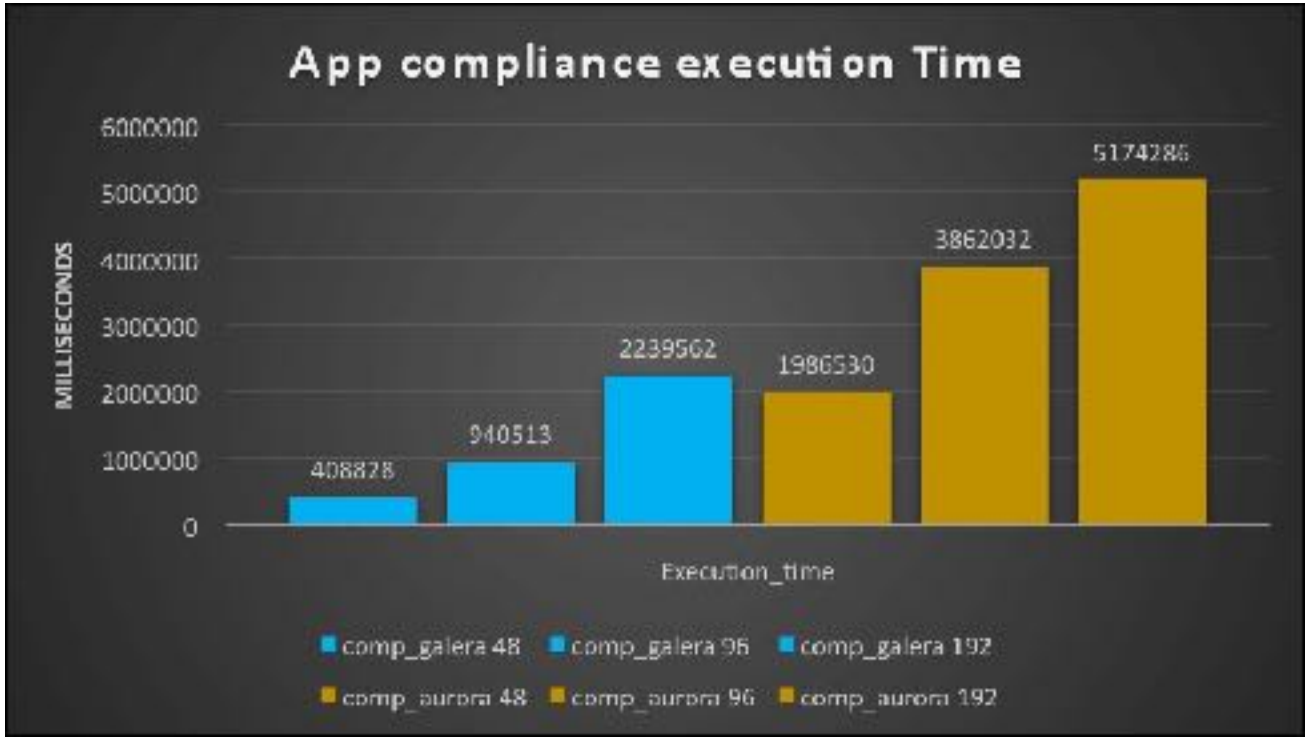
- Similar to standalone MySQL
- But when you hit a limitation, you're stuck

Aurora Performance Claims

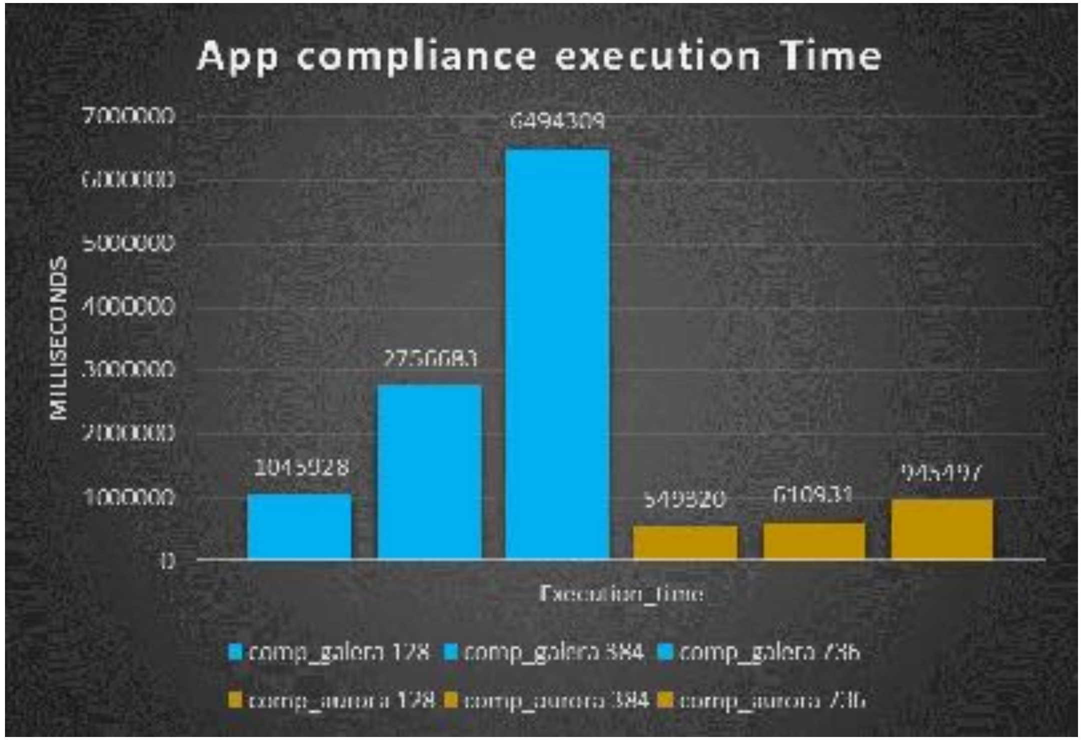
- "500,000 SELECTs/sec and 100,000 UPDATEs/sec, five times higher than MySQL running the same benchmark on the same hardware."
- Differences are more pronounced on
  - Larger instances
  - Workloads with more concurrency
- Hot spots decrease performance

Performance – Aurora vs. PXC

db.r3.xlarge



db.r3.8xlarge



Source: https://www.percona.com/blog/2016/05/26/aws-aurora-benchmarking-part-2/ By Marco Tusa

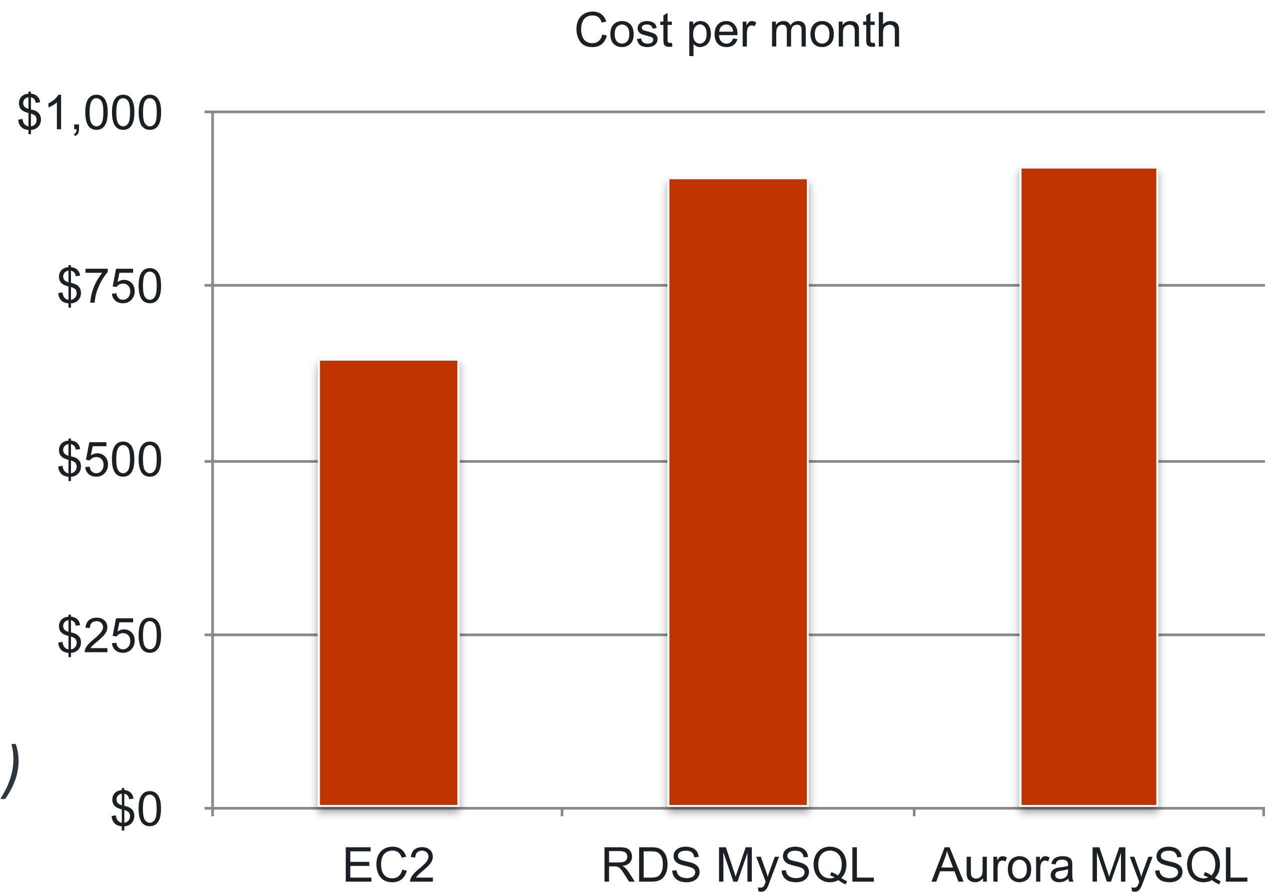
Pricing

Disclaimer

- The following chart is completely theoretical, doesn't apply to your use case, and can change at any moment.

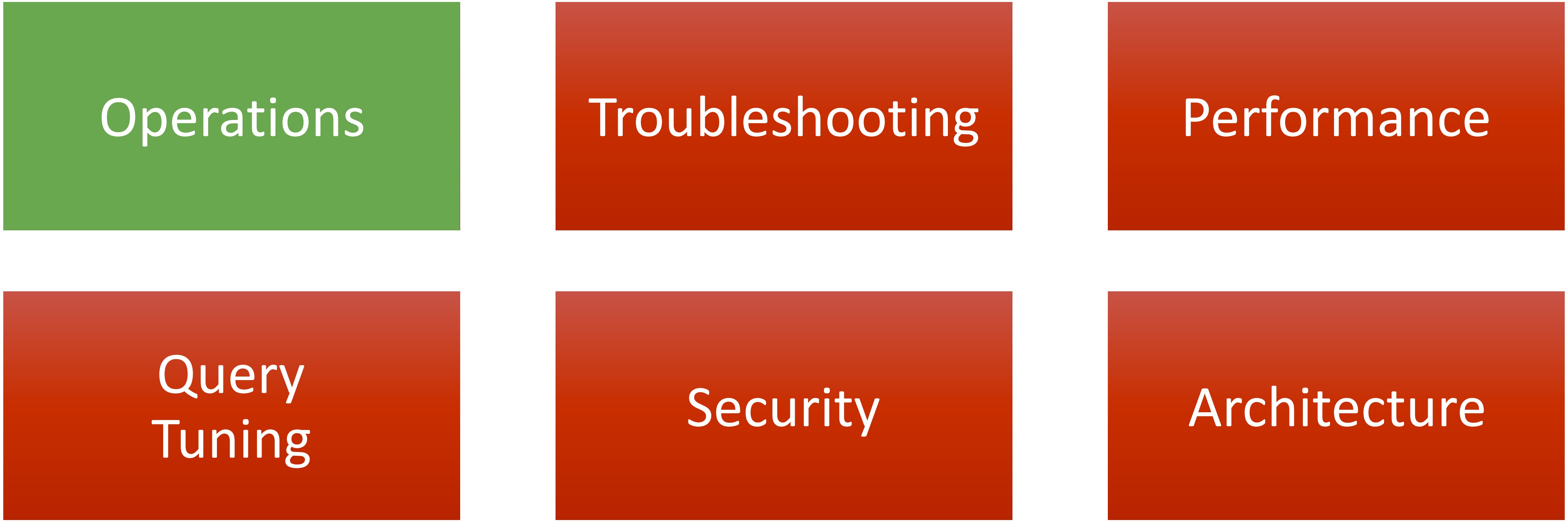
AWS Pricing

- Instance type: r3.2xlarge
  - *Memory optimized*
  - *8 vCPU*
  - *61G RAM*
  - *Enhanced Networking*
- Storage
  - *500G*
  - *1500 Provisioned IOPS*
  - *100 million IO requests (Aurora)*



Conclusion

RDS Only covers



Operations

Troubleshooting

Performance

Query Tuning

Security

Architecture

You might choose RDS when

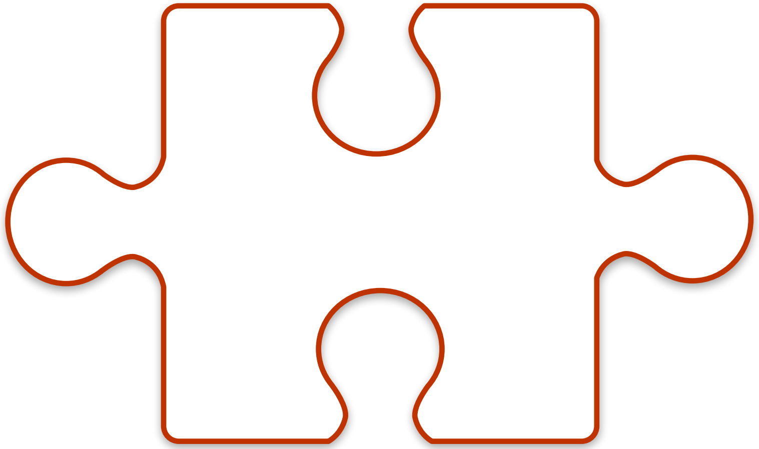
- You've double-checked all RDS limitations.
- You have a "simple" workload and it will remain simple
- When you can lose up to 5 minutes of transactions
- Replication lag is acceptable
- When failover could take more than 1 minute
- When you need to offload "Operations"
  - DBA tasks will remain
- No need to use tools with SUPER priv requirements
- Scale reads for specific events, i.e. Black Friday

You might choose Aurora when

- You've double-checked all Aurora limitations
- You have a highly concurrent workload
- When you can lose up to 5 minutes of transactions
- When you need to offload "Operations"
    - DBA tasks will remain
- No need to use tools with SUPER priv requirements
- Scale reads for specific events, i.e. Black Friday

You might roll your own when

- You have "Operations" resources and expertise
- You value flexibility
- You want to use tools that
  - use the SUPER priv
  - read binary logs
- You don't want to be locked in to a vendor



Q&A

Use code 10ATTENDPL18 for 10% discount!

- Premier Open Source Database Conference
- MySQL, MariaDB, MongoDB, PostgreSQL, Containers, Cloud
- HA, IoT, Scalability, Monitoring, Performance and More
- Learn and Network with Experts in the Open Source Community

Conference speakers include companies such as:

yelp, Alibaba Cloud, aws, facebook, slack, zalando, TESLA, Microsoft, ORACLE, shopify

OPEN SOURCE DATABASE CONFERENCE

PERCONA LIVE

percona.com/live/18

April 23-25, 2018

Santa Clara, California

Database Performance Matters