
AWS
re:Invent

DAT203

MySQL Options on AWS: Self-managed, managed, serverless

Yoav Eilat

Senior Product Manager

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

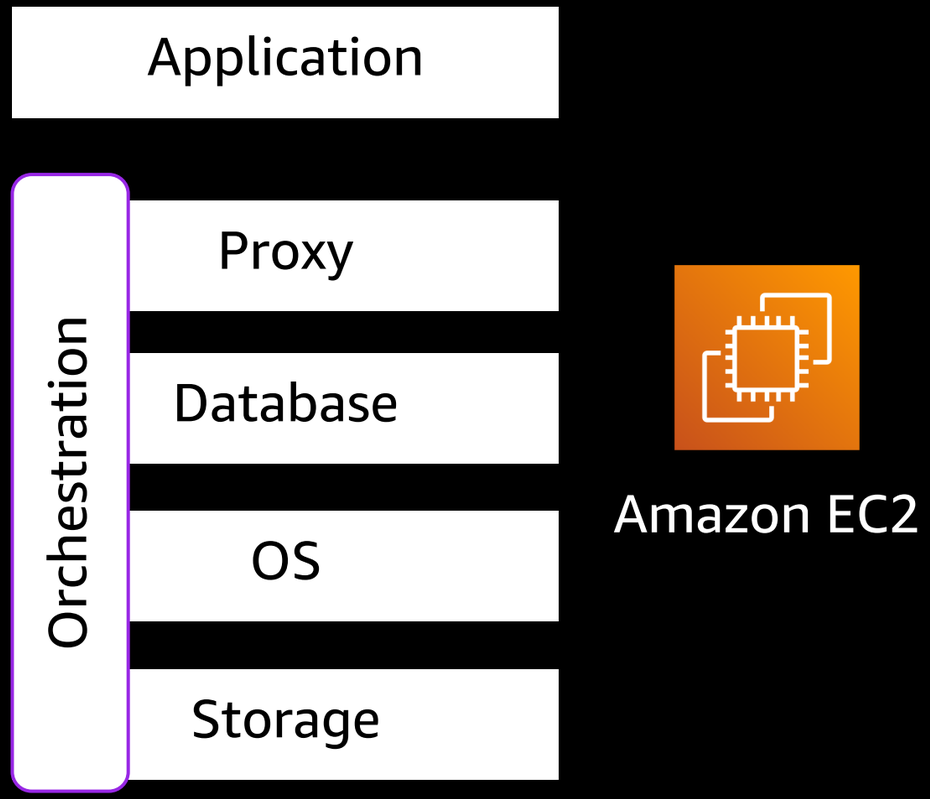
MySQL options on AWS

1. Self-managed MySQL on Amazon Elastic Compute Cloud (Amazon EC2)
2. Amazon Relational Database Service (Amazon RDS) for MySQL & Amazon RDS for MariaDB
3. Amazon Aurora MySQL – provisioned and serverless
4. Hybrid cloud: Amazon RDS for MySQL on VMware
5. Hybrid cloud: Amazon RDS for MySQL on AWS Outposts

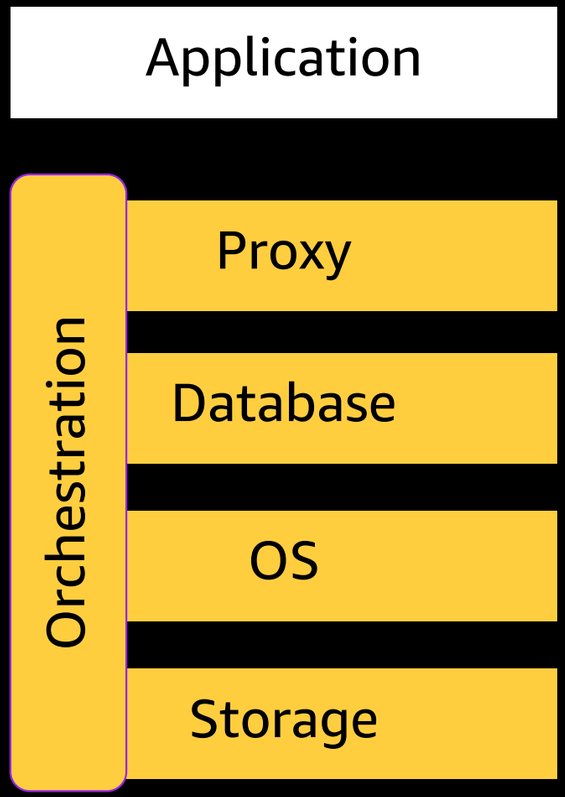
Deep dive: which option to use when

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS offers five options for running MySQL



Amazon EC2





Amazon RDS
MySQL & MariaDB



Amazon RDS
on VMware



Amazon
Aurora

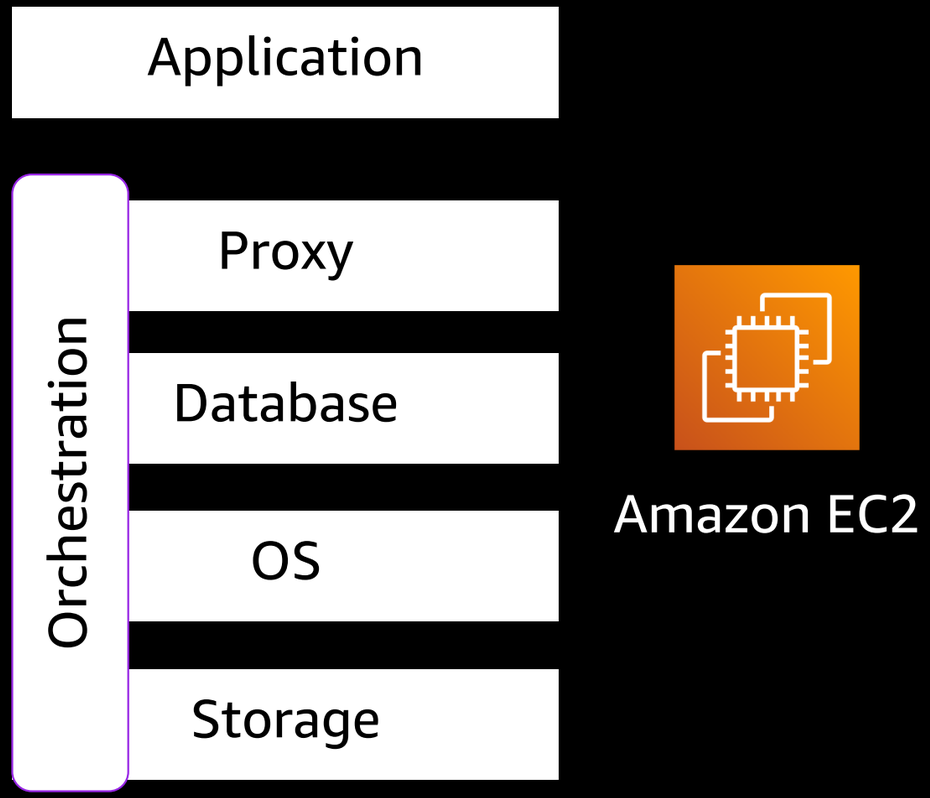


Amazon RDS
on Outposts

Self-managed

AWS-managed

Benefits of self-managed MySQL on Amazon EC2



Amazon EC2

Full administrative control

Software-controlled networking and security

Choice of ephemeral or Amazon Elastic Block Store (Amazon EBS) storage

Custom hardware and software configurations

Amazon Relational Database Service (Amazon RDS)

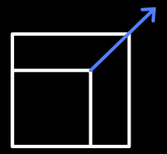
MANAGED RELATIONAL DATABASE SERVICE WITH A CHOICE OF POPULAR DATABASE ENGINES

Amazon Aurora · MySQL · MariaDB · PostgreSQL · Microsoft SQL Server · ORACLE



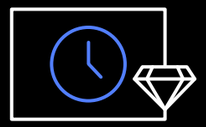
Easy to administer

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

Amazon Aurora

ENTERPRISE DATABASE AT AN OPEN SOURCE PRICE

Delivered as a **managed** service



Speed and **availability** of high-end commercial databases

Simplicity and **cost-effectiveness** of open source databases

Drop-in **compatibility** with MySQL and PostgreSQL

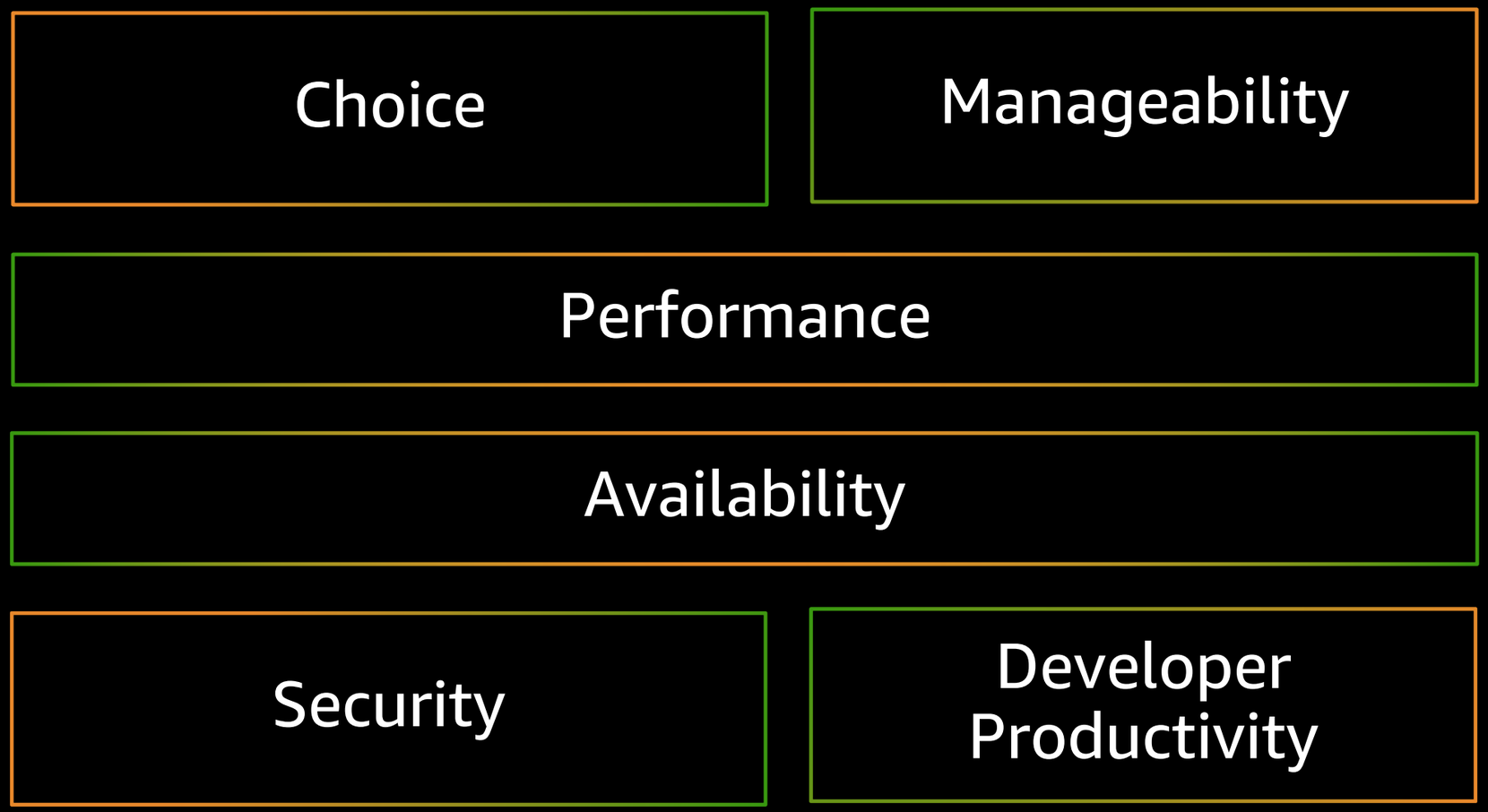
Simple **pay as you go** pricing

Which MySQL option to use when?

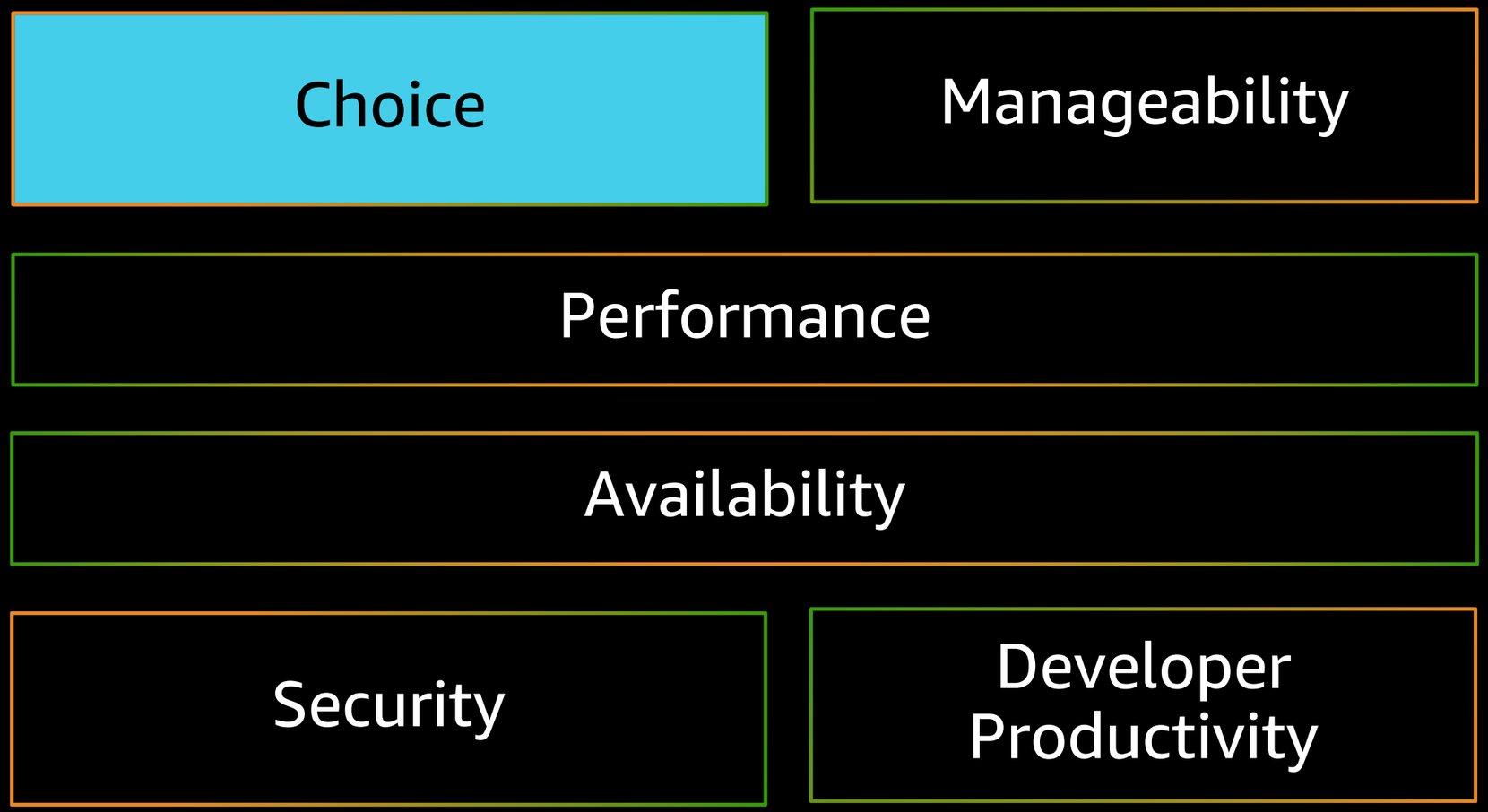


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What do you want from your database service?



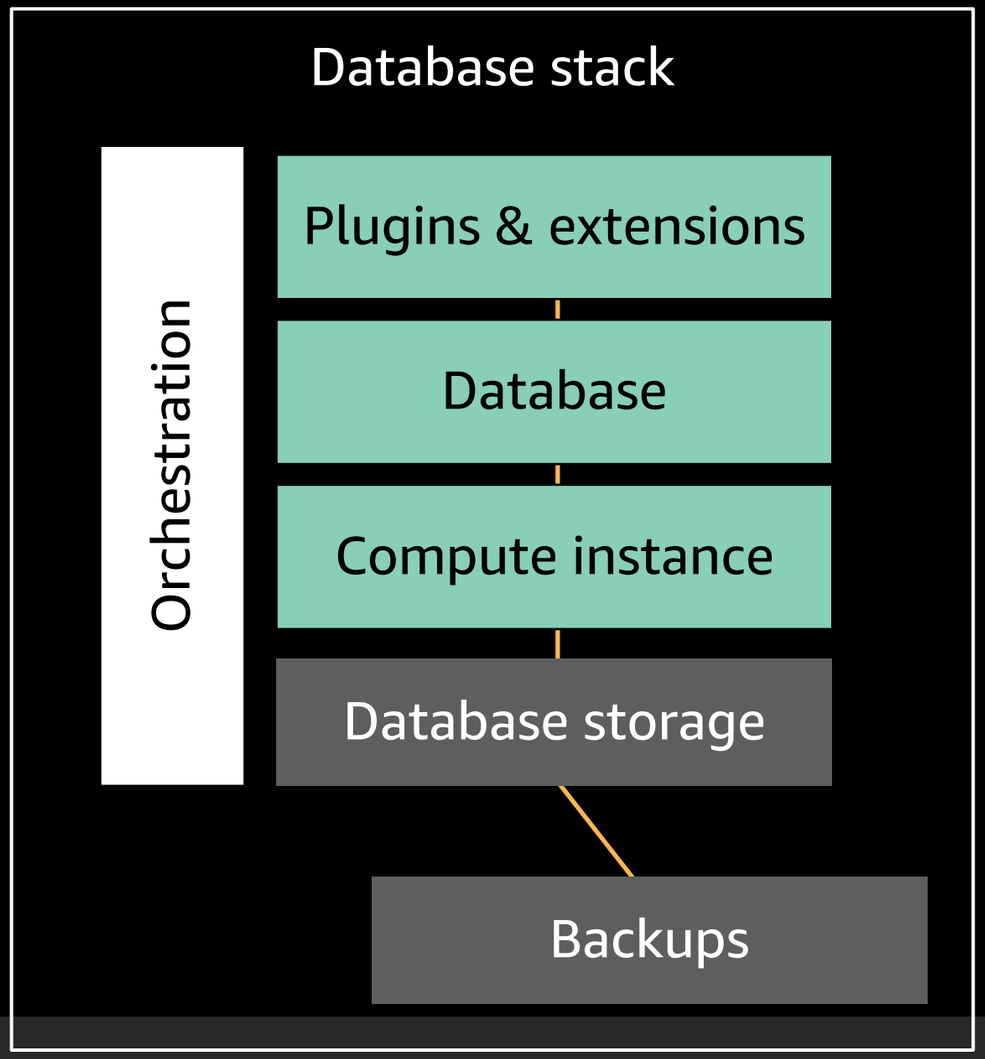
What do you want from your database service?



Amazon EC2 (self-managed) customizations

SUITABLE FOR SOME EDGE CASES

You're on your own →



← Install anything

← Select any database

← Select Amazon EC2 instance

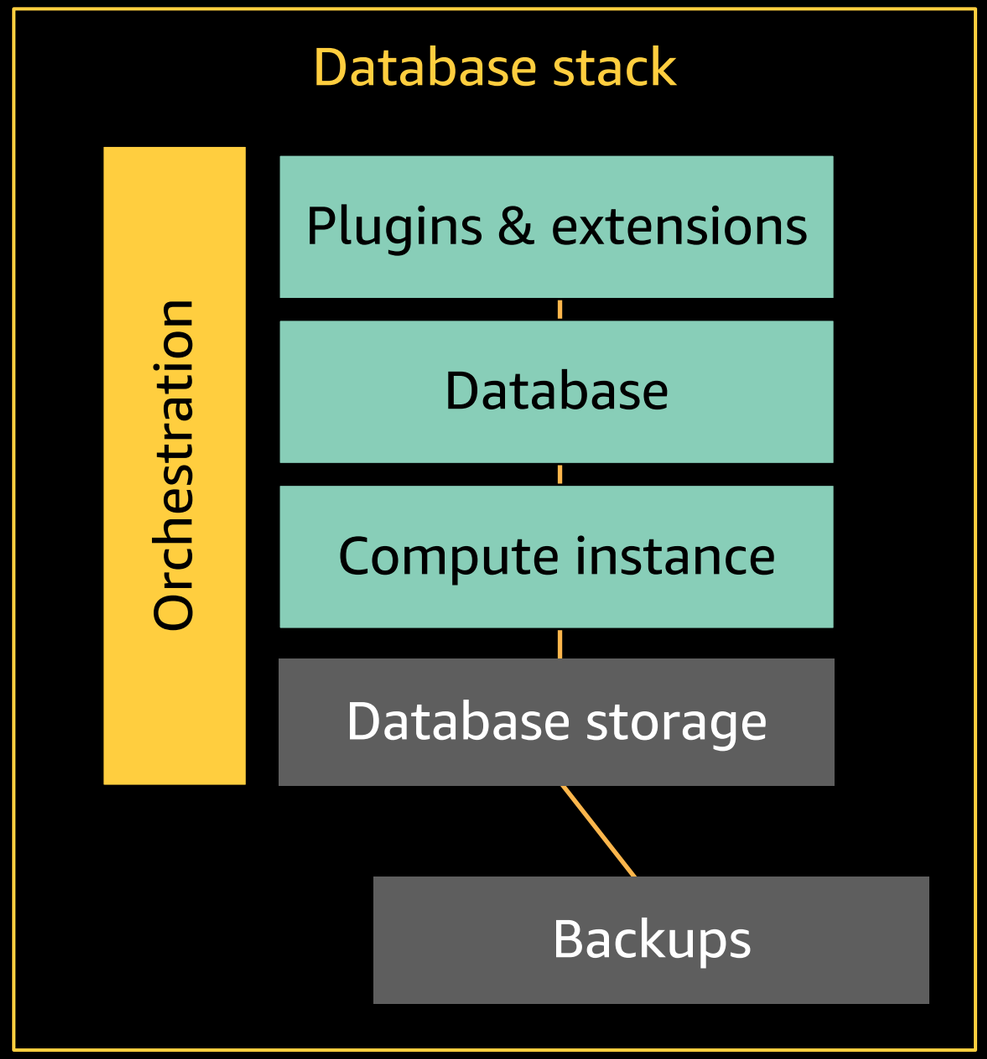
← Select any AWS storage

← You're on your own

Amazon RDS (AWS-managed) customizations

SUITABLE FOR ALMOST ALL MYSQL CUSTOMERS

Via console, CLI, API →



← Configure via parameter groups

← Select current database version

← Select supported family

← Choose storage type and size

← Set up backups and manual snapshots

Choice of database engines and instance types

INCLUDING AMAZON RDS GRAVITON2 INSTANCES

MySQL versions: 5.5, 5.6, 5.7, 8.0

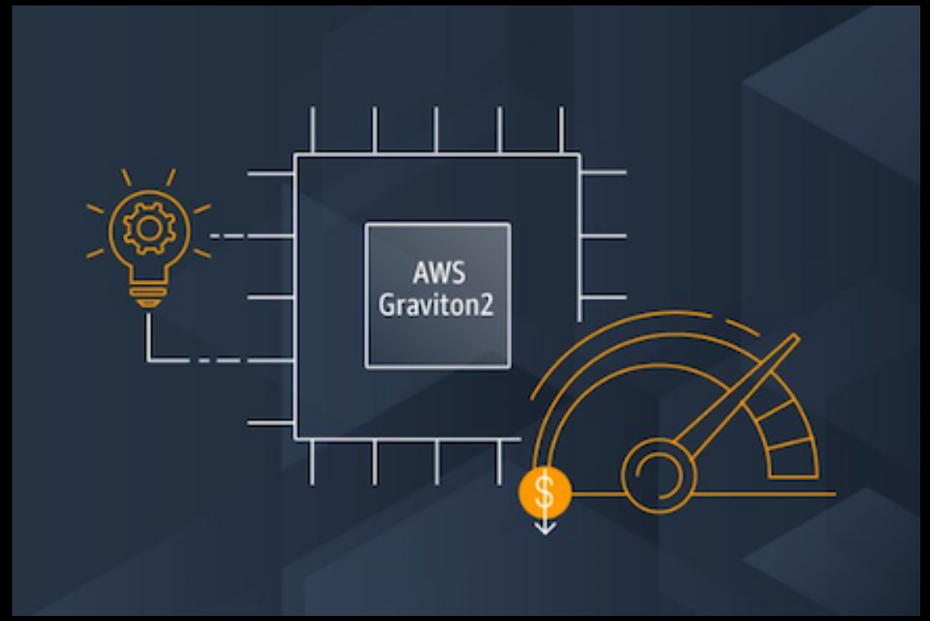
NEW!

MariaDB versions: 10.0, 10.1, 10.2, 10.3, 10.4

Aurora MySQL versions: 5.6, 5.7

NEW! NEW!

Instance families including R5, M5, T3, R6G, M6G



R6G and M6G instances powered by AWS Graviton2 –

- Available for Amazon RDS for MySQL and Amazon RDS for MariaDB
- 13% better performance and 27% better price/performance for Amazon RDS for MySQL

Amazon RDS for MySQL hybrid options

Amazon RDS on VMware

- Familiar Amazon RDS capabilities including Amazon RDS Console, API, and CLI as in the AWS regions
- Support for read replicas and RDS Proxy
- For running on VMware vSphere on-premises



Amazon RDS on VMware

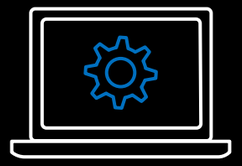
Amazon RDS on AWS Outposts

- Familiar RDS capabilities including RDS Console, API, and CLI as in the AWS regions
- For running on AWS Outposts on-premises



Amazon RDS on Outposts

Amazon RDS for MySQL hybrid use cases



Latency

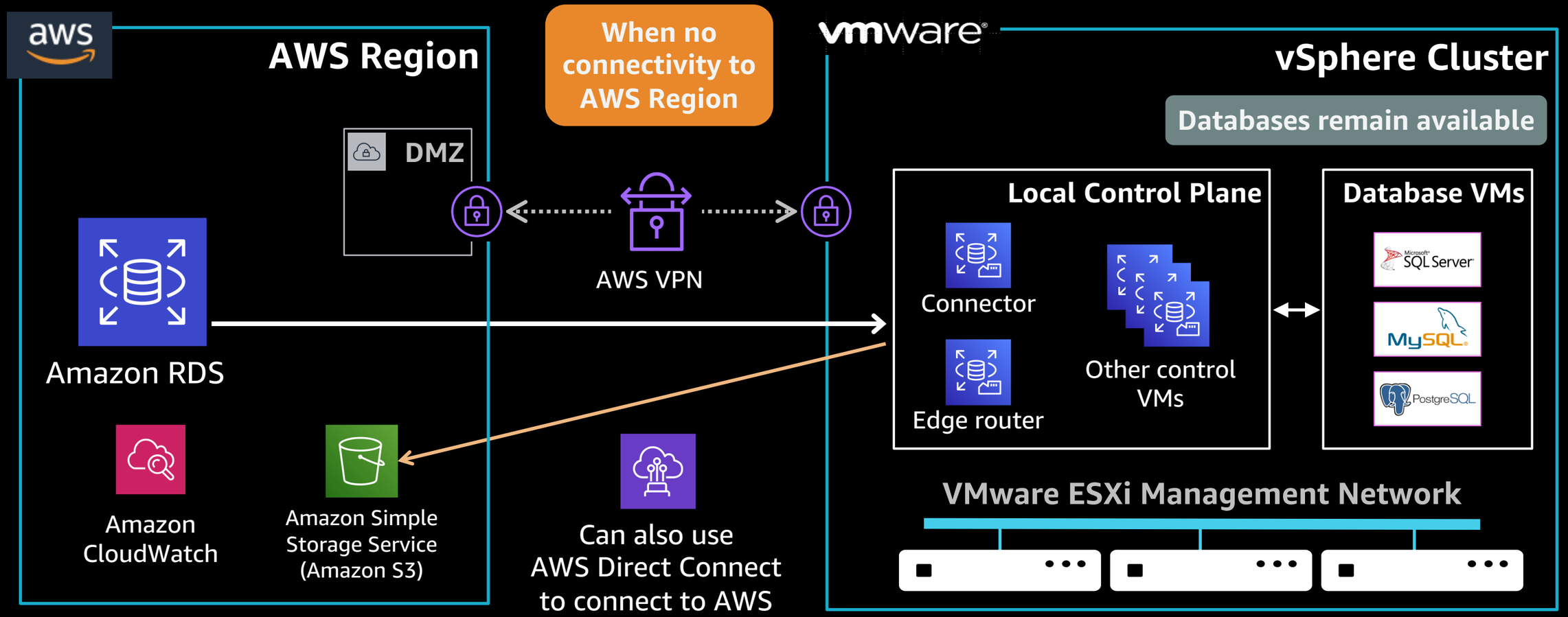
- Applications and processes sensitive to network and disk latency
- Financial services applications such as trading and brokerage
- Security and fraud applications that require a quick response time



Residency

- Regulations dictate that data and infrastructure reside inside the country
- Contracts specify where applications are deployed
- You’re note-ready to move to AWS regions for information security or other reasons

How Amazon RDS on VMware works

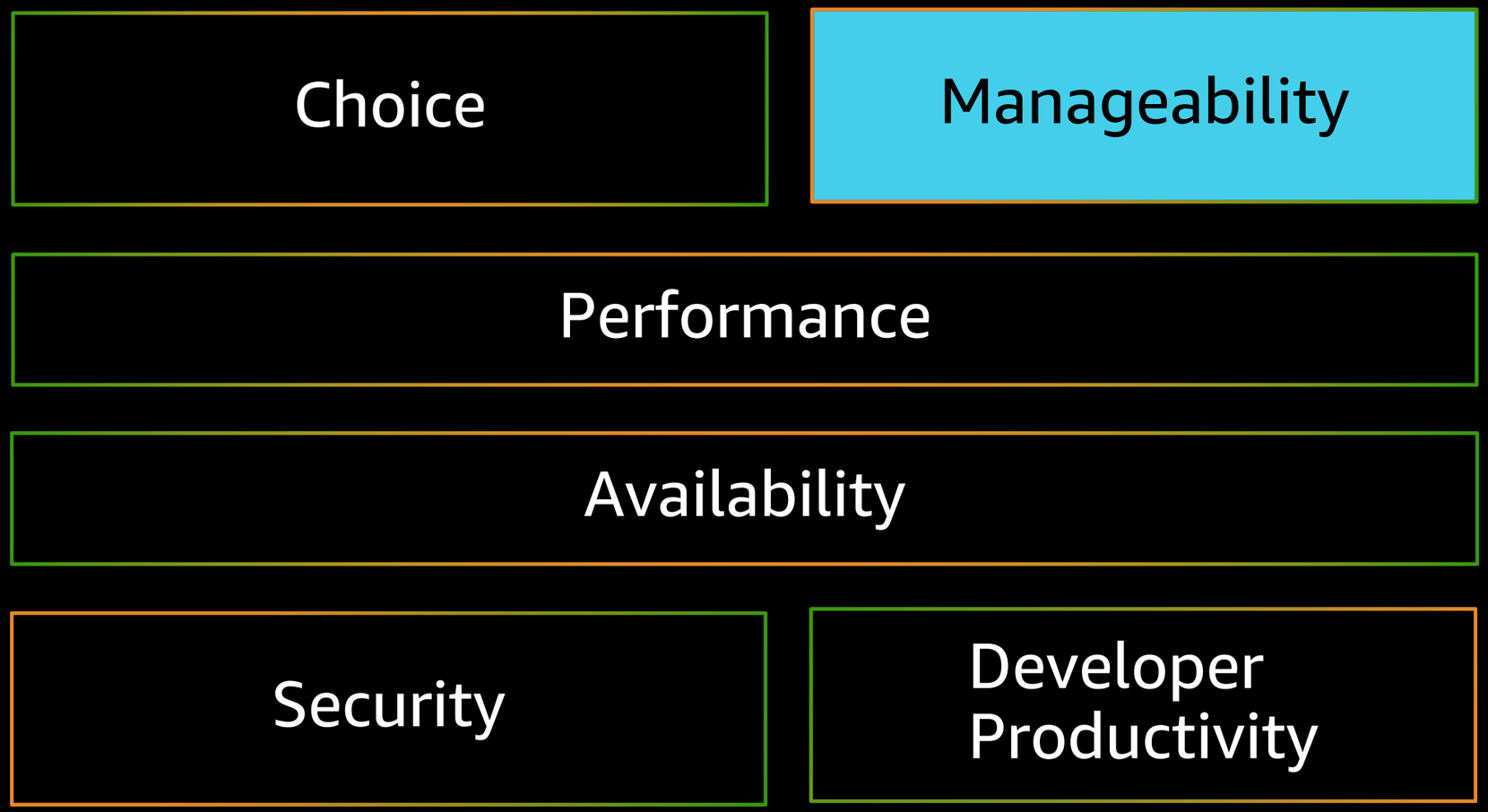


Amazon RDS connector establishes secure VPN connection with AWS region

Amazon RDS is deployed in your datacenter through VPN tunnel

Amazon RDS provisions and manages on-premises instances, including automated backups and restores

What do you want from your database service?



Amazon RDS manageability highlights

- Automated OS and database upgrades
- Push-button scaling
- Managed binlog replication
- Log upload to Amazon CloudWatch Logs
- Per-second billing
- Auto-scaling storage up to 64TiB and 80K PIOPS (Aurora: 128TiB) **NEW!**
- Snapshot export to Amazon S3 **NEW!**



Amazon RDS

Amazon RDS for MySQL backups, snapshots, and point-in-time restore

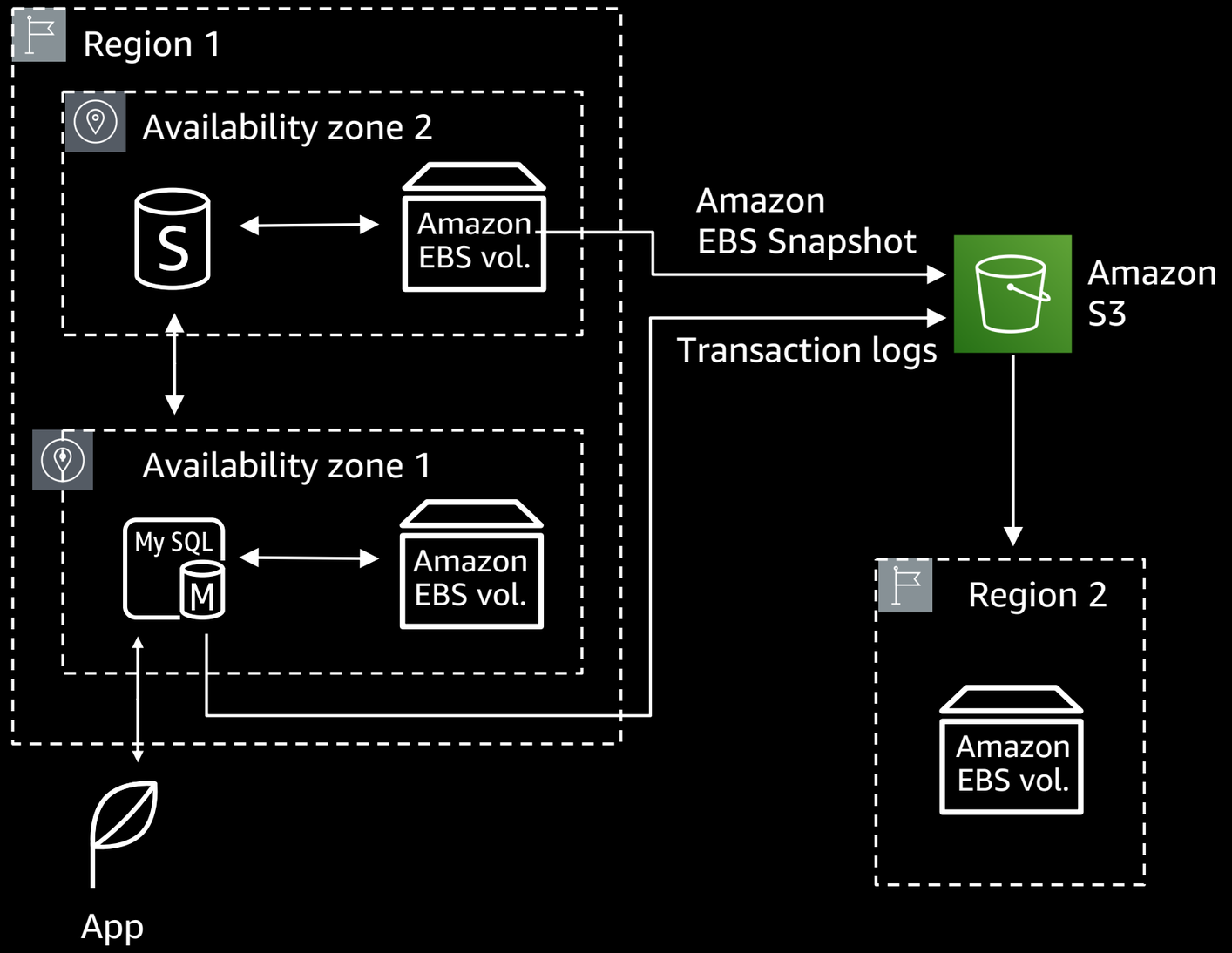
Two options—automated backups and manual snapshots

Amazon EBS snapshots stored in Amazon S3

Transaction logs stored every 5 minutes in Amazon S3 to support point-in-time recovery

No performance penalty for backups

Snapshots can be copied across regions or shared with other accounts



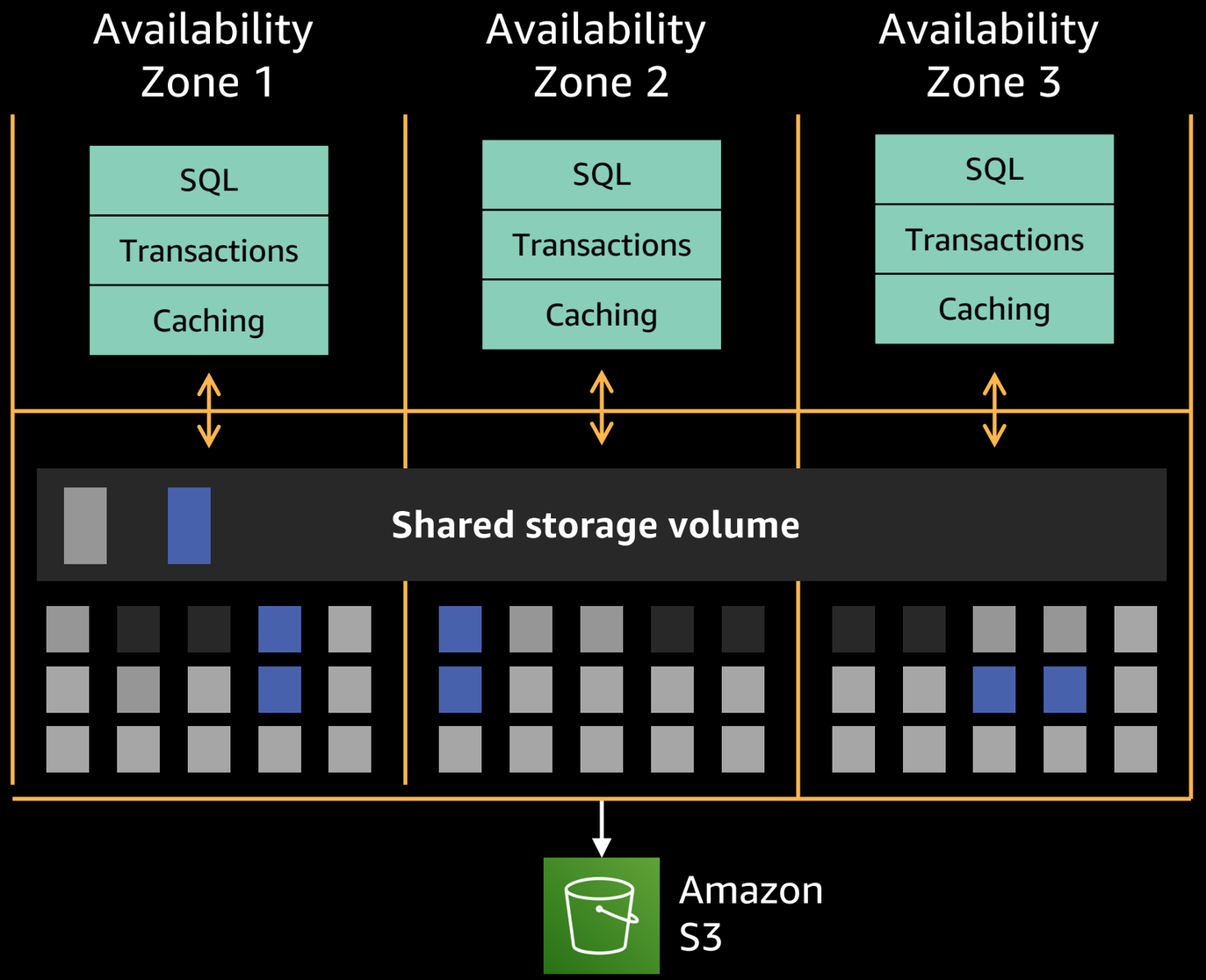
Aurora scale-out, distributed architecture

Purpose-built, log-structured, distributed storage system designed for databases

Storage volume is striped across hundreds of storage nodes distributed over 3 different availability zones

Six copies of data, two copies in each availability zone to protect against AZ+1 failures

Continuous backups to Amazon S3 and faster restores



Amazon Aurora Serverless

Starts up on demand, shuts down when not in use

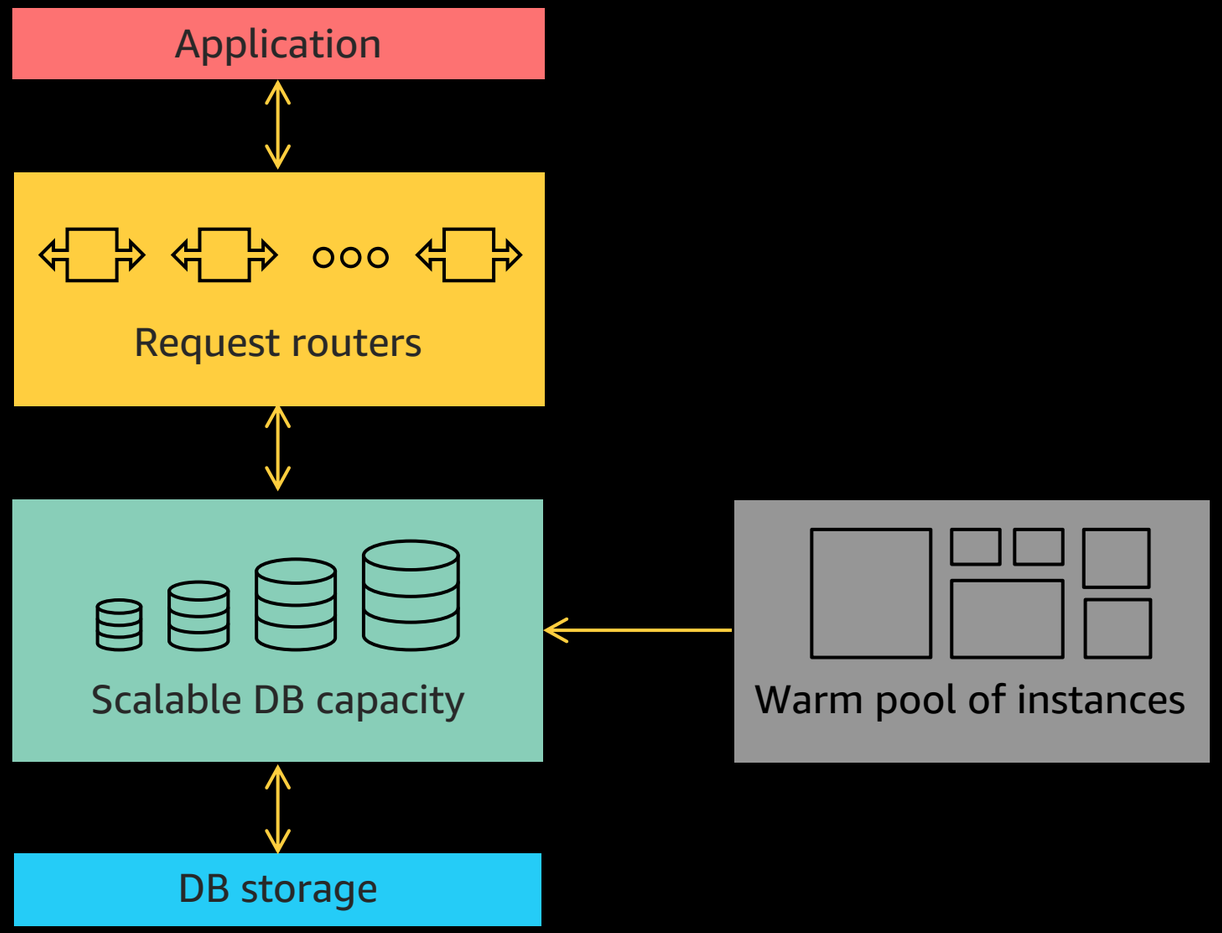
Adjusts capacity automatically

No application impact when scaling

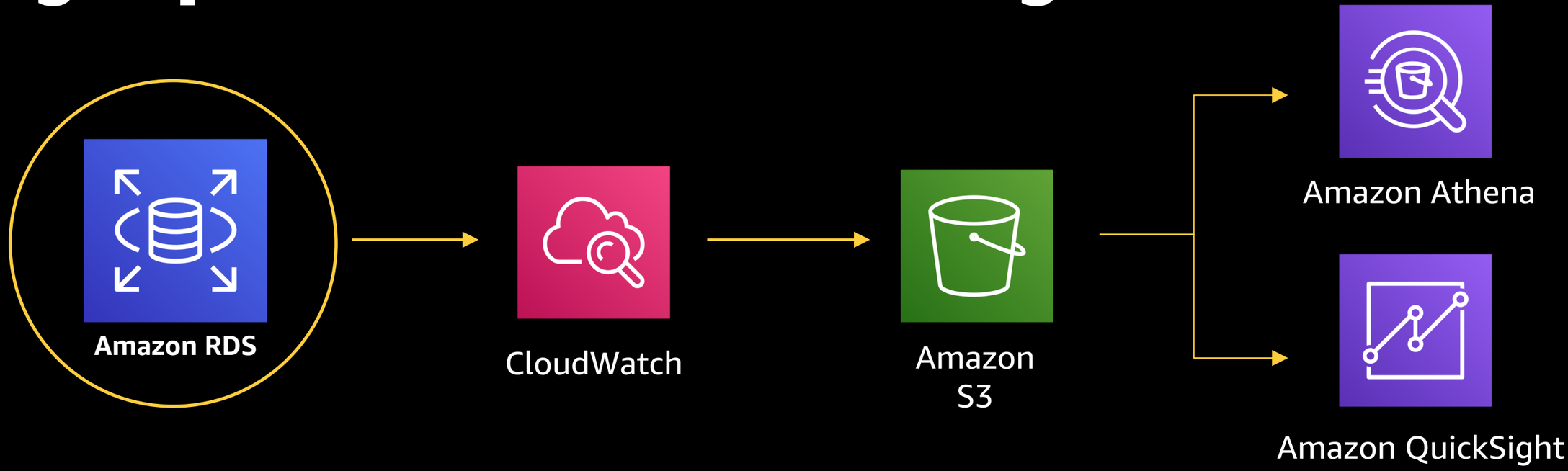
Pay per second, one-minute minimum

Great for infrequently-used, unpredictable, or cyclical workloads

Data API for running queries over HTTP



Log exports to CloudWatch Logs



- Continuously monitor activity in your DB clusters by sending audit, general, slow query, and error logs to CloudWatch Logs
- Export to Amazon S3 for long term archival; analyze logs using Athena; visualize logs with QuickSight

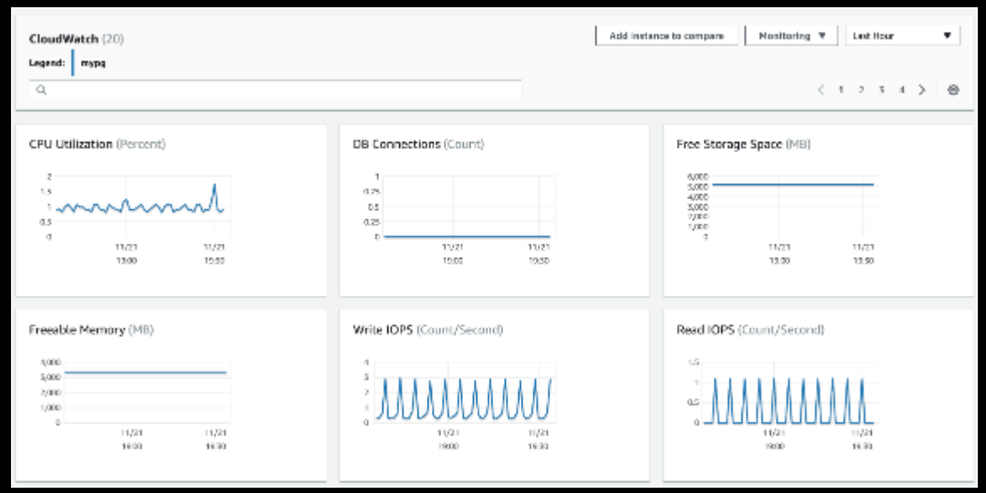
Search: Look for specific events across log files

Metrics: Measure activity in the Aurora DB cluster

Visualizations: Create activity dashboards

Alarms: Get notified or take actions

Monitor Amazon RDS with CloudWatch



CloudWatch metrics

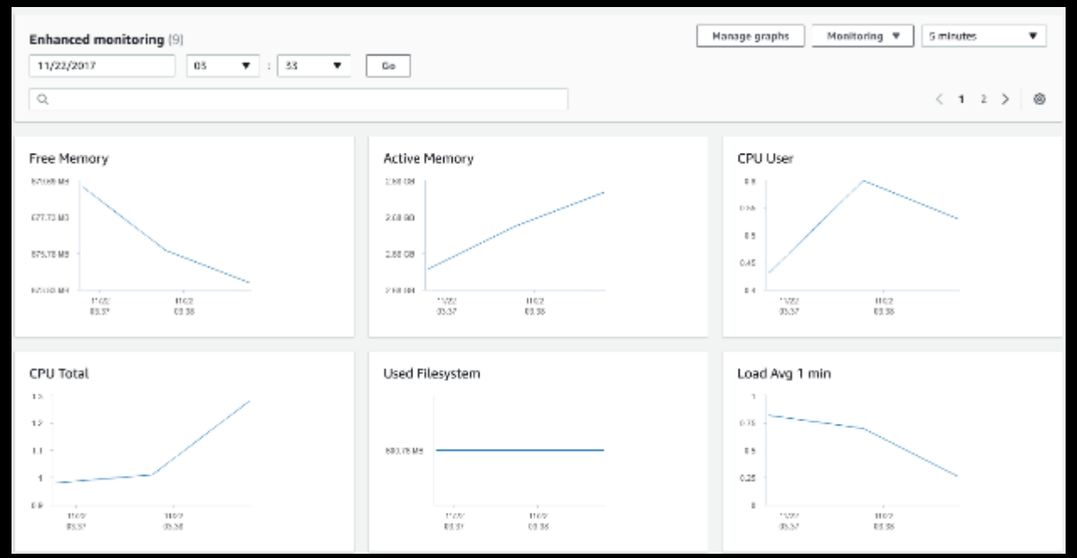
- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools

Monitor Amazon RDS with CloudWatch



CloudWatch metrics

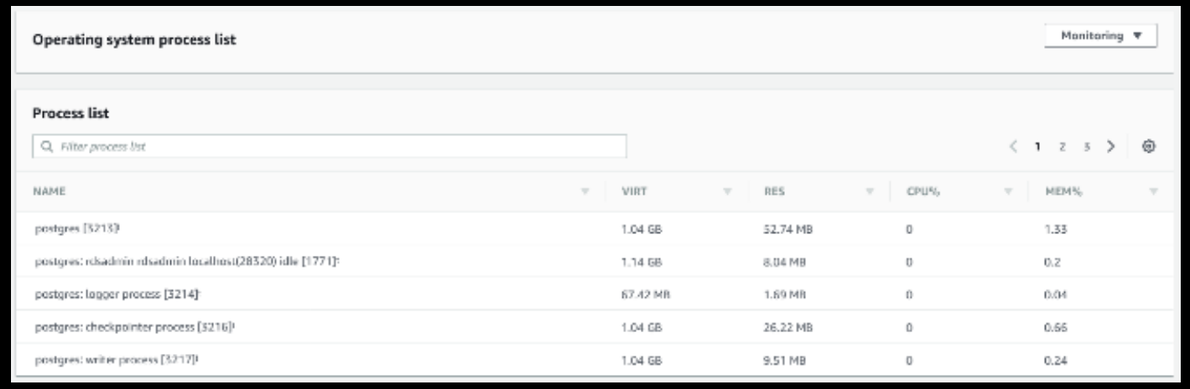
- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools

Monitor Amazon RDS with CloudWatch



CloudWatch metrics

- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools

Performance Insights

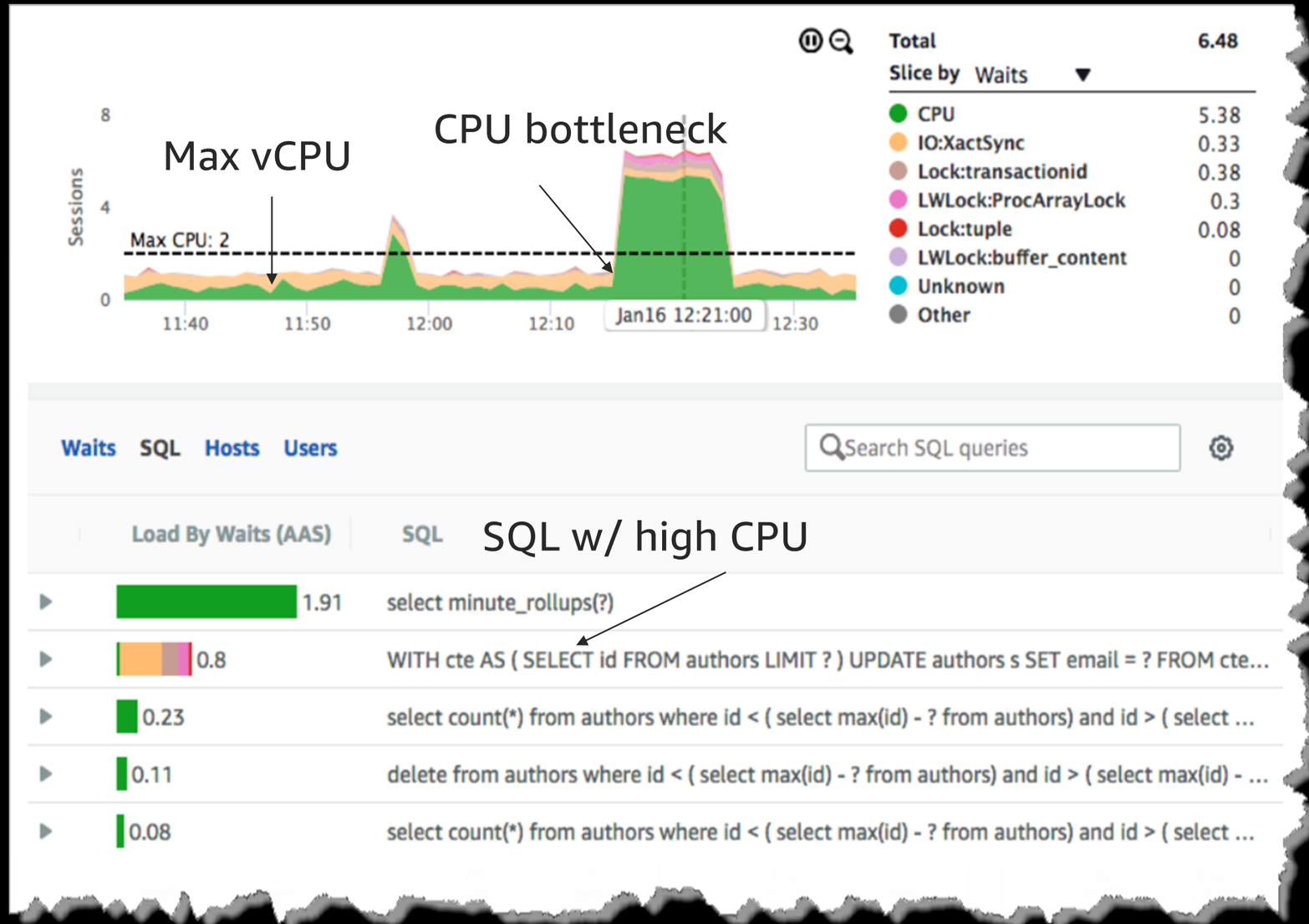
Dashboard showing database load
- Easy – e.g. drag and drop
- Powerful – drill down using zoom in

Identifies source of bottlenecks
- Sort by top SQL
- Slice by host, user, wait events

Adjustable time frame
- Hour, day, week, month
- Up to 2 years of data; 7 days free

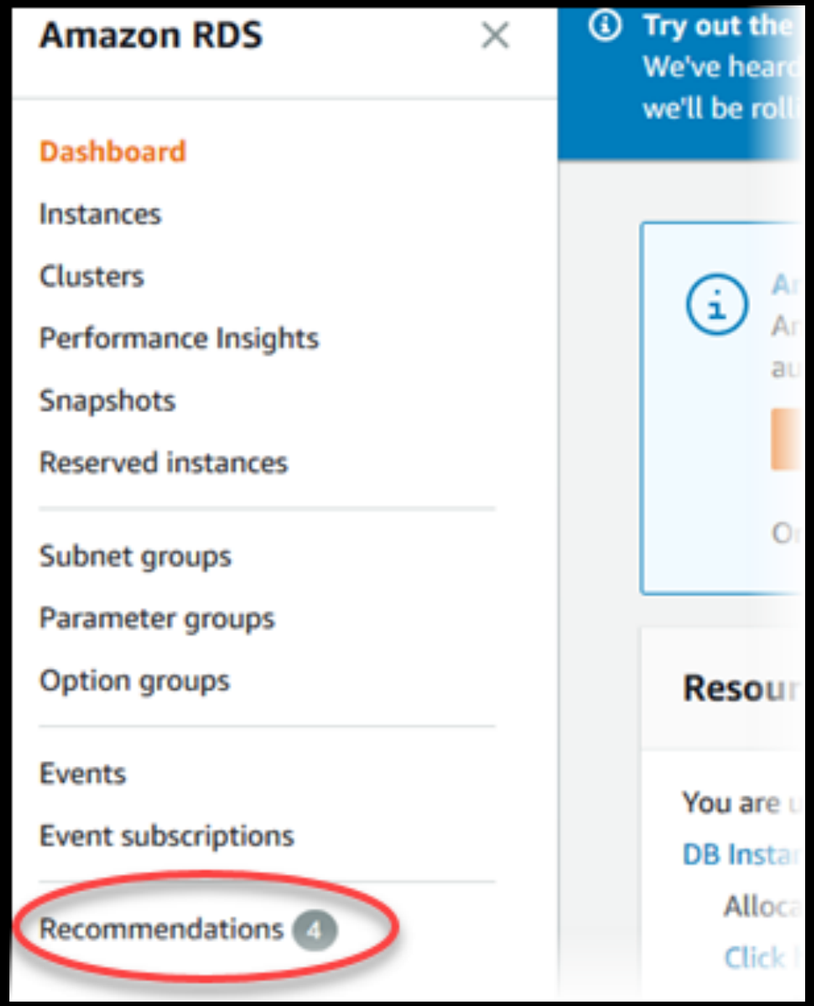
Support for MySQL and MariaDB
- SQL-level metrics

NEW!

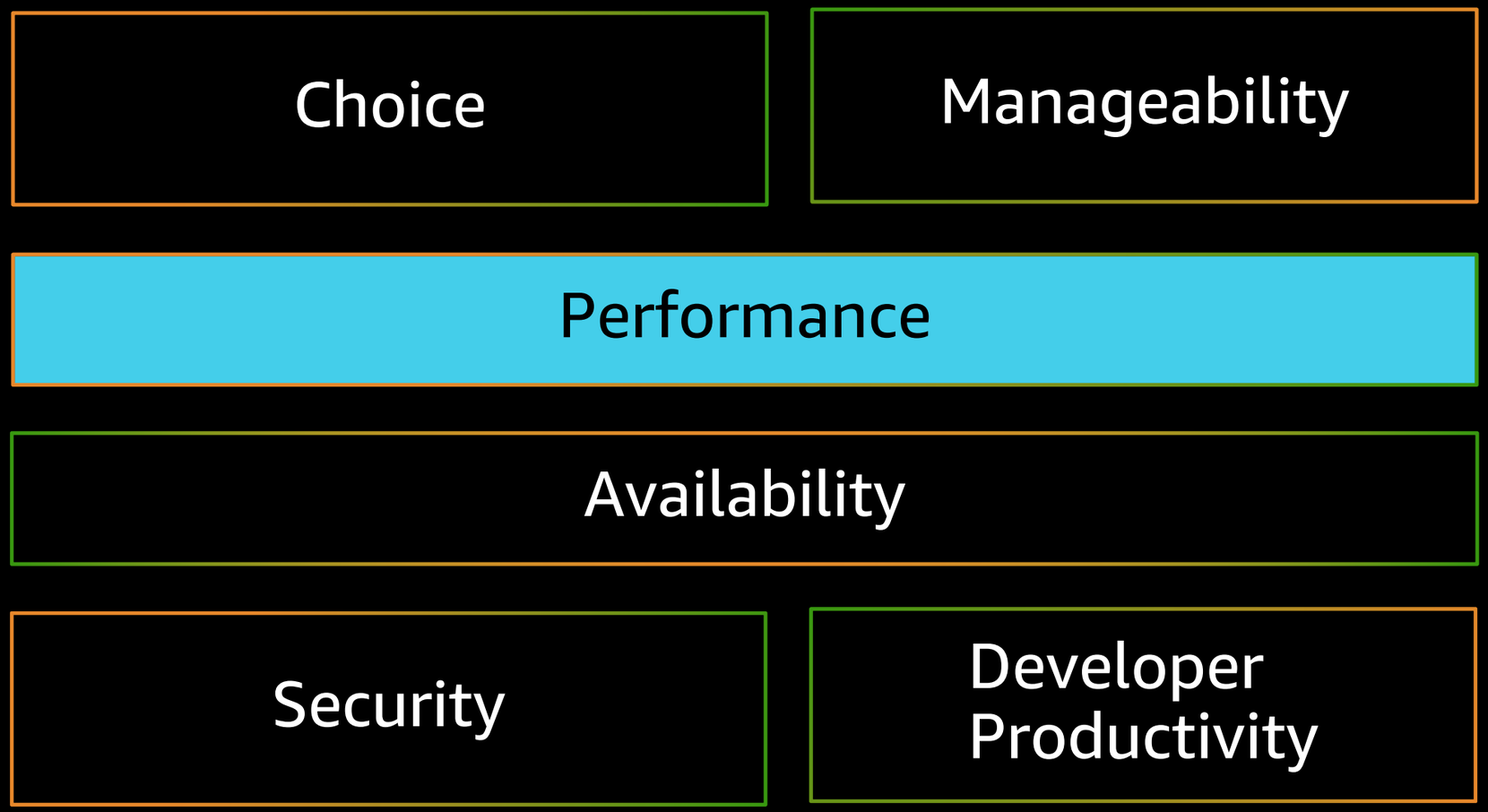


Recommendations

- Best practice recommendations from AWS
- Parameter recommendations –
  - Non-default custom memory parameters, change buffering enabled, logging to table
- Performance-related recommendations –
  - Query cache enabled, previous generation instance
- Aurora cluster recommendations –
  - One instance, one AZ, outdated DB version



What do you want from your database service?



Read scaling with RDS read replicas

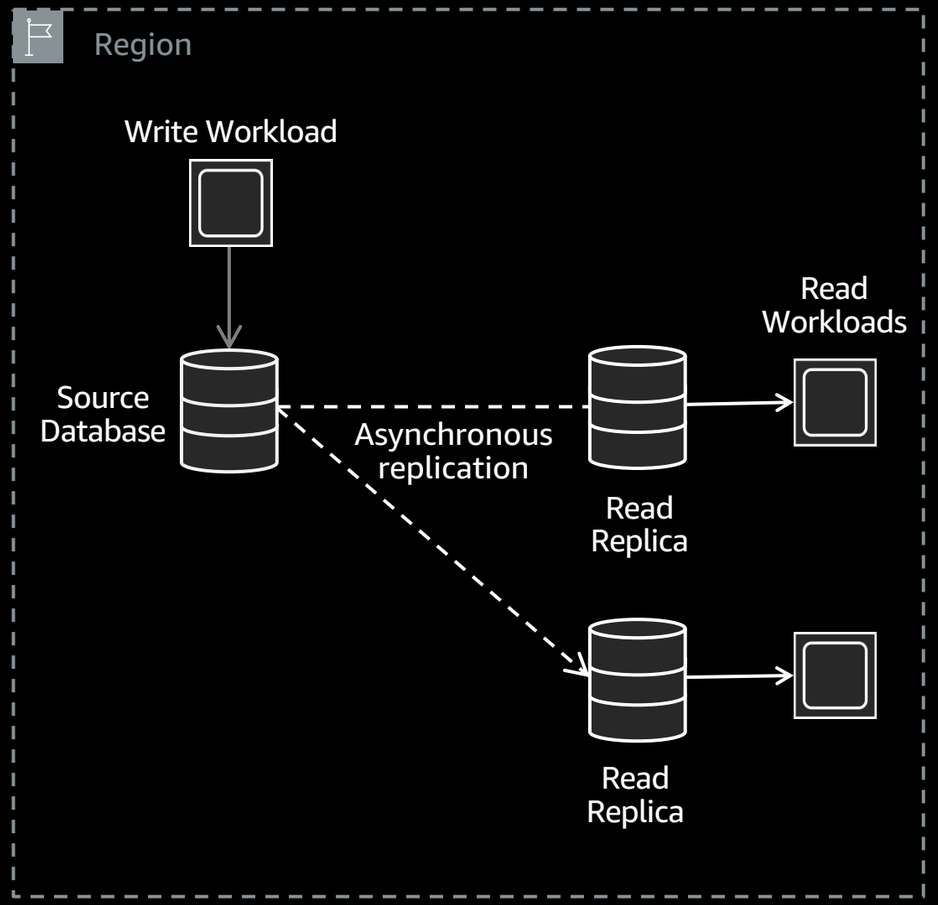
Read replicas offload reads from the source database

Create up to five replicas per source database

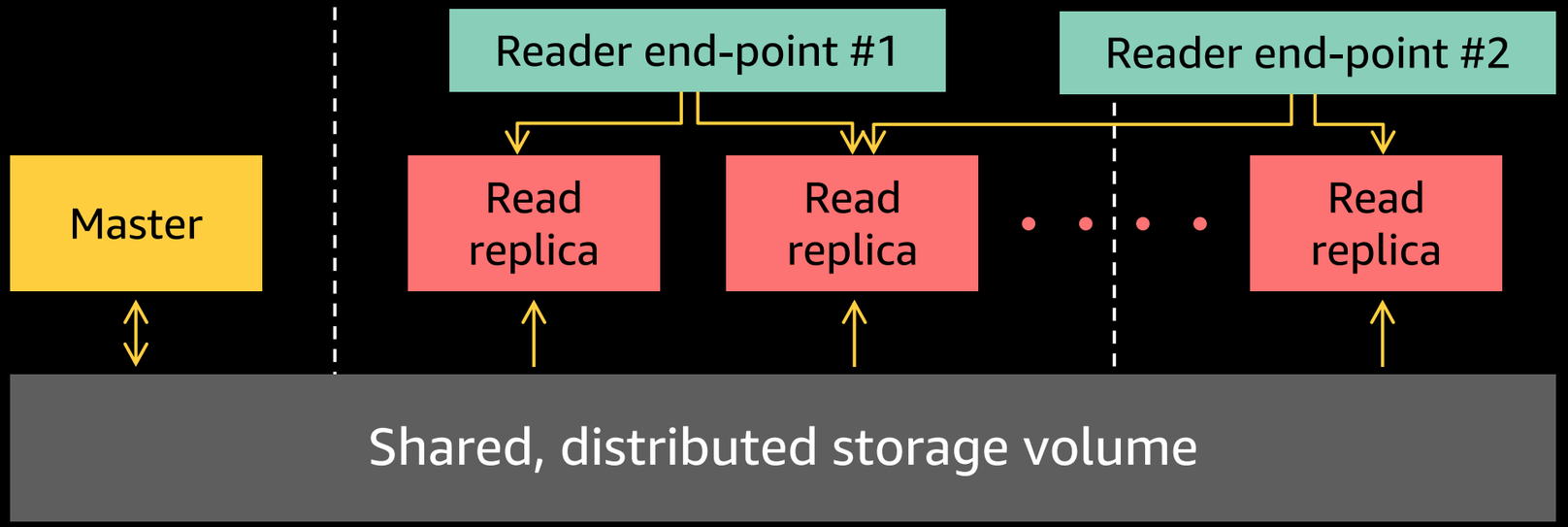
Binlog replication (file position or GTID)

Upgrades independent of the source database

Each read replica itself can be multi-AZ



Aurora read replicas and database auto-scaling



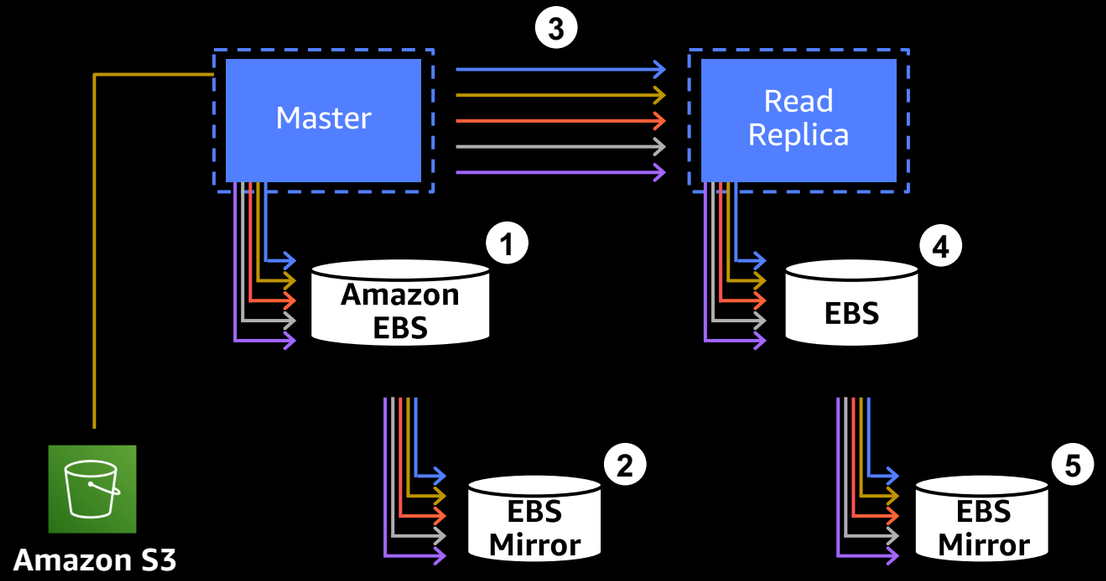
Up to 15 promotable read replicas across multiple availability zones

Redo log-based (physical) replication leads to low-replica lag—typically < 10ms

Custom reader end-point with load balancing and auto-scaling

MySQL vs. Aurora I/O profile

MYSQL with replica

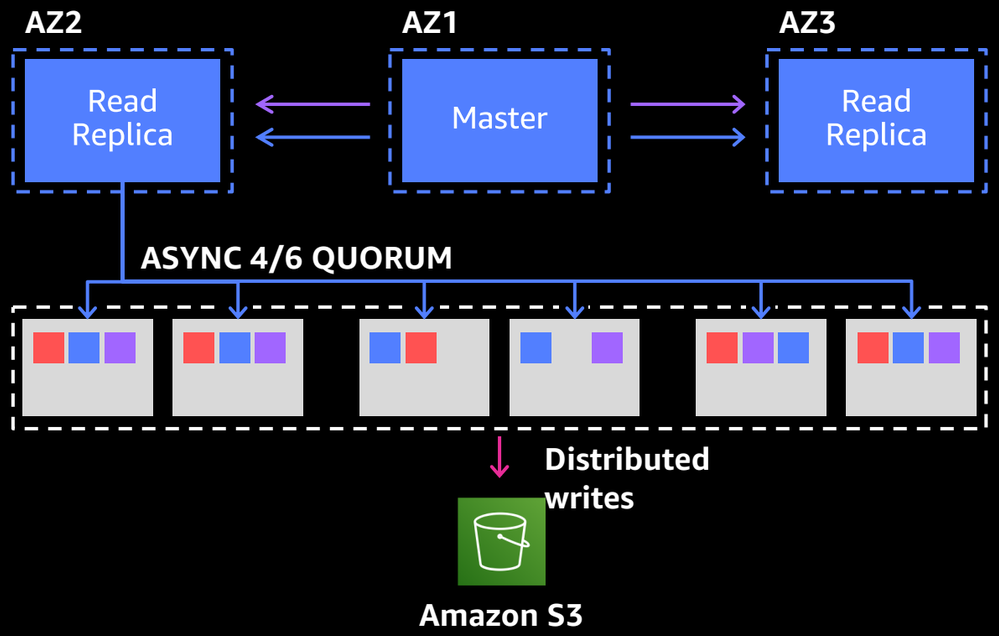


MySQL I/O profile for 30 min Sysbench run

0.78MM transactions

7.4 I/Os per transaction

Amazon Aurora



Aurora IO profile for 30 min Sysbench run

27MM transactions **35X More**

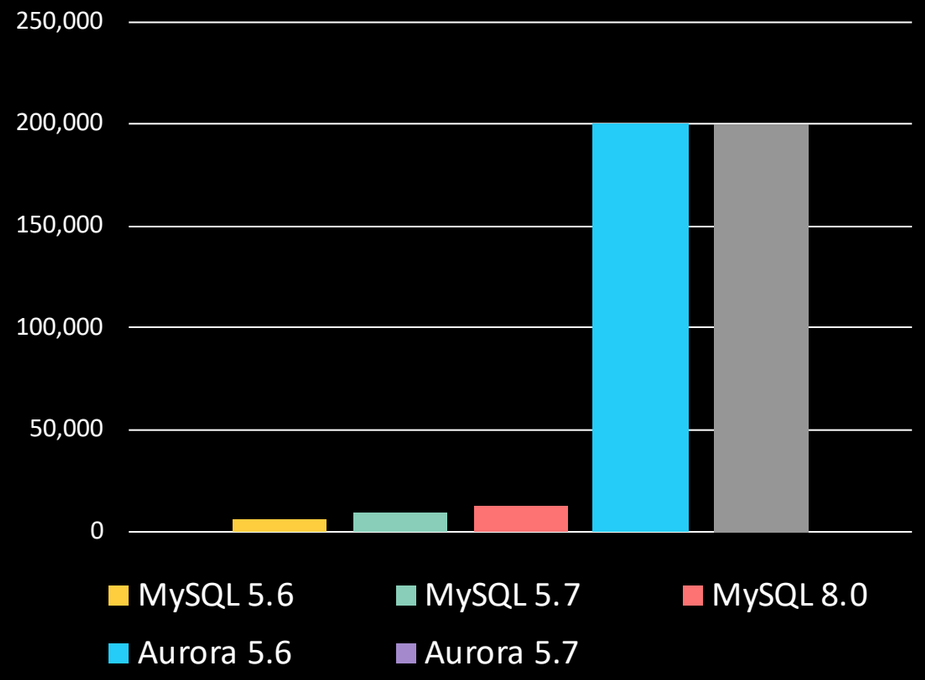
0.95 I/Os per transaction **7.7X Less**

Type of write

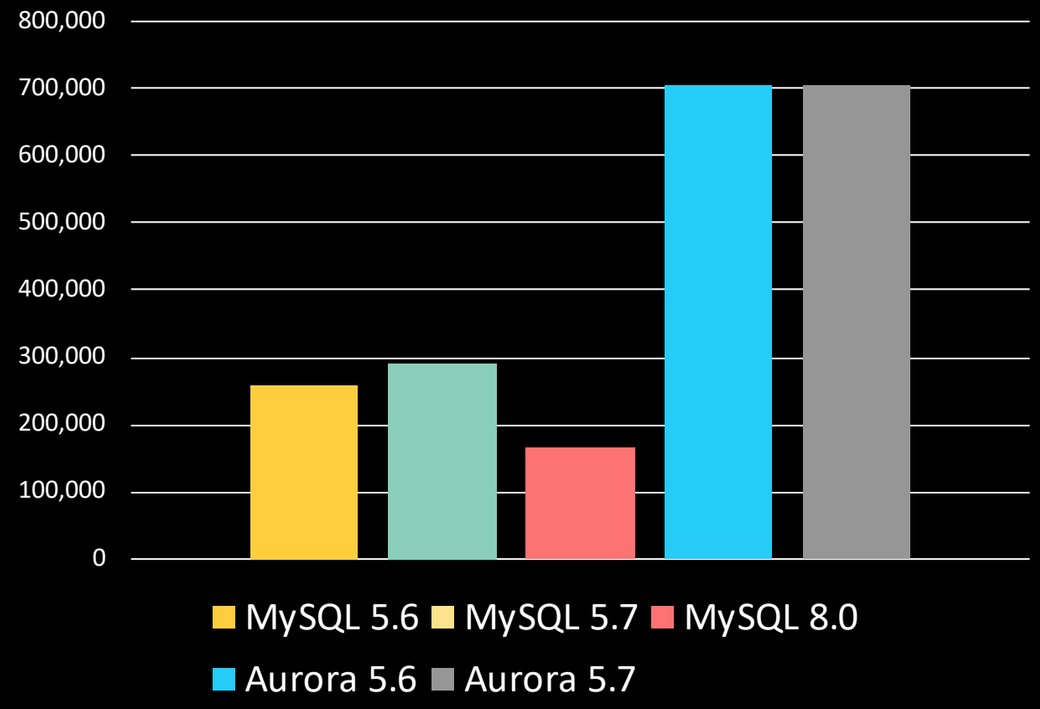
Log Binlog Data Double-write FRM files

Aurora write and read throughput

AURORA MYSQL IS 5X FASTER THAN MYSQL



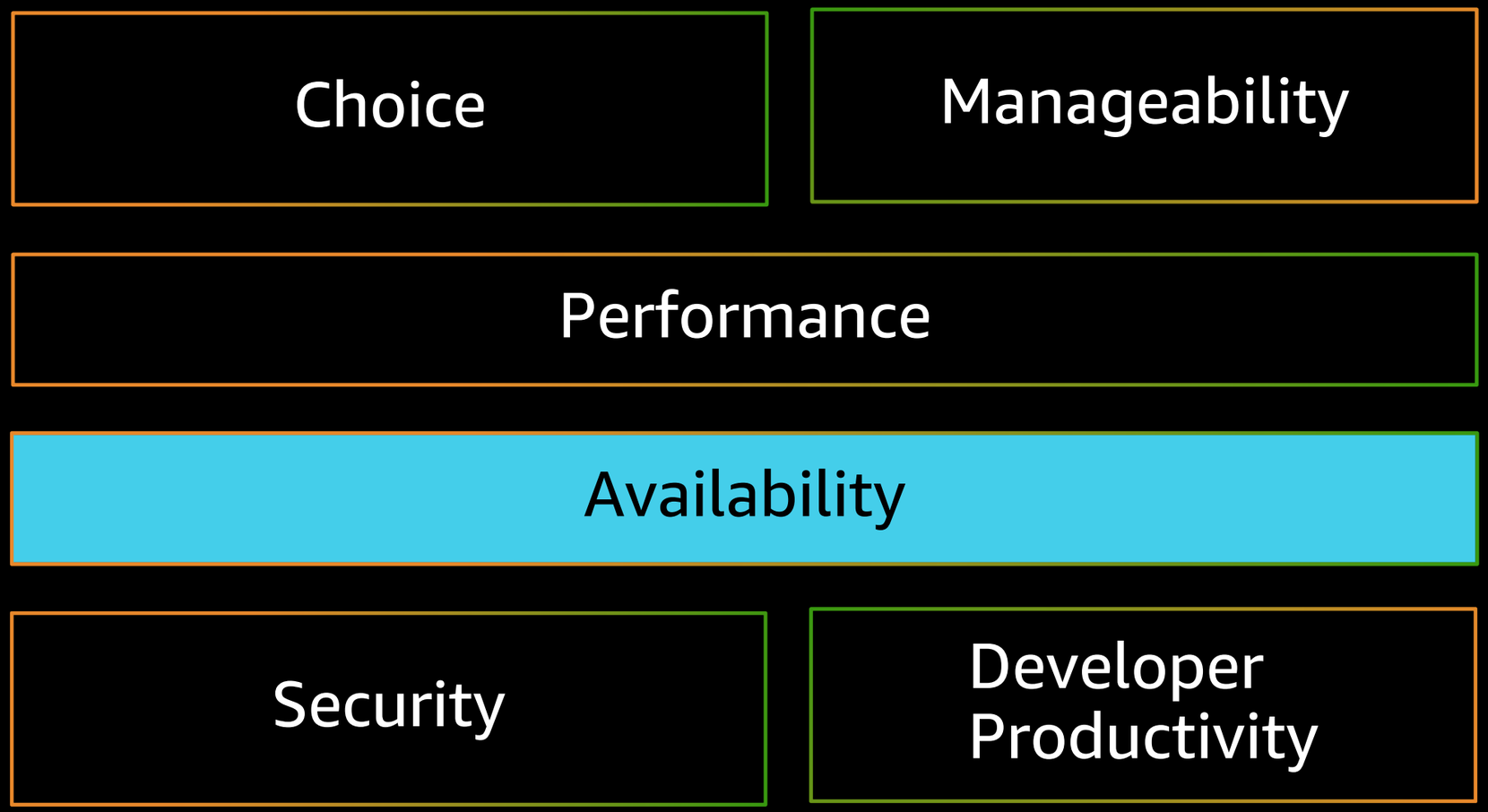
Write Throughput



Read Throughput

Using Sysbench with 250 tables and 200,000 rows per table on R4.16XL

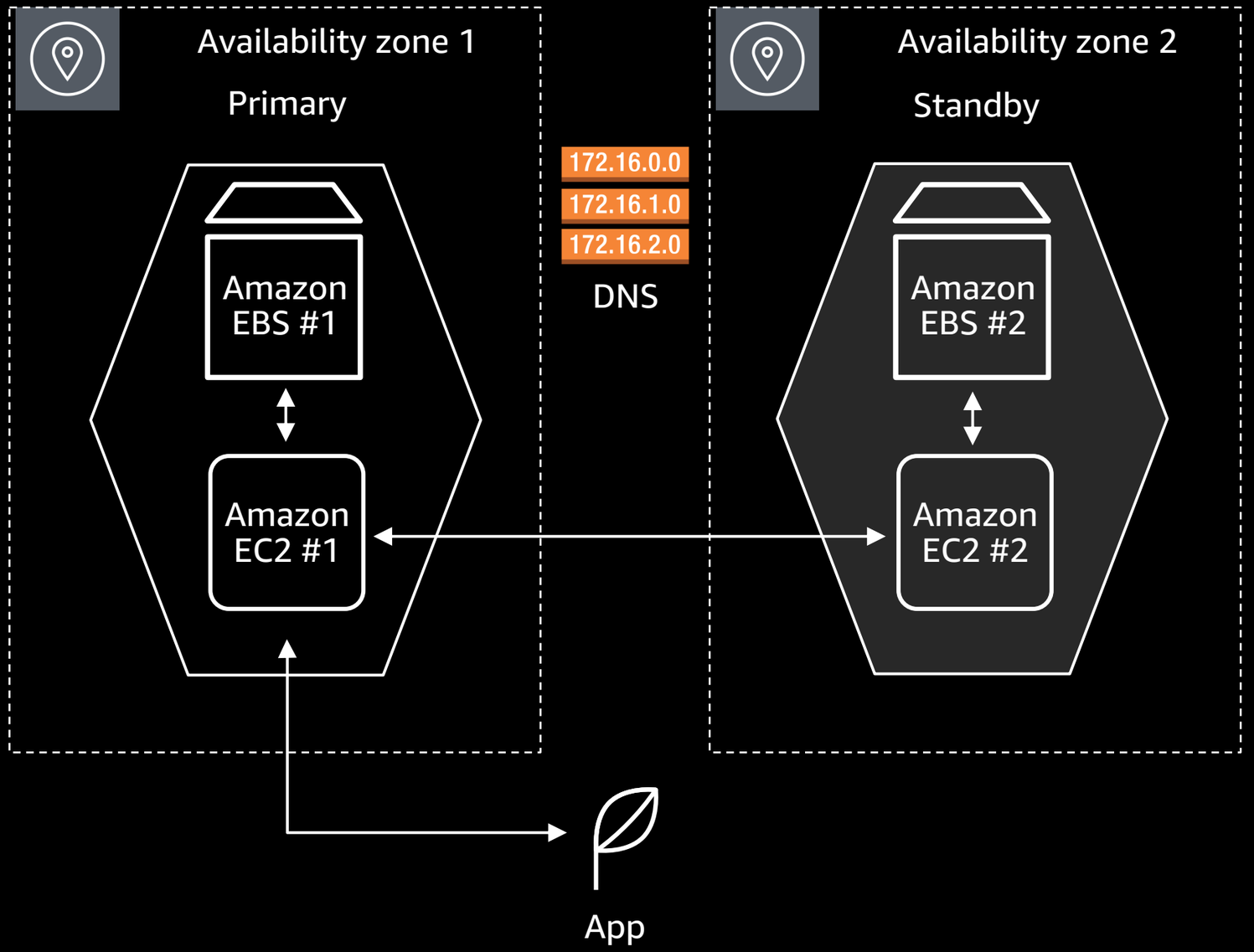
What do you want from your database service?



Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

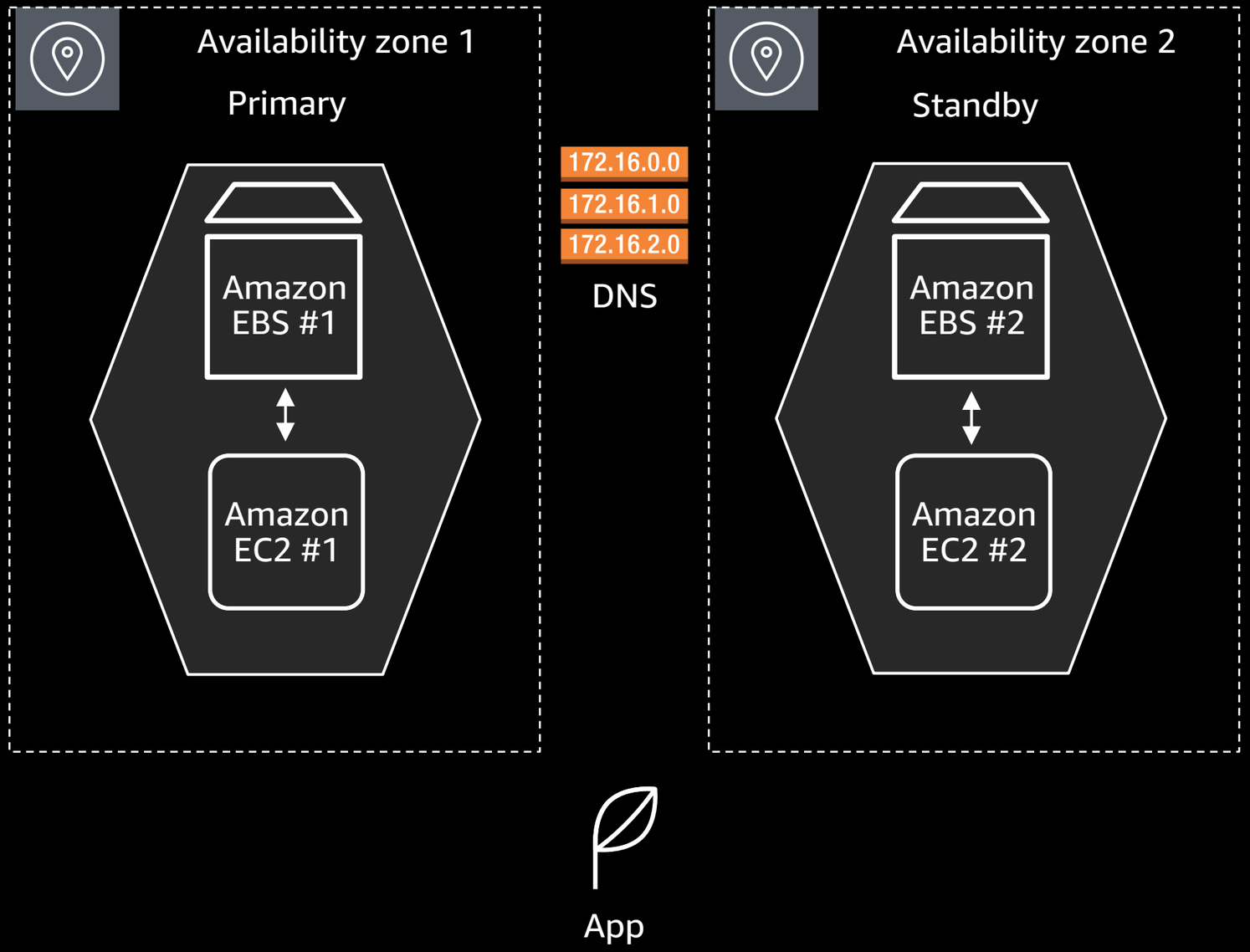


Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API



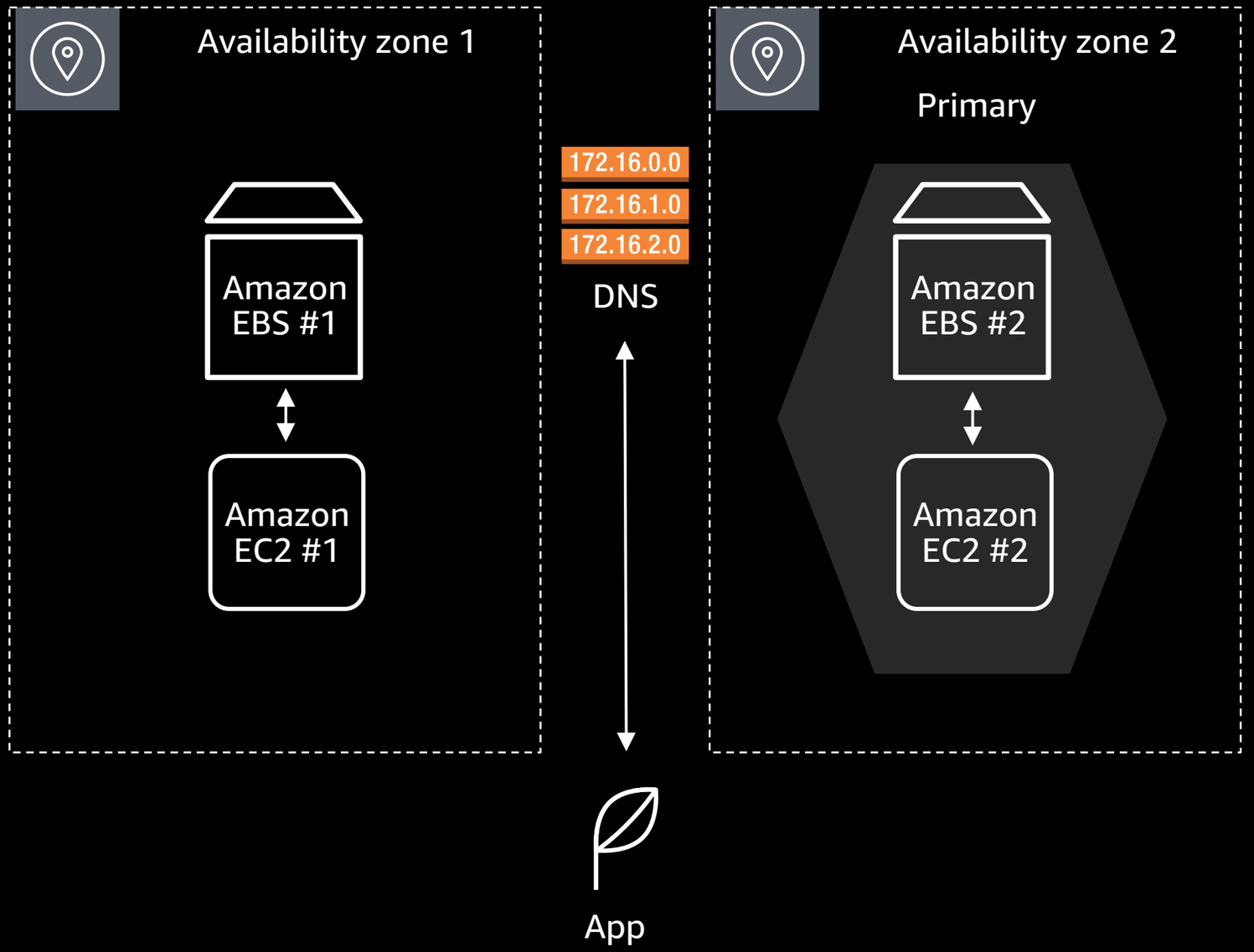
Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API

Redirection to the new primary instance is provided through DNS



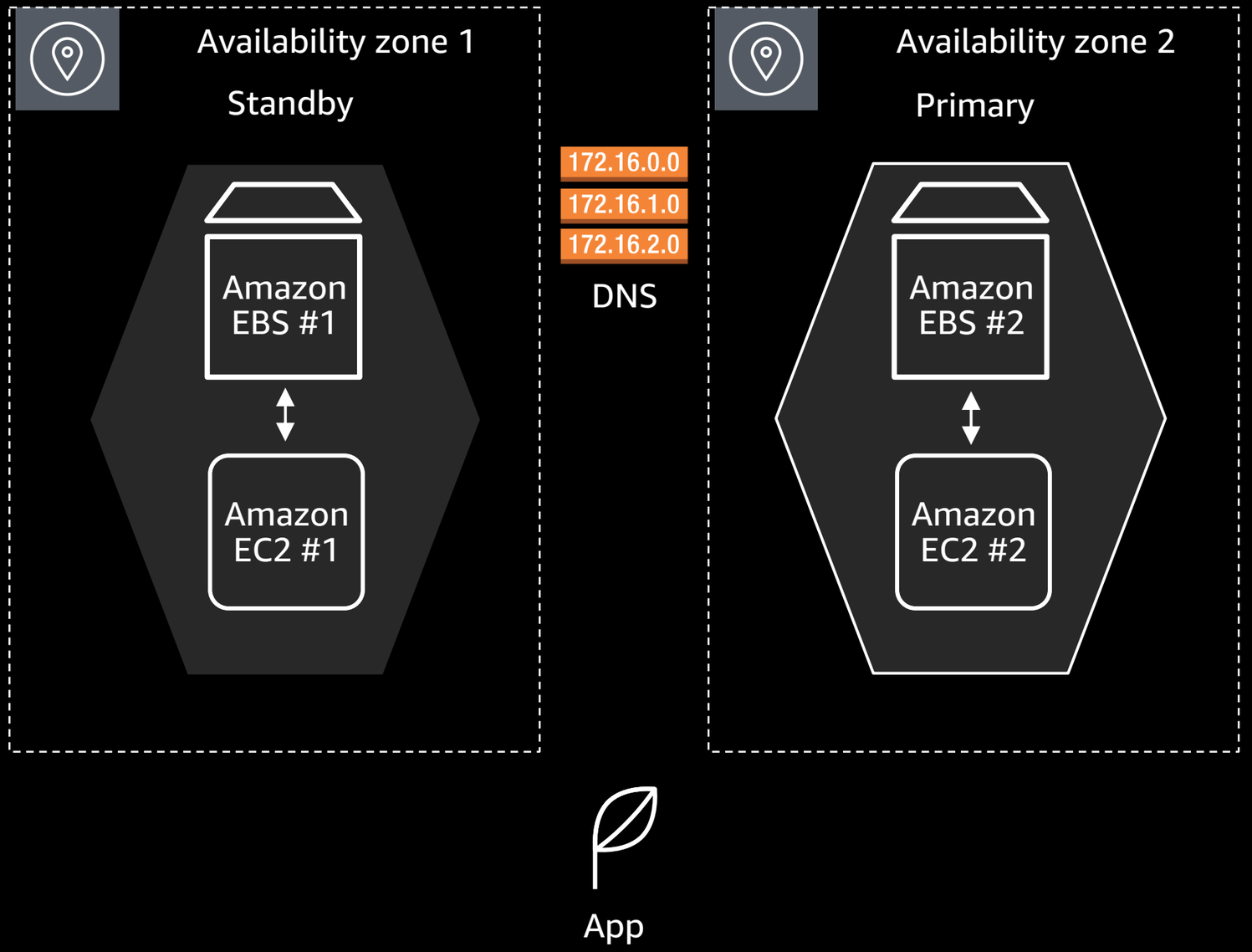
Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API

Redirection to the new primary instance is provided through DNS



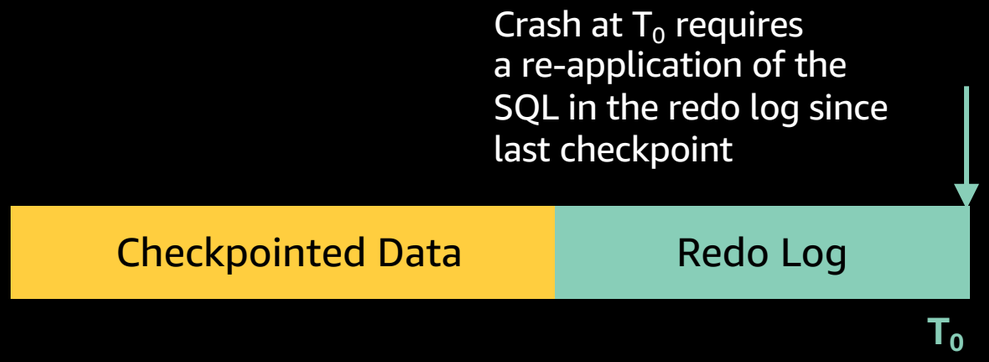
Aurora instant crash recovery

Traditional database

Have to replay logs since the last checkpoint

Typically 5 minutes between checkpoints

Single-threaded in MySQL; requires a large number of disk accesses

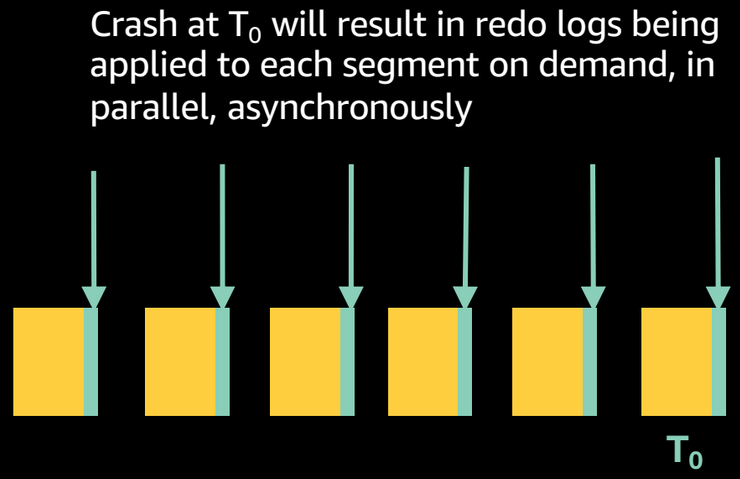


Amazon Aurora

Underlying storage replays redo records on demand as part of a disk read

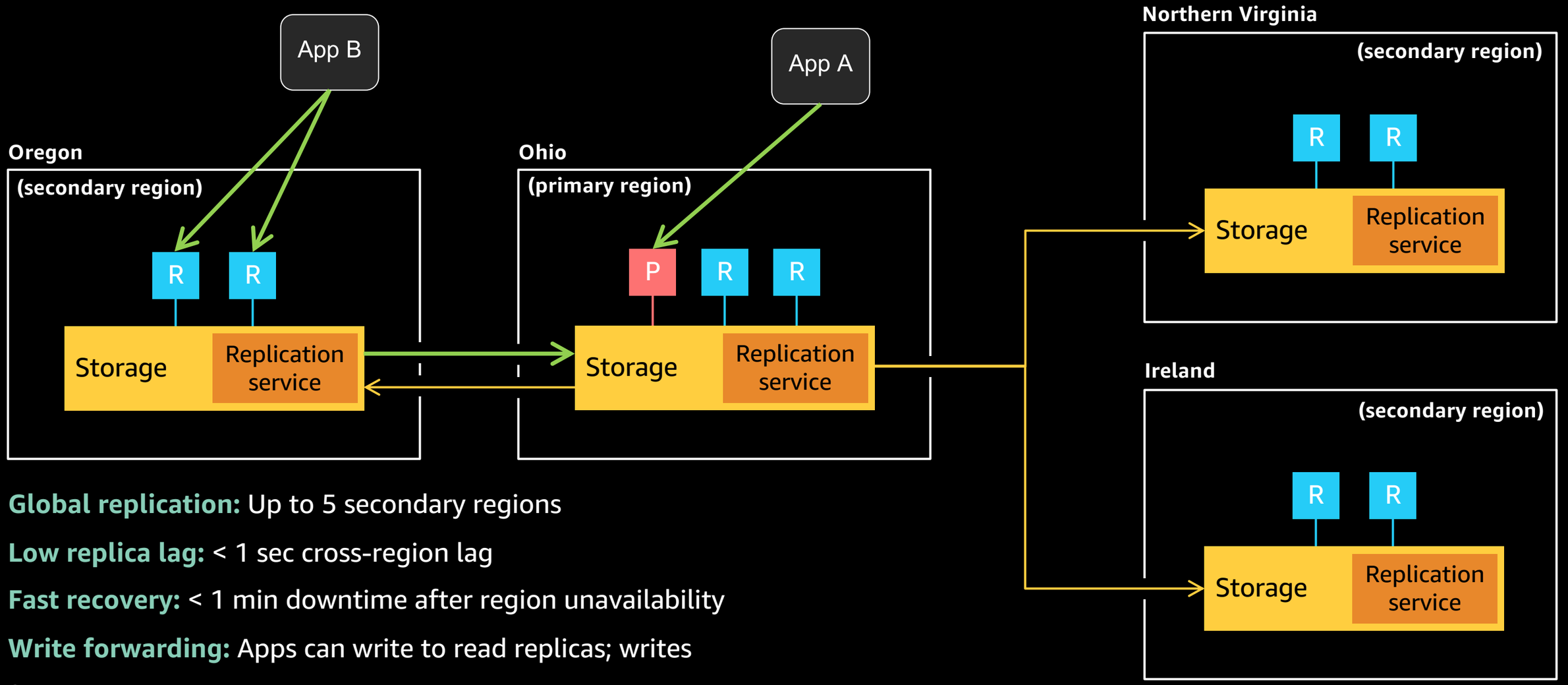
Parallel, distributed, asynchronous

No replay for startup



Aurora Global Database

FASTER DISASTER RECOVERY AND ENHANCED DATA LOCALITY



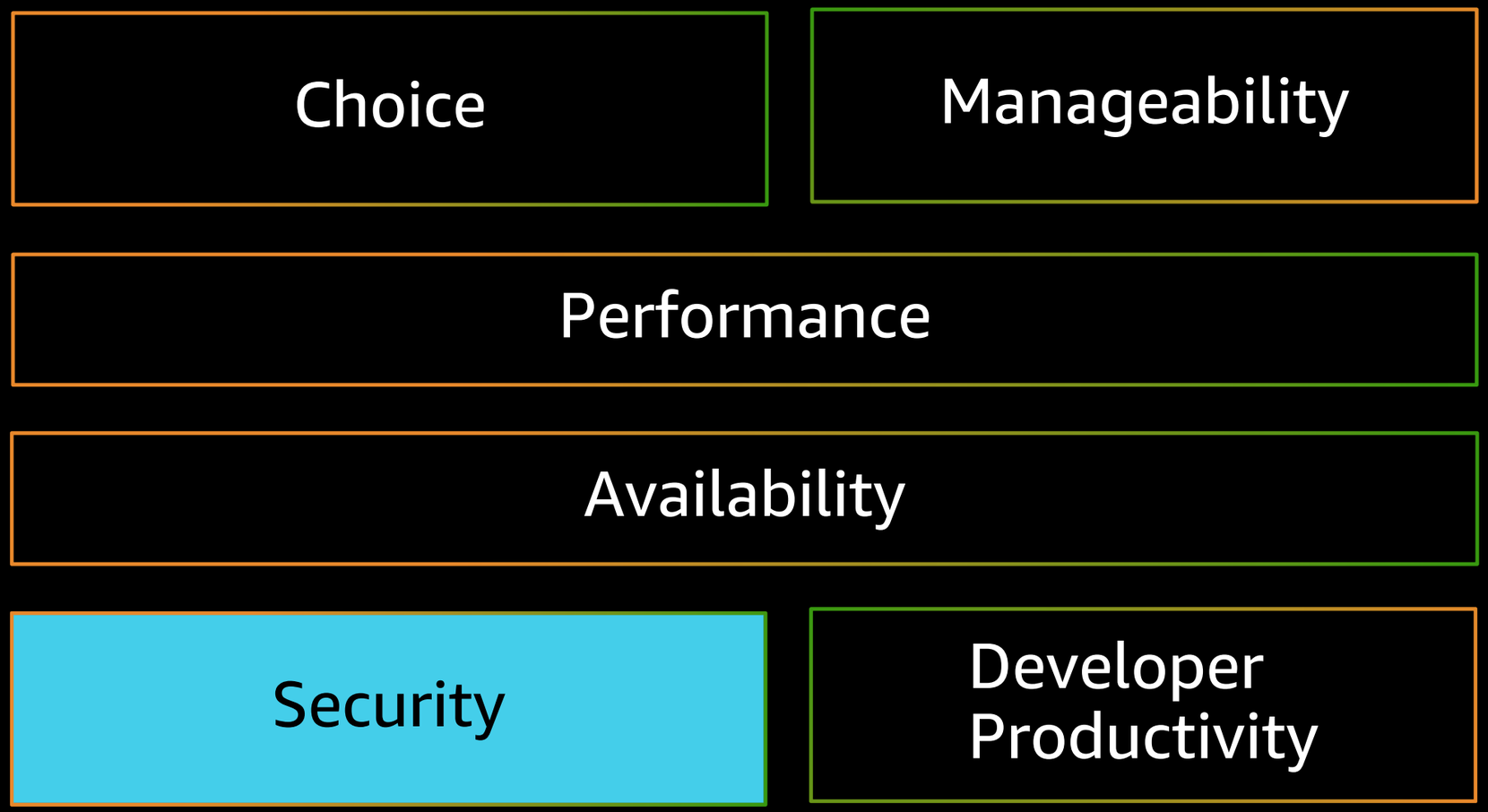
Global replication: Up to 5 secondary regions

Low replica lag: < 1 sec cross-region lag

Fast recovery: < 1 min downtime after region unavailability

Write forwarding: Apps can write to read replicas; writes forwarded to primary region and sent to all regions

What do you want from your database service?



How do I secure my database?



Amazon Virtual Private Cloud (Amazon VPC)
Network Isolation



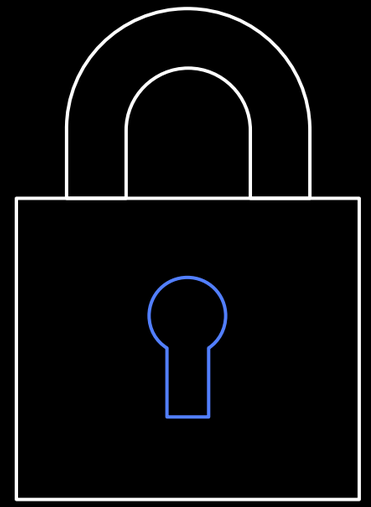
AWS Identity and Access Management (IAM)
Resource-level permission controls



AWS Key Management Service (AWS KMS)
Encryption at rest



SSL
Protection for data in transit

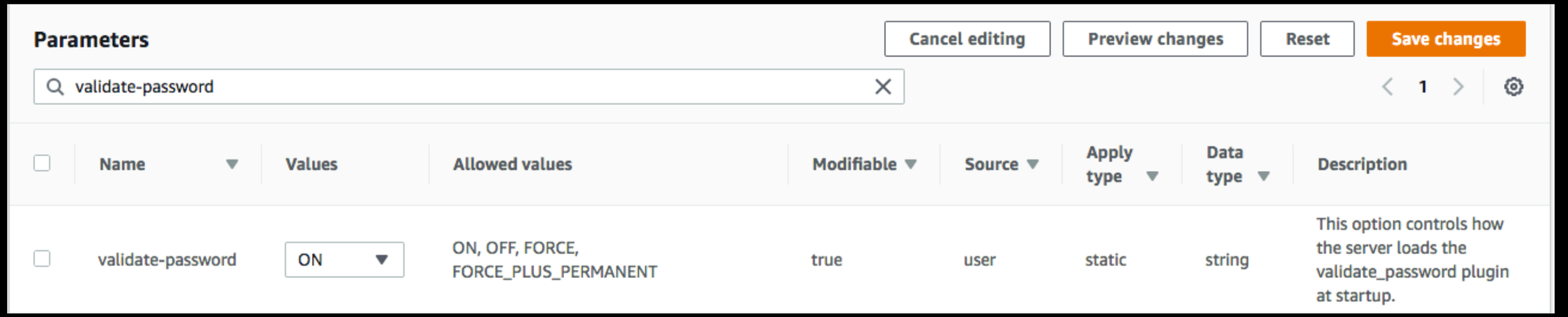


Password validation

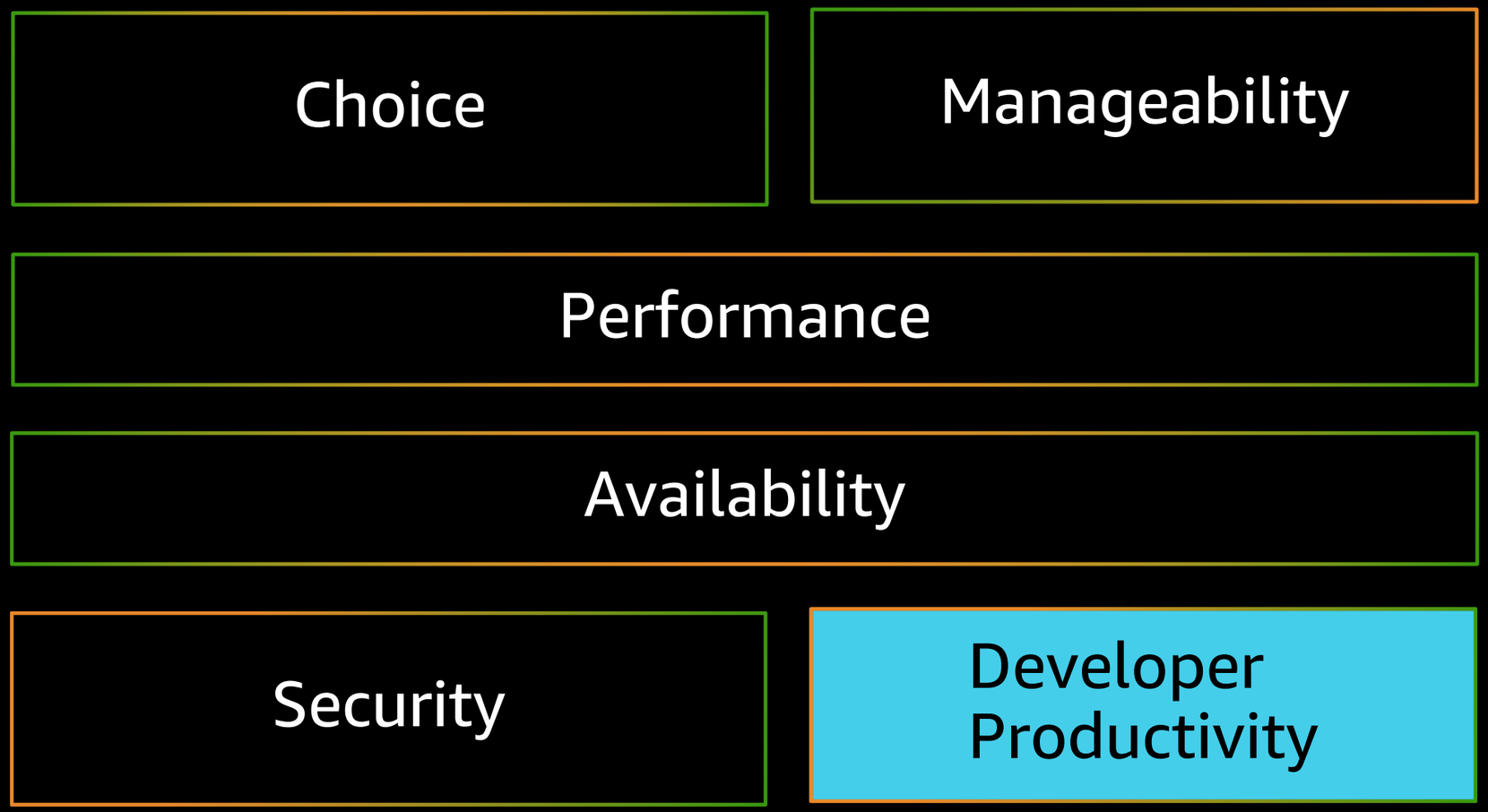
- Supports Amazon RDS for MySQL version 5.6, 5.7, and 8.0
- The MySQL DB instance performs password validation
- Install plugin

```
mysql> INSTALL PLUGIN validate_password SONAME 'validate_password.so';
Query OK, 0 rows affected (0.01 sec)
```

- Configure the Parameter group and reboot

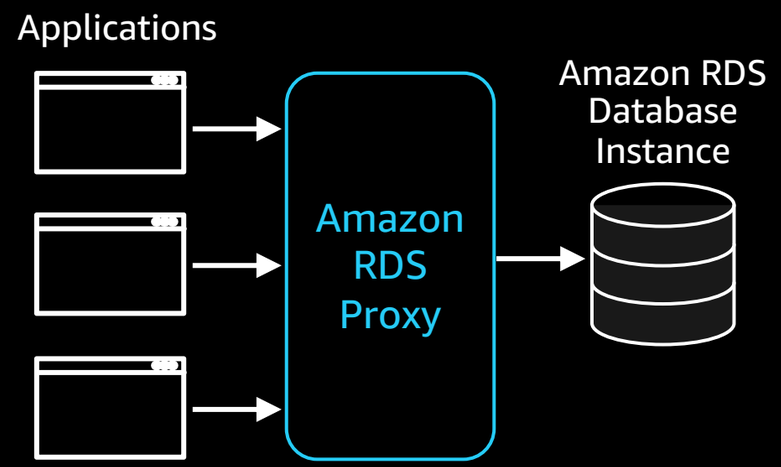


What do you want from your database service?



Amazon RDS Proxy

FULLY-MANAGED, HIGHLY-AVAILABLE DATABASE PROXY FOR AMAZON RDS



Connection Pooling

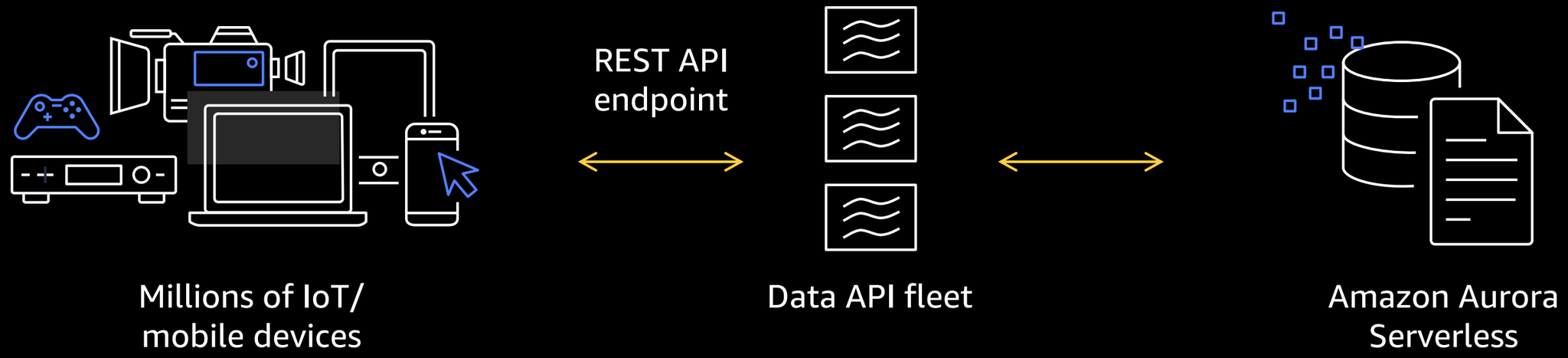
Supports a large number of application connections

Deployed across multiple AZs and fails over without losing a connection

Integrates with AWS Secrets Manager and AWS IAM

Get started with a few clicks in the console

Amazon RDS Data API for HTTP access



Serverless apps often have restrictions

Limited network connectivity to the DB

No persistent connection to the DB

Small client (e.g., IoT) with limited resources

Amazon RDS Data API provides –

Public endpoint accessible via HTTP

Access without any client configuration

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.