AWS Relational Database Service (RDS), Aurora & Global Database

A comprehensive overview of AWS's managed database solutions

Session Objectives

Understand AWS RDS fundamentals

Core concepts, architecture, and benefits of AWS's managed relational database service

Explore Amazon Aurora architecture and features

AWS's high-performance database with MySQL and PostgreSQL compatibility

Learn about Aurora Global Database

Multi-region deployment for global applications and disaster recovery

Hands-on concepts for creating and managing RDS/Aurora

Practical knowledge for implementation and operation

What is AWS RDS?

AWS RDS (Relational Database Service) is a **fully managed relational database service** that handles routine database tasks, allowing you to focus on your applications.

It supports multiple database engines:

- MySQL
- PostgreSQL
- MariaDB
- SQL Server
- Oracle

Key Benefits

Automated backups and patching

Simplified scaling operations

High availability configurations

RDS provides managed database services, unlike self-managed databases on EC2 instances which require manual administration.

RDS Architecture

Control Plane

- Manages database configuration
- Handles administrative operations
- Orchestrates backups and scaling

Data Plane

- Processes database queries
- Manages data storage and retrieval
- Handles replication between instances

RDS Features

Automated Backups

Daily snapshots and transaction logs for point-in-time recovery

Point-in-Time Restore

Recover your database to any second during the retention period

Multi-AZ Deployment

Synchronous replication to a standby instance in a different Availability Zone

Read Replicas

Scale read capacity and reduce load on primary instance

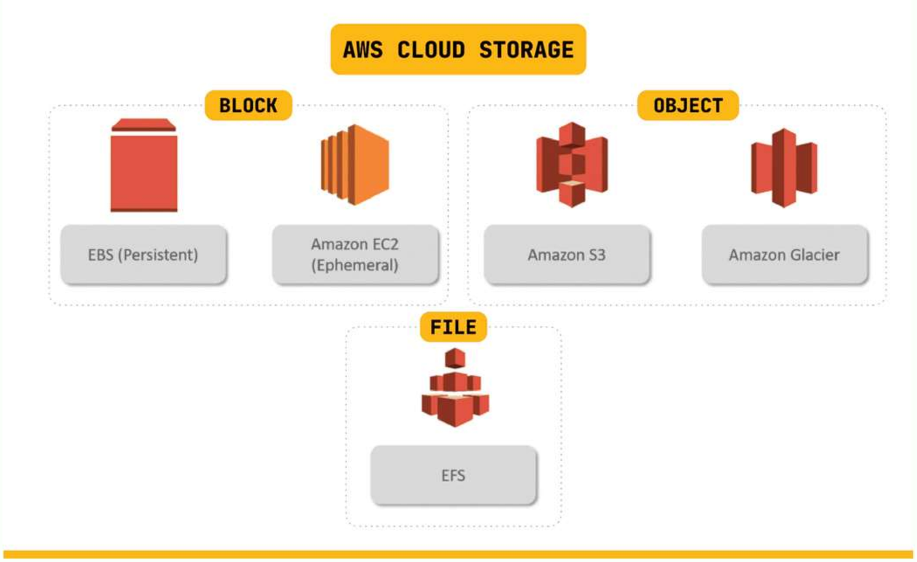
Security

IAM authentication, KMS encryption, VPC isolation

Monitoring

CloudWatch metrics and Performance Insights for database analysis

Storage Types



General Purpose SSD (gp2/gp3)

Cost-effective storage for most database workloads

3,000-16,000 IOPS baseline performance

Provisioned IOPS SSD (io1/io2)

High-performance storage for I/O-intensive workloads

Up to 256,000 IOPS for demanding applications

Magnetic (deprecated)

Legacy storage option not recommended for new instances

Lower performance compared to SSD options

Choose storage type based on your workload's performance requirements, budget constraints, and consistency needs.

Security in RDS

Encryption at Rest

AWS Key Management Service (KMS) integration for database, backups, and snapshots

Encryption in Transit

SSL/TLS connections between applications and database instances

Network Security

VPC isolation, security groups, and network ACLs to control access

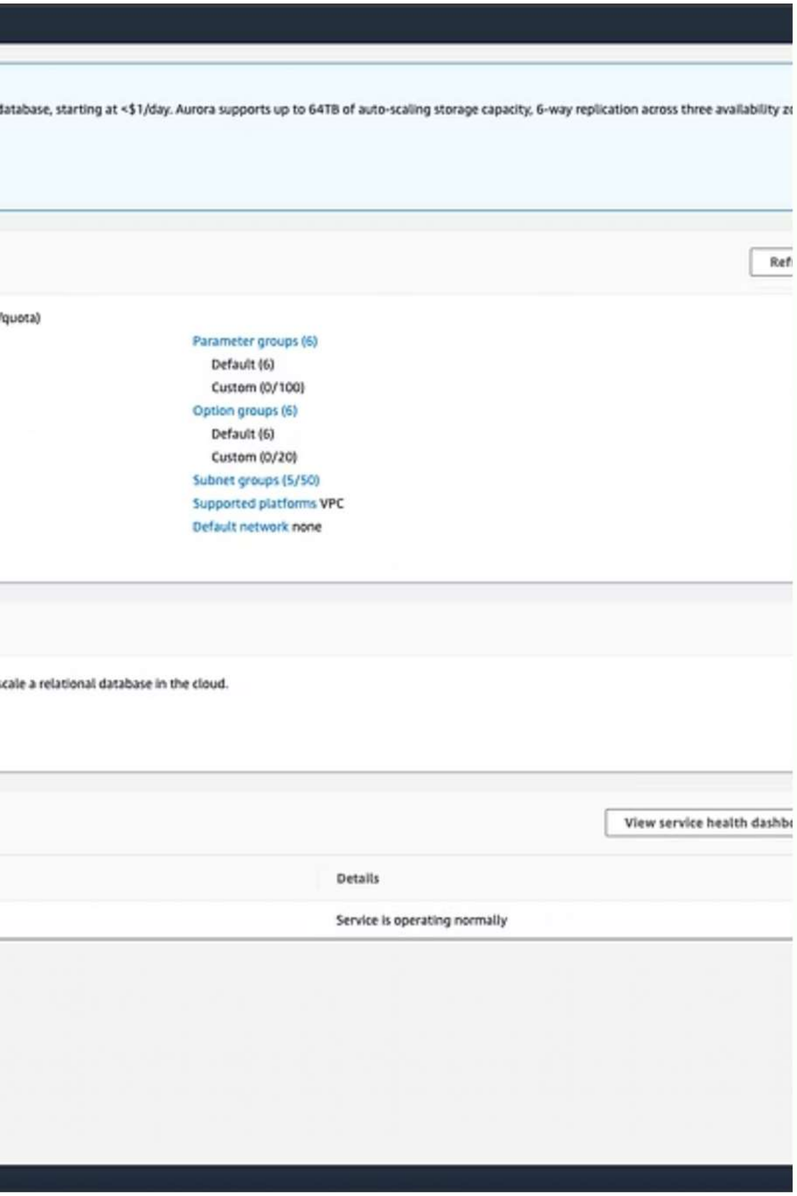
Authentication

IAM database authentication for MySQL and PostgreSQL

Parameter Groups

Configure database settings to enforce security policies

Demo: Creating an RDS Instance



database, starting at <$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability z

/quota)

Parameter groups (6)
Default (6)
Custom (0/100)
Option groups (6)
Default (6)
Custom (0/20)
Subnet groups (5/50)
Supported platforms VPC
Default network none

scale a relational database in the cloud.

Details

Service is operating normally

Select Engine

Choose MySQL from the database engine options

Configure Settings

Set instance specifications, storage, and connectivity options

Enable Security

Configure encryption, backup, and monitoring features

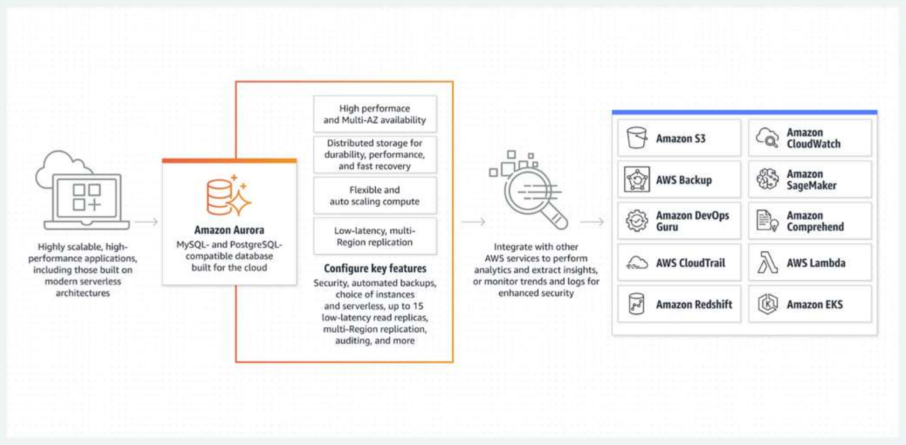
Launch Instance

Review settings and create the database

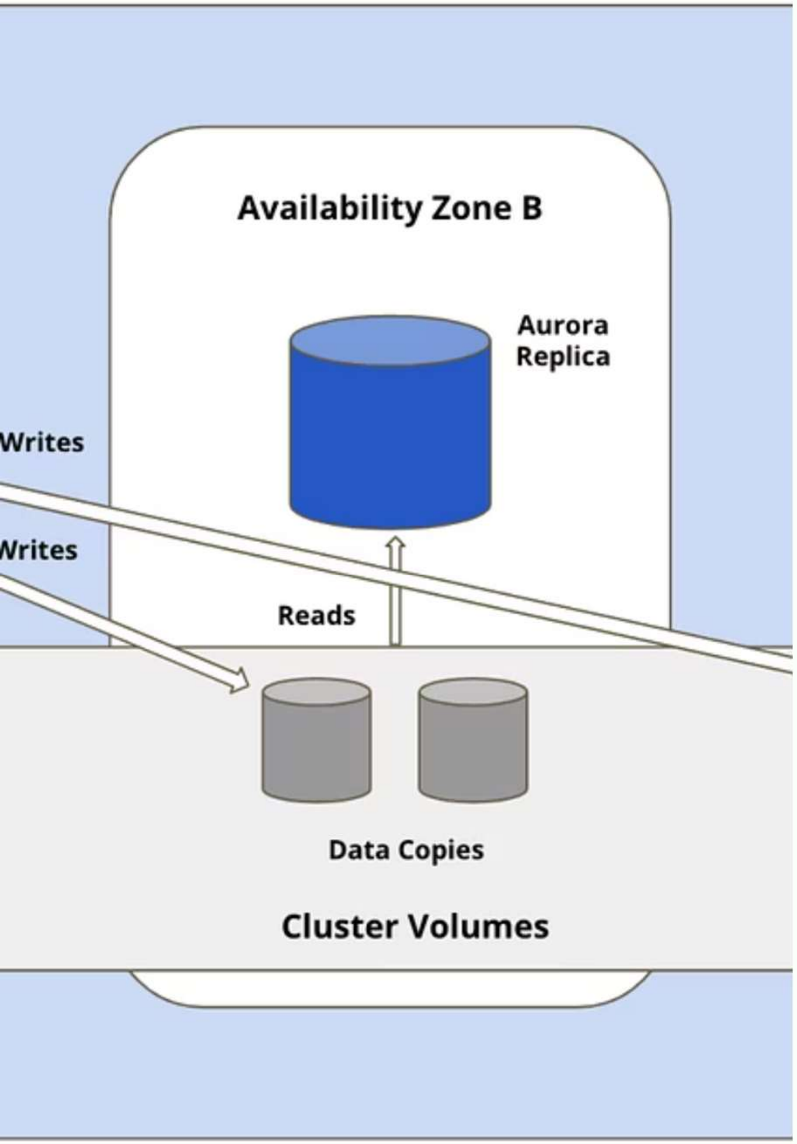
Introduction to Aurora

Amazon Aurora is AWS's proprietary database engine that provides MySQL and PostgreSQL compatibility with significantly improved performance:

- **5x faster** than standard MySQL
- **3x faster** than standard PostgreSQL
- Fully managed through RDS service
- Storage auto-scaling up to 128 TiB


Highly scalable, high-performance applications, including those built on modern serverless architectures
Amazon Aurora
MySQL- and PostgreSQL-compatible database built for the cloud
High performace and Multi-AZ availability
Distributed storage for durability, performance, and fast recovery
Flexible and auto scaling compute
Low-latency, multi-Region replication
Configure key features
Security, automated backups, choice of instances and serverless, up to 15 low-latency read replicas, multi-Region replication, auditing, and more
Integrate with other AWS services to perform analytics and extract insights, or monitor trends and logs for enhanced security
Amazon S3
Amazon CloudWatch
AWS Backup
Amazon SageMaker
Amazon DevOps Guru
Amazon Comprehend
AWS CloudTrail
AWS Lambda
Amazon Redshift
Amazon EKS

ⓘ Aurora maintains compatibility with open-source engines while delivering cloud-native performance improvements and features.



Aurora Architecture

Shared Storage Cluster

- Data distributed across 3 Availability Zones
- Complete separation of compute and storage
- Storage automatically grows in 10GB increments

Replication Model

- 6-way replication across 3 AZs
- Only need 4/6 copies for writes
- Only need 3/6 copies for reads
- Self-healing storage with continuous verification

Aurora provides dedicated **Writer and Reader endpoints** to automatically direct traffic to appropriate instances.

Aurora vs RDS

Feature	RDS MySQL/PostgreSQL	Aurora
Storage Scaling	Manual	Automatic
Replication	5 Read Replicas	15 Read Replicas
Latency	Higher	Lower
Failover Time	Minutes	<30 seconds

Aurora's architecture provides significant advantages for high-performance, mission-critical workloads that require minimal downtime.

Aurora Features

Aurora Serverless

On-demand, auto-scaling configuration for variable workloads

Pay only for resources consumed by database

Backtrack

Rewind database to a specific point in time without restoring from backup

Recover quickly from user errors like dropped tables

Parallel Query

Pushes query processing to storage layer for improved performance

Ideal for analytical queries on large datasets

Multi-Master

Multiple instances that can perform write operations (Aurora MySQL only)

Provides higher availability for write operations

Global Database

Cross-region replication with minimal latency

Global read scaling and disaster recovery

Demo: Creating an Aurora Cluster

Select Aurora Engine

Choose Aurora with MySQL or PostgreSQL compatibility

Configure Cluster

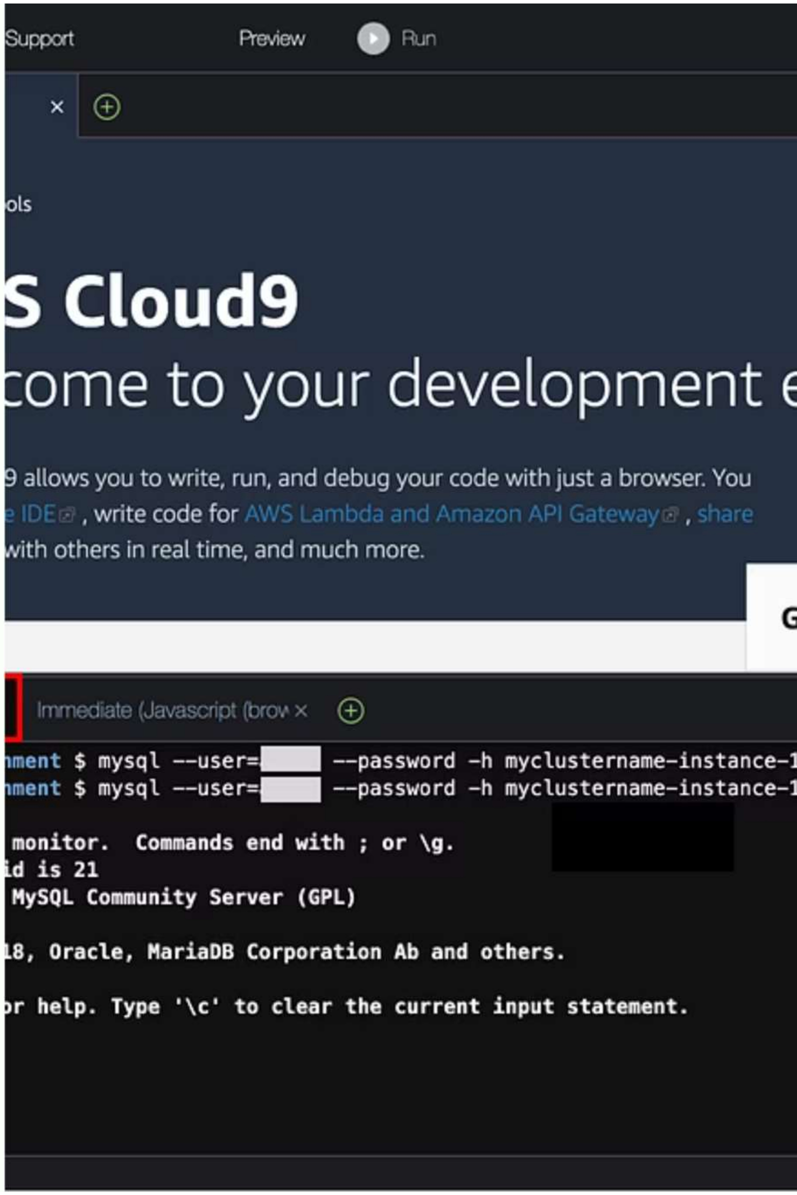
Set instance class, availability, and capacity settings

Network & Security

Configure VPC, subnet group, and security settings

Create Cluster

Review configuration and launch the Aurora cluster



S Cloud9

come to your development e

9 allows you to write, run, and debug your code with just a browser. You e IDE , write code for AWS Lambda and Amazon API Gateway , share with others in real time, and much more.

```
nment $ mysql --user=[redacted] --password -h myclustername-instance-1
nment $ mysql --user=[redacted] --password -h myclustername-instance-1

 monitor.  Commands end with ; or \g.          [redacted]
id is 21
 MySQL Community Server (GPL)

18, Oracle, MariaDB Corporation Ab and others.

or help. Type '\c' to clear the current input statement.
```

What is Aurora Global Database?

Aurora Global Database extends Aurora's capabilities across multiple AWS regions, providing:

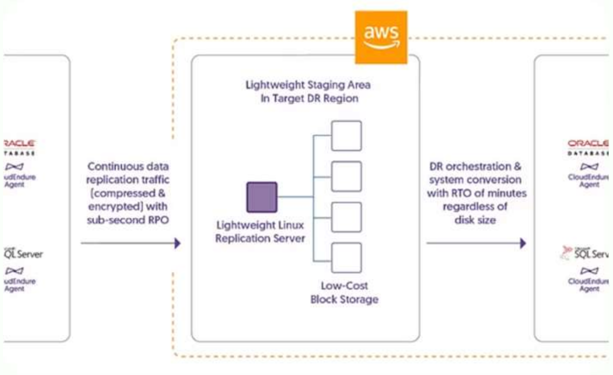
- Primary region for **write operations**
- Up to 5 secondary regions for **read operations**
- Replication lag **less than 1 second**
- Failover to another region in **minutes**

Global Database uses dedicated infrastructure for replication, not relying on traditional database binlog mechanisms.

Use Cases for Aurora Global Database

Global Applications

Provide low-latency reads to users worldwide by placing read replicas in regions closest to users

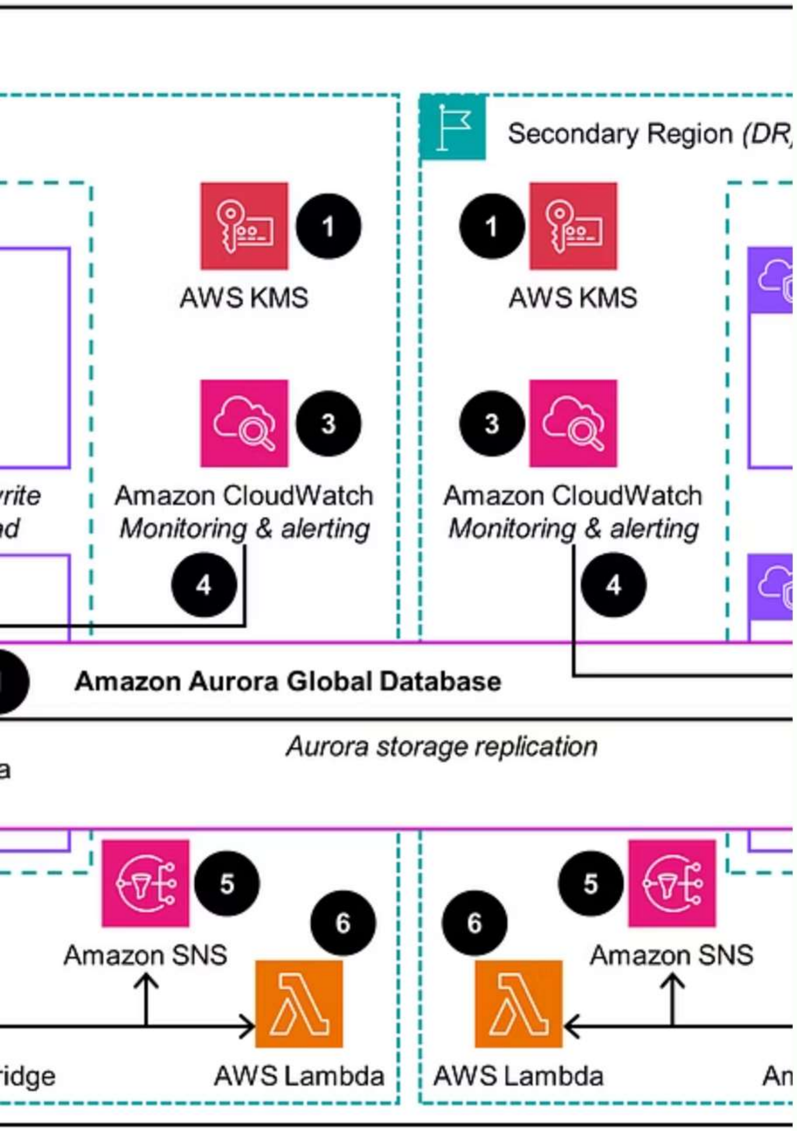

aws
Lightweight Staging Area
In Target DR Region
Continuous data replication traffic (compressed & encrypted) with sub-second RPO
Lightweight Linux Replication Server
Low-Cost Block Storage
DR orchestration & system conversion with RTO of minutes regardless of disk size
ORACLE DATABASE
CloudEndure Agent
SQL Server
CloudEndure Agent

Disaster Recovery

Maintain a complete copy of your database in another region for business continuity during regional outages

Multi-Region Analytics

Run reporting and analytical queries against local read replicas without impacting production workloads



Architecture of Aurora Global DB

Primary Region

- Contains writer instance
- Can include multiple reader instances
- Handles all write operations

Secondary Regions

- Contain read-only instances
- Typically 1-16 Aurora replicas per region
- Receive updates via dedicated replication infrastructure

Data changes are replicated from the primary to secondary regions with typical latency under 1 second, using Aurora's storage-based replication rather than traditional binlog mechanisms.

Creating Aurora Global DB

ⓘ You can create a Global Database from an existing Aurora cluster or create a new one with global capabilities from the beginning.

Key settings to consider include primary region selection based on where most write operations occur, secondary region selection based on user proximity and disaster recovery requirements, and regular failover testing to ensure recovery procedures work as expected.

Pricing Considerations

2x

Instance Costs

Pay for Aurora instances in each region

$0.02

Replication Cost

Per GB of data transferred between regions

1x

Storage

Pay for storage in each region

Cost Optimization Tips

- Use smaller instance sizes in secondary regions if read load is lower
- Consider Aurora Serverless for variable workloads
- Monitor cross-region data transfer costs
- Balance availability needs with cost constraints

The additional cost of Global Database must be weighed against the business value of global availability and disaster recovery capabilities.

Demo: Creating a Global Database

Start with Aurora Cluster

Begin with an existing Aurora cluster or create a new one

Add Secondary Region

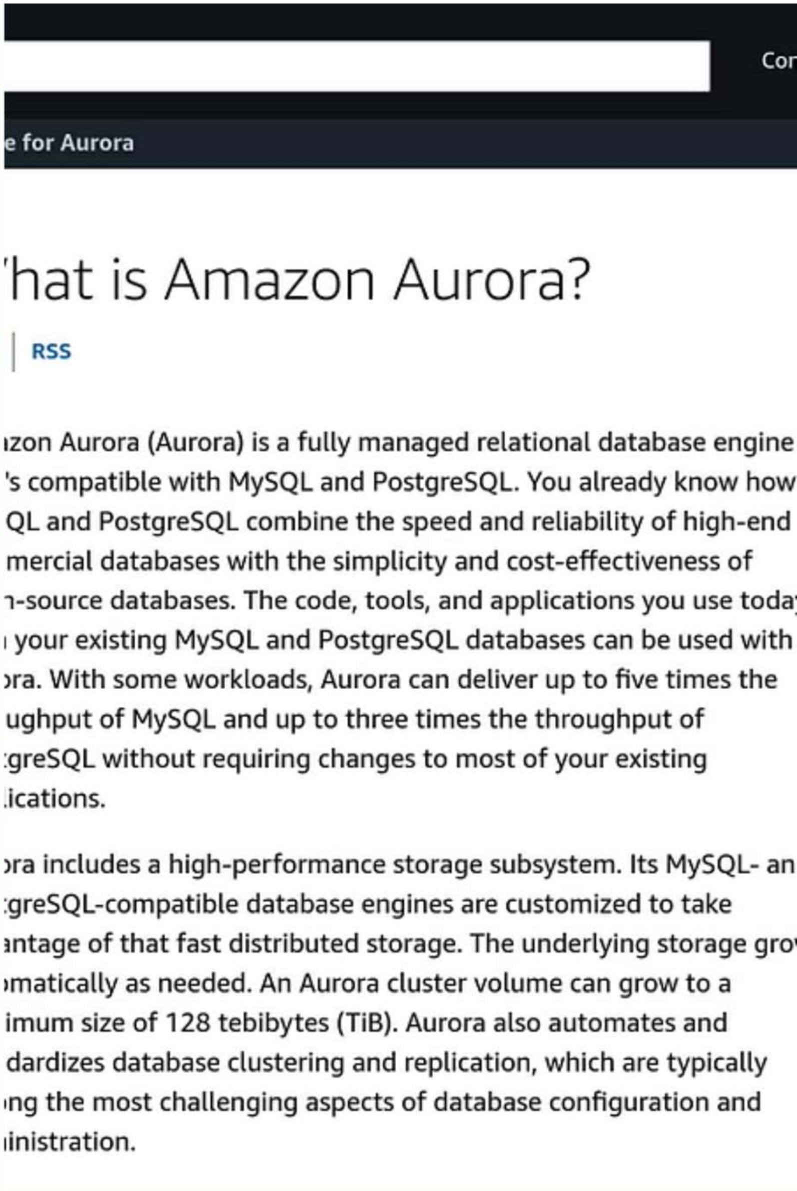
Select "Add region" from the cluster actions menu

Configure Settings

Specify instance type, count, and other parameters

Verify Replication

Monitor replication status and test read operations



'hat is Amazon Aurora?

ızon Aurora (Aurora) is a fully managed relational database engine
's compatible with MySQL and PostgreSQL. You already know how
QL and PostgreSQL combine the speed and reliability of high-end
mercial databases with the simplicity and cost-effectiveness of
ı-source databases. The code, tools, and applications you use toda
ı your existing MySQL and PostgreSQL databases can be used with
ora. With some workloads, Aurora can deliver up to five times the
ughput of MySQL and up to three times the throughput of
:greSQL without requiring changes to most of your existing
.ications.

ora includes a high-performance storage subsystem. Its MySQL- an
:greSQL-compatible database engines are customized to take
antage of that fast distributed storage. The underlying storage gro
omatically as needed. An Aurora cluster volume can grow to a
imum size of 128 tebibytes (TiB). Aurora also automates and
dardizes database clustering and replication, which are typically
ıng the most challenging aspects of database configuration and
ıinistration.

Best Practices

1

Instance Selection

Choose the right instance type based on workload characteristics:

- Memory-optimized for caching-intensive workloads
- Compute-optimized for CPU-bound operations

2

High Availability

Enable Multi-AZ deployment for all production databases to ensure business continuity during infrastructure failures

3

Performance Optimization

Use Performance Insights to identify bottlenecks and tune database parameters accordingly

4

Security

Enforce encryption at rest using KMS and implement proper IAM policies for access control

5

Monitoring

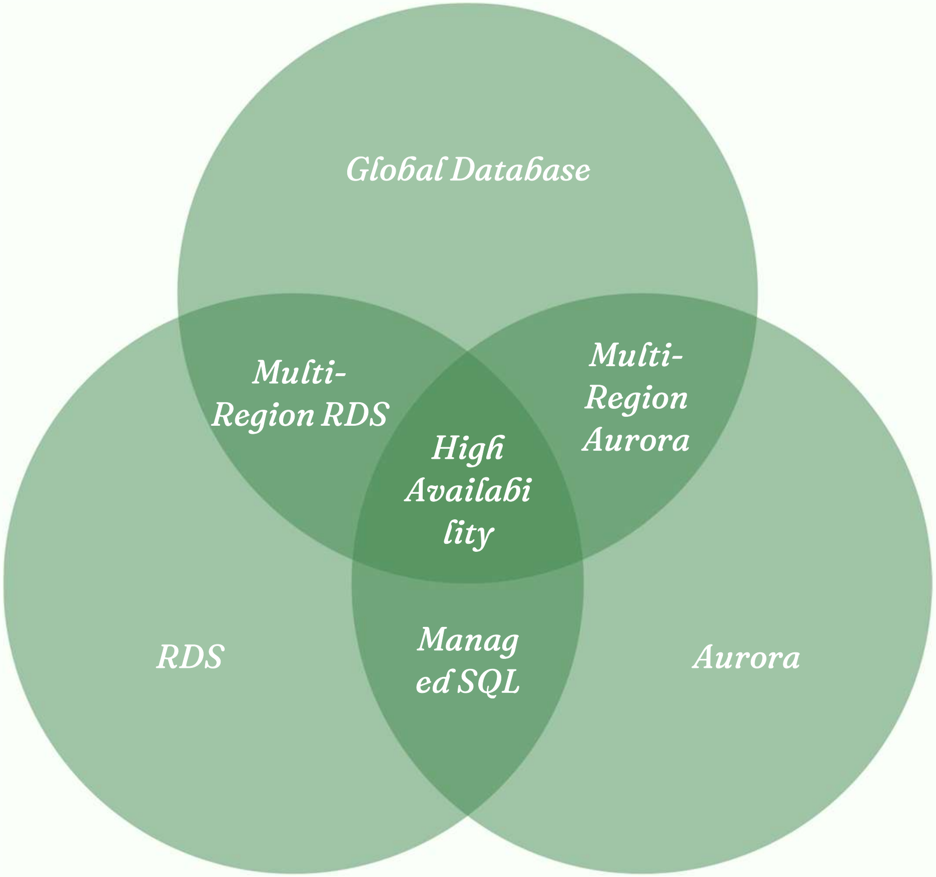
Implement comprehensive CloudWatch alarms for key metrics and set up automated responses

6

Disaster Recovery

Test failover procedures regularly to ensure they work as expected when needed

Summary



Key Takeaways

- RDS provides managed relational database services with multiple engine options
- Aurora delivers high performance with MySQL/PostgreSQL compatibility
- Global Database extends Aurora across regions for global applications and disaster recovery

Next Steps

- Explore hands-on labs to gain practical experience
- Review AWS documentation for detailed configuration options
- Consider migration paths for existing databases

Questions & Answers

Thank you for your attention!

Please feel free to ask any questions about AWS RDS, Aurora, or Global Database.

ⓘ For additional resources, visit the AWS Database Blog and documentation at aws.amazon.com/blogs/database/