



Amazon Aurora cost optimization best practices

Aditya Samant, Krishna Sarabu

Sr. Database Specialist Solutions Architects

Agenda

- Aurora cost model
- Aurora cost components
- How to monitor and optimize Aurora I/O cost
- Tools for cost observability and optimization
- Demo!

Amazon Aurora

COMMERCIAL-GRADE CLOUD NATIVE DATABASE

Delivered as a **managed** service



Drop-in compatibility with MySQL and PostgreSQL

Simplicity and cost-effectiveness of open-source databases

Throughput and availability of commercial databases

Simple pay-as-you-go pricing



Aurora Cost Model (pay-as-you-go)



Use what you need, pay for what you use

- **Traditional cost model:**

- Provision for peak usage, pay for peak usage (regardless of actual use).
- Example: Provisioned 100GB in anticipation of db growth, but current utilization maybe only 10GB.

- **Aurora cost model:**

- Aurora gives you the flexibility to provision exactly what you need.
- You only pay for what you use.
- Storage, read replicas and compute* can scale automatically.
- Aurora decouples storage and compute. These components are charged individually.

* Only applies to serverless.



Aurora Cost Components



Primary cost components

Compute

- Instance type and size (provisioned)
 - (e.g. db.r6g.large)
- Aurora Capacity Units (Serverless)

Storage

- Storage volume size
- IOPS consumed (read and write)

Additional cost components

Backup

- Automated backups
- Manual snapshots

Data Transfer

- Network utilization for data coming IN and going OUT.

Additional cost components

Advance Aurora features

- Features follow the same pay-as-you-use cost model
- Charged for resources consumed
- Global Database, Backtrack, Machine Learning integration, Parallel Query, Snapshot export S3, Database Activity Streams

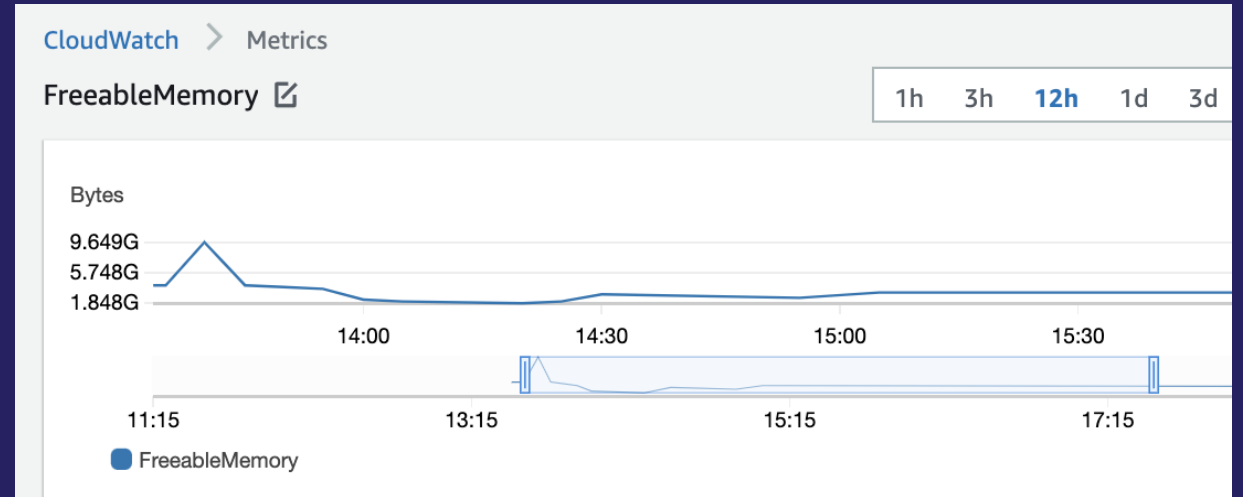
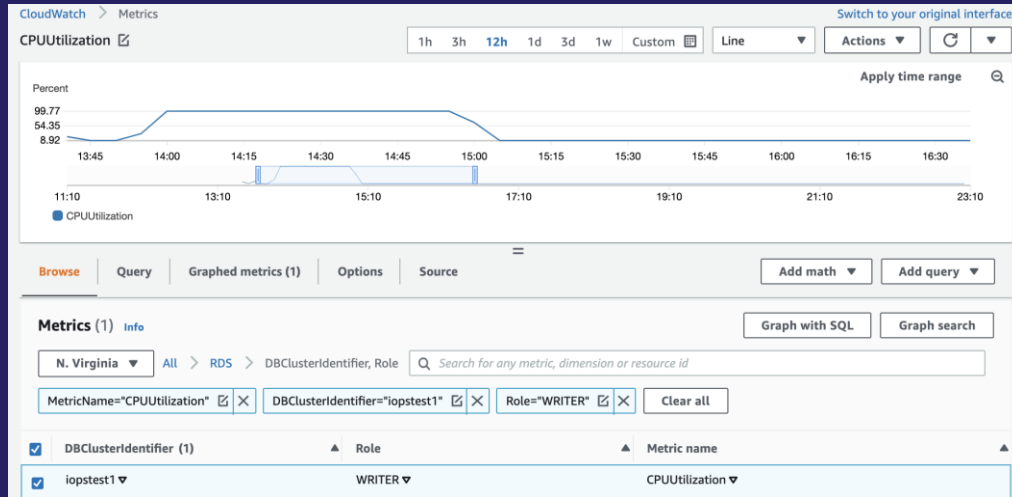
Compute cost and best practices (Provisioned)

- Simple hourly pricing.
- Use CloudWatch (CW) to understand the utilization. Example (CPU Utilization, Freeable memory, etc.)
- AWS Trusted advisor can be used to find underutilized or idle instances.
- Best Practices for optimization:
 - Consider using Reserved Instances, Graviton instance types.
 - CW and Trusted advisor data points to scale down.
 - Utilization is driven by resource intensive queries. Tuning queries, reducing frequency will help significantly.
 - Consider using read replicas, utilize auto scaling feature for workload tailored scaling.
 - Scheduled stop/start of Aurora instances using Lambda.
 - Compute cost can be reduced by using headless clusters (Global Database, Clones)

Compute cost and best practices (Serverless)

- On-demand, auto-scaling configuration that automatically adjusts database capacity based on application needs.
- Scales compute capacity on-demand (up and down) in ACU (Aurora Capacity Units)
- Current iteration - Serverless V1, with Serverless V2 on the horizon.
- Monitor CloudWatch metric : **Serverless Database Capacity (Count)**
- Best practices for optimization:
 - Carefully select the minimum and maximum ACU.
 - Utilize Pause and Resume feature if application can tolerate warm-up latency.

CloudWatch Metrics – Compute



Compute cost examples (Provisioned)

Provisioned Example:

Instance type : db.r6g.xlarge

Instance active : 10 days (240 hours)*

Price per hour (US-EAST) : \$0.519

You will be charged: $240 \times \$0.519 = \124.56

* Assuming instance was shutdown after 10 days

Compute cost examples (Serverless)

Serverless Example (V1):

ACUs consumed : 1 ACUs X 50 hours + 2 ACU X 100 hours

ACU price per hour (US-EAST) : \$0.6

You will be charged: $1 \times 50 \times \$0.6 + 2 \times 50 \times \$0.6 = \$90$

Storage cost

- Storage consumption is billed in per GB-month increments.
- Only pay for what you consume and no need for advance provisioning.
- Storage is billed at \$0.10 per GB-Month (US-EAST).

Example:

Initial Database Size = 100 GB

Days in this month = 30

Daily Growth = 20 GB

You will be charged: $100/30 \times \$0.10 + 120/30 \times \$0.10 \dots + 680/30 \times \$0.10 = \$39$



Day 1



Day 2



Day 30

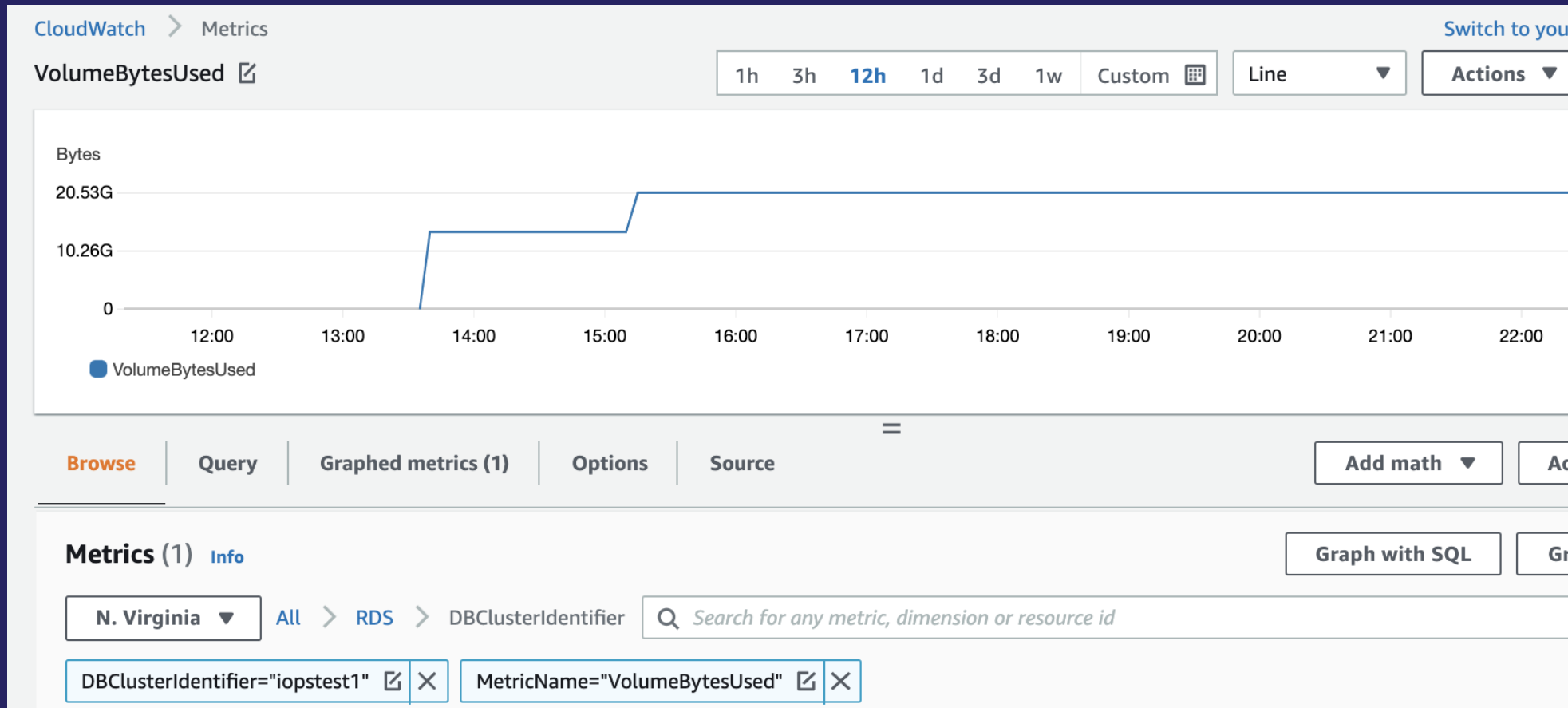
Storage cost optimization best practices

- Monitor usage using **Volume Bytes Used (GiB)** CloudWatch metric.
- Take advantage of Aurora's dynamic volume sizing.*
- Delete unused objects (tables, indexes).
- Tune/optimize Vacuum process.

* MySQL versions 1.23 and 2.09, 3.01 and above and Aurora PostgreSQL versions 3.30 and 2.6 and above



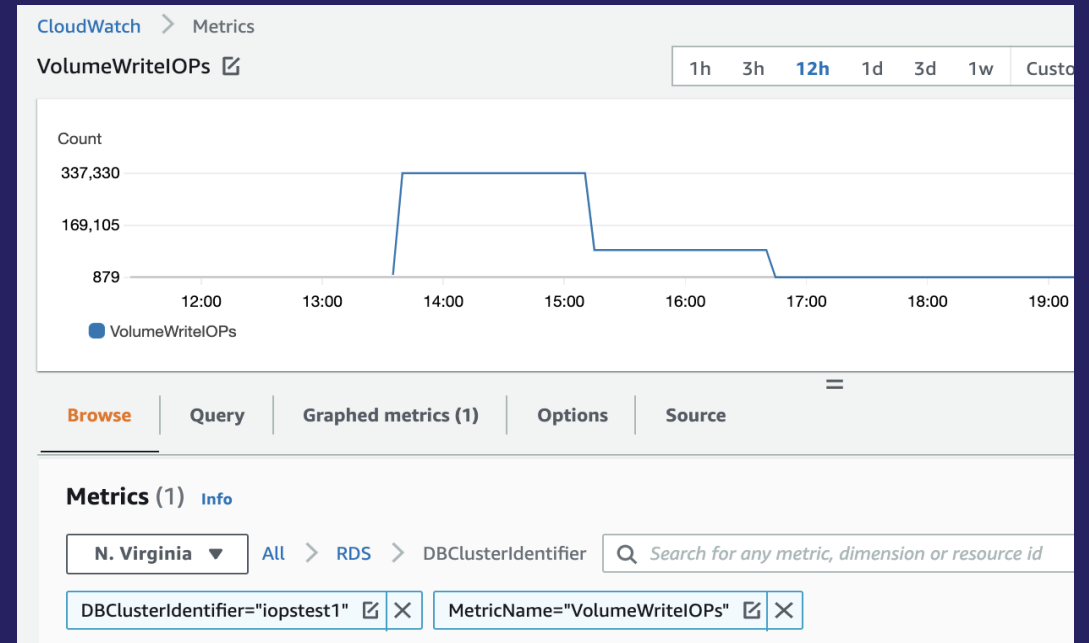
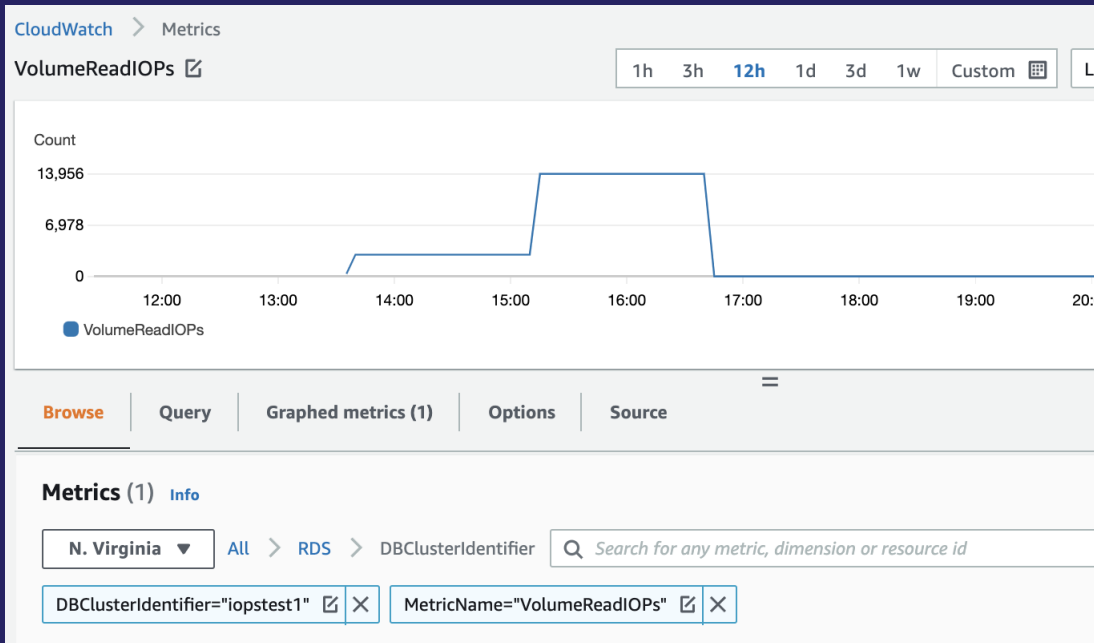
CloudWatch Metrics –Storage



IO cost components

- IO billed separately from storage.
- Billed for Read IOPS and Write IOPS. Only charged for the IOPS you consume, no pre-provisioning.
- IO is charged per one million requests. (example \$0.20 for one million requests in US-EAST).
- Read and Write requests are calculated using physical page access.
- Logical reads (buffer cache hits) are not billed.
- CloudWatch Metrics to monitor:
 - For Read IOPS : **Volume Read IOPS (Count)**
 - For Write IOPS : **Volume Write IOPS (Count)**

CloudWatch Metrics – IOPS



Examples of IOPS billing

- Example Read IOPS:

Example:

Pages Read per second: 100

Total pages read per month: $100 \times 730 \times 60 \times 60 = 262.8$ million reads

$\underbrace{\hspace{1.5em}}_{\text{Pages per second}} \times \underbrace{\hspace{1.5em}}_{\text{Hours}} \times \underbrace{\hspace{1.5em}}_{\text{Minutes}} \times \underbrace{\hspace{1.5em}}_{\text{Seconds}}$

Total billed end of month: $262.8 \times \$0.20 = \52

Examples of IOPS billing

- Example Write IOPS:

Example:

Pages Modified per second: 10*

Total pages written per month: $10 \times 730 \times 60 \times 60 = 26.2$ million writes

Pages per second Hours Minutes Seconds

Total Writes billed end of month: $26.2 \times \$0.20 = \5.24

Total IOPS billed (READ+WRITE) = \$57.24

* upto 4KB changes are box-cared as a single IO operation

IO cost optimization best practices:

- Best practices to optimize Read IO cost:
 - Tune Read IO intensive queries. For example - avoid Full scans, use covering indexes so only small number of pages are read.
 - Utilize the memory (buffer cache) for reads. Monitor buffer cache hit ratio. Should ideally be 100% most of the time.
 - Tune the autovacuum process (PostgreSQL only).
 - Use native snapshots when possible. Logical backups (mysqldump, pg_dump) will generate excessive reads.
 - Use Aurora native replication (read replicas) when possible. Avoid logical replication (binlogs\WAL)
- Best Practices to optimize Write IO cost:
 - Tune Write IO intensive queries.
 - Find and remove unused and duplicate indexes to avoid excessive writes.
 - Use table partitioning.
 - Make use of appropriate fill factor so HOT (Heap Only Tuple) updates can be used (PostgreSQL only).

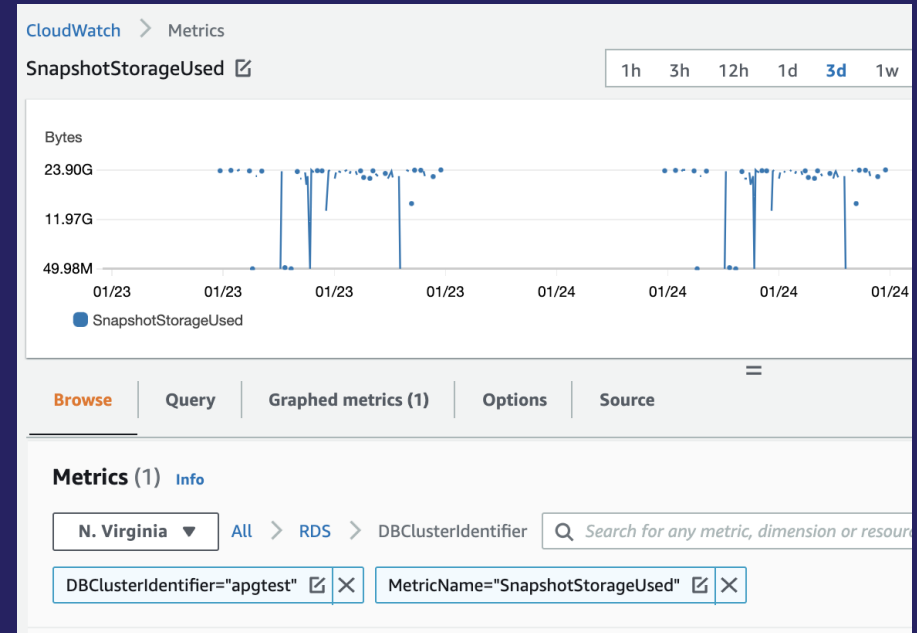
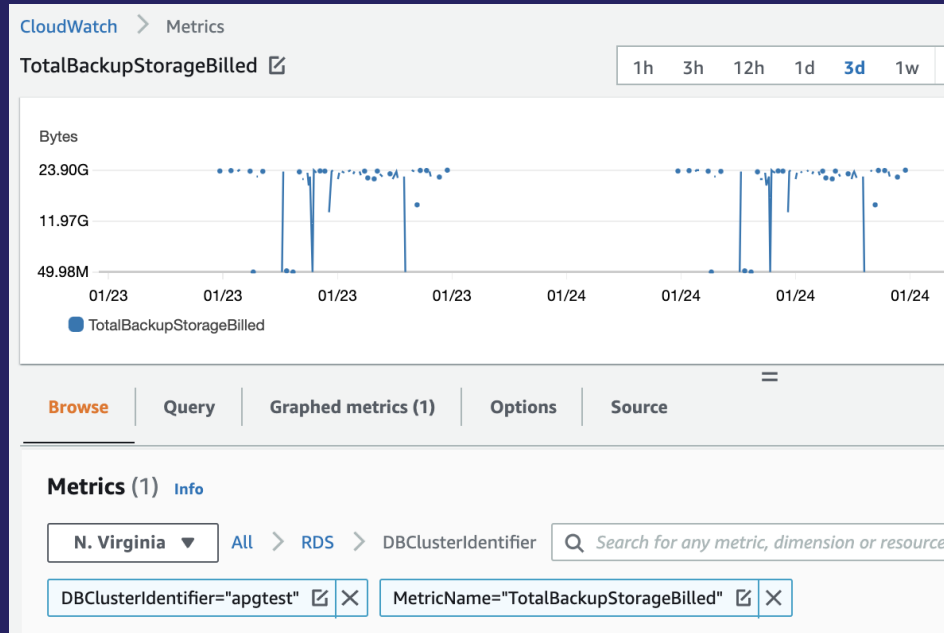
Backup and Snapshot cost

- Automated Backups and manual snapshots.
- No charge for backups up to 100% of total Aurora database size.
 - Example: If your database size is 100GB, and your backup size is 100 GB. You are not charged for backups.
- If backup retention period is 1 day, you are not charged for the backup.
- Billed as per GB-month. Average backup storage space used through the month is billed.
 - Example : US-EAST \$0.021 per GB-month.
- Manual snapshots that fall within the retention period are not charged separately during the period.

Backup cost optimization

- CloudWatch Metrics to monitor:
 - Snapshot Storage Used (GiB)
 - Backup Retention Period Storage Used (GiB)
 - Total Backup Storage Used (GiB)
- Carefully evaluate backup retention period as appropriate per your policies.
- Higher backup retention will incur higher backup cost.
- Keep an eye on manual snapshots. Manual snapshots are not deleted automatically.
- Lingering manual snapshot can add to a considerable cost.

CloudWatch Metrics –Backup



Data transfer cost

- Data transfer cost is billed by GB data transferred “IN” and “OUT” of an Aurora cluster.
- Data transfer for in region cluster replication is free.
- Data transfer between EC2 client and Aurora in the same AZ is free.
- Data transfer between Aurora and EC2 in different AZs and regions is billed.
- Data Transfer IN from internet (VPN) is free. Outgoing data transfer has tiered charges.
- CloudWatch Metrics to monitor:
 - Network In (Bytes)
 - Network Out(Bytes)

Data transfer cost optimization

- Carefully plan your application location.
- Clients in multiple AZs provide higher resiliency, but also add to data transfer cost.
- When possible use VPC endpoints (S3, DynamoDB).

Aurora features, cost and best practices

- Aurora Global Database
 - Billed for replicated write I/O between regions.
 - Instances, storage, cross-region data transfer, backup storage.
 - Optimize by choosing appropriate number of regions, replicas and headless cluster.
- Aurora Fast Clones
 - Since no pages are moved, no additional storage cost initially.
 - Instances, backup storage, cost applies.
 - Optimize by dropping unneeded clusters, using headless cluster.
- Snapshot Export
 - Billed per GB of snapshot size (\$.010 in US-EAST)
 - Example: Export of 100GB snapshot will cost $100 \times \$ 0.01 = \1 . Storage, encryption and PUT requests are charged by S3 separately.
 - Optimize by dropping unneeded snapshots exports.

Aurora features, cost and best practices (Continued)

- Backtrack
 - Billed per 1 million change records. (\$0.012 per 1 million change records in US-EAST)
 - CloudWatch Metrics: Backtrack Change Records Stored (Count), Backtrack Change Records Creation Rate (Count), Backtrack Window Actual (Minutes)
 - Optimize by carefully setting backtrack window based on SLAs.

Example:

Backtrack Window : 10 Hours

Change records/hour : 10,000

Billed amount per hour : $\$0.012 \times (100,000 / 1,000,000) = \0.0012

Tools available for cost tracking and estimation

- CloudWatch metrics – track usage
- Cost Explorer – overall billing
- Cost Allocation Tags
- Cost and Usage Reports
- AWS Pricing Calculator

Cost Explorer

AWS Cost Management

Home
Cost Explorer
 Reports
 Budgets
 Cost Anomaly Detection
 Rightsizing recommendations
 Savings Plans
 Overview
 Inventory
 Recommendations
 Purchase Savings Plans
 Utilization report
 Coverage report
 Cart 0
 Reservations
 Overview
 Recommendations
 Utilization report
 Coverage report
 Preferences
 Billing Console
 Documentation

AWS Cost Management > Cost Explorer

Save Save as... Recent reports New report

cost_explorer_report

Jan 02, 2022 - Jan 24, 2022 Monthly

Group by: Usage Type Service Linked Account Region Instance Type Resource Cost Category Tag More

Costs (\$ in thousands)

Jan 2022*

Legend: RDS:Multi-AZ-PIOPS, Multi-AZUsage:db.r5.8xl, InstanceUsage:db.r5.2xl, USE1-NodeUsage:db.r6g.large, RDS:PIOPS, Others

To see usage data, filter by "Usage Type" or "Usage Type Group" filters with matching units (e.g., hours).

Download CSV

Usage Type	Jan 2022*	Usage Type Total
Total cost (\$)	13,559.78	13,559.78
RDS:Multi-AZ-PIOPS (\$)	2,847.31	2,847.31
Multi-AZUsage:db.r5.8xl (\$)	7,108.87	2,037.17

cost_explorer_report

Jan 02, 2022 - Jan 24, 2022 Monthly

Group by: Usage Type Service Linked Account Region Instance Type Resource Cost Category Tag More

Costs (\$)

Jan 2022*

Legend: InstanceUsage:db.r5.2xl, InstanceUsage:db.r5.large, USE1-RDS:ProxyUsage, Aurora:StorageUsage, Aurora:BackupUsage

To see usage data, filter by "Usage Type" or "Usage Type Group" filters with matching units (e.g., hours).

Download CSV

Usage Type	Jan 2022*	Usage Type Total
Total cost (\$)	141.78	141.78
InstanceUsage:db.r5.2xl (\$)	117.26	117.26
InstanceUsage:db.r5.large (\$)	8.97	8.97
USE1-RDS:ProxyUsage (\$)	8.35	8.35
Aurora:StorageUsage (\$)	7.01	7.01
Aurora:BackupUsage (\$)	0.20	0.20
Aurora:StorageUsage (\$)	0.20	0.20
Aurora:BackupUsage (\$)	0.00	0.00

FILTERS CLEAR ALL

- Service Include all
- Linked Account Include all
- Region Include all
- Instance Type Include all
- Usage Type Include all
- Usage Type Group Include all
- Resource Include all
- Cost Category Include all
- Tag Include all
- CostTest Include only
 - AuroraPG Include all
 - AuroraPG1 Include all
- API Operation Include all
- Charge Type Include all
- Availability Zone Include all
- Platform Include all
- Purchase Option Include all
- Tenancy Include all
- Database Engine Include all
- Legal Entity Include all
- Billing Entity Include all

ADVANCED OPTIONS

Show costs as Net unblended costs

Include costs related to



Cost Allocation Tag

RDS > Databases > iopstest1

iopstest1

Related

Filter by databases

DB identifier	Role	Engine	Region & AZ
iopstest1	Regional cluster	Aurora PostgreSQL	us-east-1
iopstest1-instance-1	Writer instance	Aurora PostgreSQL	us-east-1a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | **Tags**

Tags (1)

Filter by tag key

Tags	Value
CostTest	iopstest1

AWS Billing > Cost allocation tags

Cost allocation tags Info

User-defined cost allocation tags | AWS-generated cost allocation tags

User-defined cost allocation tags (1/71) Info

Undo | Deactivate | **Activate**

Search for a tag key | All statuses

Tag key	Status
<input type="checkbox"/> AWSServiceName	Inactive
<input type="checkbox"/> AmazonMemoryDBManaged	Inactive
<input type="checkbox"/> Application	Inactive
<input type="checkbox"/> ApplicationVersion	Inactive
<input type="checkbox"/> AuroraDb	Inactive
<input type="checkbox"/> CL9Name	Inactive
<input type="checkbox"/> Compliance	Inactive
<input type="checkbox"/> Confidentiality	Inactive
<input checked="" type="checkbox"/> CostTest	Inactive

Cost and Usage Reports

- Billing -> Cost & Usage Reports -> Create Report

AWS Cost and Usage Reports > Create report

Step 1
Report content

Step 2
Delivery options

Step 3
Review

Delivery options

S3 bucket - required
pglogsforkdf Valid Bucket

Report path prefix - required

Time granularity
 Hourly
 Daily
 Monthly
The time granularity on which report data are measured and displayed.

Report versioning
 Create new report version
 Overwrite existing report

Enable report data integration for
 Amazon Athena
 Amazon Redshift
 Amazon QuickSight

Compression type

File format
Parquet

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings

Data

Data Source: AwsDataCatalog
Database: auroradbathena

Tables and views

▼ Tables (1)
aurora_db_athena
Partitioned

▼ Views (0)

```
1 select
2   line_item_usage_start_date,
3   line_item_usage_end_date,
4   line_item_product_code,
5   line_item_usage_type,
6   line_item_operation,
7   line_item_usage_amount,
8   resource_tags_user_cost_test
9 from AuroraDBAthena.aurora_db_athena
10 where line_item_product_code = 'AmazonRDS'
11 ;
```

SQL Ln 2, Col 5

Completed Time in queue: 0.13 sec Run time: 1.975 sec

Results (100+)

line_item_usage_start_date	line_item_usage_end_date	line_item_product_code	line_item_usage_type	line_item_operation	line_item_usage_t
2022-01-14 18:00:00.000	2022-01-14 19:00:00.000	AmazonRDS	USE2-DataTransfer-Out-Bytes	CreateDBInstance:0021	1.57841E-5
2022-01-14 22:00:00.000	2022-01-14 23:00:00.000	AmazonRDS	USE2-DataTransfer-Out-Bytes	CreateDBInstance:0021	1.23158E-5
2022-01-15 01:00:00.000	2022-01-15 02:00:00.000	AmazonRDS	USE2-DataTransfer-Out-Bytes	CreateDBInstance:0021	1.55717E-5
2022-01-15 02:00:00.000	2022-01-15 03:00:00.000	AmazonRDS	USE2-DataTransfer-Out-Bytes	CreateDBInstance:0021	1.23158E-5

Demo!

- Setup cost allocation tags
- Setup and analyze Cost and Usage Reports
- Track resource usage with CloudWatch & Performance Insights

Resources

- Cost Explorer

<https://aws.amazon.com/aws-cost-management/aws-cost-explorer/>

- AWS pricing Calculator

<https://calculator.aws/>

- Planning Aurora IO cost

<https://aws.amazon.com/blogs/database/planning-i-o-in-amazon-aurora/>

- Planning Aurora IO cost

<https://docs.aws.amazon.com/cur/latest/userguide/what-is-cur.html>

- Aurora pricing page

<https://aws.amazon.com/rds/aurora/pricing/>

Partner Packages – Aurora Cost Optimization



Cost
Optimization Offer



Cost
Optimization Offer



Cost
Optimization Offer



Cost
Optimization Offer



Cost
Optimization Offer



Cost
Optimization Offer





Thank you!