Amazon Aurora pricing

For Aurora PostgreSQL and Aurora MySQL

With Aurora, you only pay for what you use. Aurora charges for database instances and storage, along with any optional features you choose to enable. Aurora PostgreSQL and Aurora MySQL offer On-Demand and Reserved Instance pricing. Aurora DSQL has a serverless pricing model, and you can learn more on Aurora DSQL pricing page.

Your Aurora usage may be eligible for Database Savings Plans, a flexible pricing model that offers savings in exchange for a commitment to a specific amount of usage (measured in $/hour) over a 1-year term. You can learn more about eligible usage on the Database Savings Plans pricing page.

Aurora cluster configuration

With Aurora PostgreSQL and Aurora MySQL, you can configure your database clusters to run **cost effectively**, regardless of the scaling needs or evolving data access patterns of your applications. You have the flexibility to choose between the Aurora Standard and Amazon Aurora I/O-Optimized configuration options to best match the **price-performance** and **price-predictability** requirements of your unique workload characteristics. Your database instance, storage, and I/O charges will vary based on the option you choose. Instance charges apply to both Aurora primary instances and replicas.

Aurora I/O-Optimized delivers **improved price performance** for I/O-intensive applications and predictable pricing. With Aurora I/O-Optimized, if your I/O spend exceeds 25% of your total Aurora database spend, you can **save up to 40% on costs**. You only pay for your database instances and storage usage, and there are zero charges for read and write I/O operations.

Aurora Standard offers cost-effective pricing for many applications running on Aurora with typical data access patterns and low to moderate I/O usage. With Aurora Standard, you pay for your database instances, storage, and pay-per-request I/O.

Pricing by database instances

With Aurora PostgreSQL and Aurora MySQL, you have the choice of:

1. Amazon Aurora Serverless which automatically starts up, shuts down, and scales capacity up or down based on your application's needs. You pay only for capacity consumed.
2. Provisioned instances operate within the parameters of what instance type has been provisioned. You pay per DB instance-hour consumed with no long-term commitments or upfront fees with On-Demand, or choose Provisioned Reserved Instances for additional savings.
3. Amazon Aurora PostgreSQL Limitless Database is available for workloads that require massive horizontal scale. It can process millions of write transactions per second and manages petabytes of data while maintaining the simplicity of operating inside a single database.

Select Aurora MySQL-Compatible Edition or Aurora PostgreSQL-Compatible Edition to view database instance pricing.

Aurora MySQL-Compatible Edition

Serverless | Provisioned On-Demand Instance | Provisioned Reserved Instance

Amazon Aurora Serverless is an on-demand, auto scaling configuration that automatically adjusts database capacity based on application needs. With Aurora Serverless, your database capacity automatically scales up or down to meet your application workload needs and shuts down during periods of inactivity, saving you money and administration time. Aurora Serverless measures database capacity in Aurora Capacity Units (ACUs) billed per second. 1 ACU has approximately 2 GiB of memory with corresponding CPU and networking, similar to that used in Aurora provisioned instances.

Aurora Serverless

Amazon Aurora Serverless instantly scales databases to support hundreds of thousands of transactions per second and supports all features of Aurora, including Multi-AZ deployments, read replicas, and Global Database.

You can choose to configure all the instances in your database cluster to use the Aurora Standard or Aurora I/O-Optimized configuration based on the price-performance and price-predictability needs of your application.

Region:
Europe (London) ▼

Measure	Aurora Standard (per ACU hour)	Aurora I/O-Optimized (per ACU hour)
Aurora Capacity Unit	$0.14	$0.19

Scale up latency	Instant scaling to hundreds-of-thousands of transactions per second
Scale down latency	Up to 15X faster
Starting capacity	0 ACU
Capacity granularity	Fine-grained, with increments as small as 0.5 ACU
Read replicas	Up to 15 Aurora replicas for read scalability
Multi-AZ and SLA	Distribute read replicas in separate AZs for high availability. Refer Amazon Aurora SLA for details.
Aurora Global Database	Sub-Second Data Access in Any Region and Cross-Region Disaster Recovery. Refer Aurora Global Database for details.

Aurora PostgreSQL-Compatible Edition

Serverless | Provisioned On-Demand Instance | Provisioned Reserved Instance | Limitless Database

Amazon Aurora Serverless is an on-demand, auto scaling configuration that automatically adjusts database capacity based on application needs. With Aurora Serverless, your database capacity automatically scales up or down to meet your application workload needs and shuts down during periods of inactivity, saving you money and administration time. Aurora Serverless measures database capacity in Aurora Capacity Units (ACUs) billed per second. 1 ACU has approximately 2 GiB of memory with corresponding CPU and networking, similar to that used in Aurora provisioned instances.

Aurora Serverless

Amazon Aurora Serverless instantly scales databases to support hundreds of thousands of transactions per second and supports all features of Aurora, including Multi-AZ deployments, read replicas, and Global Database.

You can choose to configure all the instances in your database cluster to use the Aurora Standard or Aurora I/O-Optimized configuration based on the price-performance and price-predictability needs of your application.

Region:
Europe (London) ▼

Measure	Aurora Standard (per ACU hour)	Aurora I/O-Optimized (per ACU hour)
Aurora Capacity Unit	$0.14	$0.19

Scale up latency	Instant scaling to hundreds-of-thousands of transactions per second
Scale down latency	Up to 15X faster
Starting capacity	0 ACU
Capacity granularity	Fine-grained, with increments as small as 0.5 ACU
Read replicas	Up to 15 Aurora replicas for read scalability
Multi-AZ and SLA	Distribute read replicas in separate AZs for high availability. Refer Amazon Aurora SLA for details.
Aurora Global Database	Sub-Second Data Access in Any Region and Cross-Region Disaster Recovery. Refer Aurora Global Database for details.

Pricing by database storage and I/Os

With Aurora, you do not need to provision either storage or I/O operations in advance, and both scale automatically with the demands of your application. **The fault-tolerant distributed storage of Aurora automatically makes your data durable across three Availability Zones in a Region. You only pay for one copy of the data.**

Aurora storage is billed in per GB-month increments at the rates shown in the following table for the Aurora Standard and Aurora I/O-Optimized configurations. With Aurora Standard, you pay for the storage and I/O operations that your Aurora database consumes. I/O charges are billed in per million request increments and may vary depending on workload and database engine. With Aurora I/O-Optimized, you are not charged for read and write I/O operations.

See below database storage and I/O operations pricing example and FAQs to learn more about I/O operations and how its calculated.

Region:
Europe (London) ▼

Component	Aurora Standard	Aurora I/O-Optimized
Storage Rate	$0.10 per GB-month	$0.225 per GB-month
I/O Rate	$0.20 per 1 million requests	Included

AWS Pricing Calculator

Calculate your Amazon Aurora and architecture cost in a single estimate.

Create your custom estimate for Aurora MySQL now »

Create your custom estimate for Aurora PostgreSQL now »

Additional features and costs

Aurora Global Database costs

Amazon Aurora Global Database is designed for globally distributed applications, allowing a single Aurora database to span multiple Regions. It replicates your data with no impact on database performance, enables fast local reads with low latency in each Region, and provides disaster recovery from Region-wide outages.

With Aurora Global Database, you pay for replicated write I/O operations between the primary Region and each secondary Region. The number of replicated write I/O operations to each secondary Region is the same as the number of in-Region write I/O operations performed by the primary Region. You pay for replicated write I/O operations with both Aurora Standard and Aurora I/O-Optimized configuration options. Apart from replicated write I/Os, you pay for instances, storage, and I/O usage in the primary and secondary Regions based on the cluster configuration you choose, along with cross-Region data transfer, backup storage, and other billable Aurora features.

Region:
Europe (London) ▼

Measure	Pricing
Backup Storage	$0.022 per GB-month

Optimized reads instances for Aurora PostgreSQL costs

Optimized reads instances available for Amazon Aurora PostgreSQL for read-heavy applications with large datasets that exceed instance memory and require fast query performance. It uses local NVMe-based SSD block-level storage to improve query latency of applications with datasets exceeding the memory capacity of a database instance. There is no incremental cost to using optimized reads.

Optimized reads instances includes: tiered caching and temporary objects.

Tiered caching delivers up to 8x improved query latency and up to 30% cost savings for read-heavy, I/O-intensive applications, such as operational dashboards, anomaly detection, and vector-based similarity searches. It automatically stores data evicted from the in-memory database buffer cache onto local storage to speed up subsequent data accesses. Tiered caching is available for Aurora PostgreSQL with the Aurora I/O-Optimized configuration.

Temporary objects achieve faster query processing by placing temporary tables on local storage, improving the performance of queries involving sorts, hash aggregations, high-load joins, and other data-intensive operations. Temporary objects are available for Aurora PostgreSQL with the Aurora I/O-Optimized and Aurora Standard configurations.

Backtrack costs

Backtrack lets you quickly move an Aurora database to a prior point in time without needing to restore data from a backup. This lets you quickly recover from user errors, such as dropping the wrong table or deleting the wrong row. This feature is currently available for Aurora MySQL.

Specify how far in the past you want to be able to backtrack (for example, up to 24 hours). Aurora will retain logs, called change records, for the specified backtrack duration. You pay an hourly rate for storing change records.

Region:
Europe (London) ▼

Measure	Price Per Hour
Change Records	$0.014 per 1 million Change Records

Zero-ETL integration costs

AWS does not charge an additional fee for zero-ETL integrations. You pay for existing resources used to create and process the change data created as part of a zero-ETL integration. These resources could include additional I/O and storage used by enabling enhanced binlog, Snapshot export costs for the initial data export to seed your databases, additional storage for storing replicated data and compute for processing data replication, and cross-AZ data transfer costs for moving data from source to target. Ongoing processing of data changes by zero-ETL integration is offered at no additional charge.

Backup storage costs

Backup storage for Aurora is the storage associated with your automated database backups and any customer-initiated database cluster snapshots.

You are **not charged for the backup storage of up to 100% of the size of your database cluster.** There is also **no charge for database snapshots created within the backup retention period.** For all other backups and snapshots (including those from deleted clusters), usage is metered per GB-month at the following rates:

Region:
Europe (London) ▼

Measure	Pricing
Backup Storage	$0.022 per GB-month

Data API costs

Data API is an easy-to-use, secure HTTPS API for executing SQL queries against Aurora databases that accelerates modern application development. Data API eliminates network and application configuration tasks needed to securely connect to an Aurora database, which makes accessing Aurora as simple as making an API call.

With Data API, you only pay when your API is in use. There are no minimum fees or upfront commitments. You pay only for the API and data requests your applications make. Data API request data payloads are metered at 32KB per request for data sent to the API or received from the API. If your API request data payload size is larger than 32KB, then for each 32KB increment there is a charge of one additional API request. So, if your payload is 35KB for example, you would be charged for two API requests.

Data API free tier includes one million API requests per month, aggregated across AWS Regions, for the first year.

Customers will also incur charges for AWS Secrets Manager and may also incur additional charges for AWS CloudTrail, if activated.

Region:
Europe (London) ▼

API Requests

Number of Requests (per month)	Price (per million)
First 1 Billion requests	$0.397
Above 1 Billion requests	$0.227

Aurora PostgreSQL and Aurora MySQL are each billed separately for the purposes of price tiers.

Data transfer costs

The pricing below is based on data transferred "in" and "out" of Aurora.

- As part of the AWS Free Tier, AWS customers receive 100 GB of free data transferred out to the internet for free each month, aggregated across all AWS services and Regions (except China (Beijing), China (Ningxia), and GovCloud (US)).
- Data transferred between Aurora and Amazon Elastic Compute Cloud (Amazon EC2) instances in the same Availability Zone is free.
- **Data transferred between Availability Zones for DB cluster replication is free.**
- For data transferred between an Amazon EC2 instance and Aurora DB instance in different Availability Zones of the same Region, Amazon EC2 Regional Data Transfer charges apply.

Region:
Europe (London) ▼

Data Transfer IN To Amazon RDS From Internet	Pricing
All data transfer in	$0.00 per GB

Amazon RDS Extended Support costs

Amazon RDS Extended Support allows you to continue use Aurora MySQL-Compatible and PostgreSQL-Compatible Edition major versions after the community end-of-life. During this time, AWS provides fixes for critical security issues and bugs through patch releases, giving you more time, up to three years, to upgrade to a new major version to help you meet your business requirements.

For provisioned instances on Amazon Aurora, RDS Extended Support is priced per vCPU per hour. For Aurora Serverless, RDS Extended Support is priced per Aurora Capacity Unit (ACU) per hour consumed by your database. RDS Extended Support pricing is also dependent on AWS Region and the calendar date. Aurora MySQL-Compatible Edition customers have at least a year to migrate to a standard support version after community end of life before RDS Extended Support charges begin. Refer to the Amazon Aurora documentation for calendar details.

Region:
Europe (London) ▼

	Year 1 and Year 2 Pricing	Year 3 Pricing*
Aurora Serverless V2	$0.10 per ACU-hr	$0.20 per ACU-hr
Provisioned Instances	$0.118 per VCPU-hr	$0.235 per VCPU-hr

Partial hours are billed in one-second increments with a 10-minute minimum charge following a billable status change such as creating, starting, or modifying.

*Amazon RDS Extended Support year 3 pricing is only available for Amazon Aurora PostgreSQL-Compatible Edition.

Snapshot or cluster export costs

Snapshot export provides an automated method to export data within an Aurora snapshot to Amazon Simple Storage Service (Amazon S3) in Parquet format. Cluster export provides an automated method to export data from an Aurora database cluster to Amazon S3 in Parquet format. The Parquet format is up to 2x faster to unload and consumes up to 6x less storage in Amazon S3 compared to text formats. You can analyze the exported data using AWS services such as Amazon Athena, Amazon EMR, and Amazon SageMaker.

For all snapshots and cluster exports, usage is metered per gigabyte of snapshot data. Subsequent exports of data from the same snapshot are not incremental.

Region:

Europe (London) ▼

Measure	Pricing
Charge per GB of snapshot size:	$0.012

Additional charges apply for encrypting or decrypting data with AWS Key Management Service (AWS KMS). Find more about these charges on the AWS KMS pricing page. Charges also apply for storing exported data in Amazon S3 and for PUT requests made against your S3 bucket. Find more about these charges on the Amazon S3 pricing page.

Feature availability and support varies across specific versions of each database engine and across AWS Regions. For more information on version and Region availability of exporting DB cluster snapshot data to S3, see Exporting snapshot data to Amazon S3.

Aurora pricing examples

Aurora Serverless pricing with Standard and I/O-Optimized configurations

Consider a workload that needs 5 ACUs and runs for 30 minutes. We want to calculate the compute costs of running this workload on Aurora Serverless with the database cluster configured as Aurora Standard and Aurora I/O-Optimized in US East (N. Virginia).

With **Aurora Serverless**, the minimum database capacity you can set is 0.5 ACUs. The database scales instantly from 0.5 ACUs to 5 ACUs, runs for 30 minutes, and scales down to the minimum capacity of 0.5 ACUs. Since it scales down up to 15x faster, the database goes from 5 ACUs to 0.5 ACUs in approximately 3 minutes. Therefore, the following is the compute cost for running this workload.

Example 1: Aurora Serverless pricing with Aurora Standard

Usage	Calculation	Cost
Scale from 0.5 ACUs to 5 ACUs instantly		
Run at 5 ACUs for 30 minutes	5 ACUs * $0.12 per ACU-hour * 30/60 hour	$0.30
Scale down from 5 ACUs to 0.5 ACUs in 3 minutes	5 ACUs * $0.12 per ACU-hour * 3/60 hour	$0.03
Compute cost with Aurora Serverless (Aurora Standard)		$0.33

Example 2: Aurora Serverless pricing with Aurora I/O-Optimized

Usage	Calculation	Cost
Scale from 0.5 ACUs to 5 ACUs instantly		
Run at 5 ACUs for 30 minutes	5 ACUs * $0.156 per ACU-hour * 30/60 hour	$0.39
Scale down from 5 ACUs to 0.5 ACUs in 3 minutes	5 ACUs * $0.156 per ACU-hour * 3/60 hour	$0.04
Compute cost with Aurora Serverless (Aurora I/O-Optimized)		$0.43

Aurora Standard Reserved Instances with Aurora I/O-Optimized

You can reuse your existing Aurora Standard Reserved Instances (RI) with Aurora I/O-Optimized. To fully utilize the benefits of your RI discounts with Aurora I/O-Optimized, you can purchase 30% additional RIs similar to your current RIs. The table below shows examples of how to estimate additional RIs when using Aurora I/O-Optimized. If the required RIs are a fraction, you can take advantage of the size flexibility available with RIs to get to a whole number as shown below. In the examples below, current RIs refer to the Aurora Standard RIs you have today and additional RIs refer to the number of Aurora Standard RIs you must purchase to maintain your current RI discounts when using Aurora I/O-Optimized. If you choose not to purchase additional RIs, On-Demand rates will apply.

Instance type	Quantity of current RIs	Required RI quantity for Aurora I/O-Optimized	Additional RIs	Additional RIs (with size flexibility)
db.r6g.large	10	10 * 1.3 = 13	3 db.r6g.large	3 db.r6g.large RIs
db.r6g.4xlarge	20	20 * 1.3 = 26	6 db.r6g.4xlarge	6 db.r6g.4xlarge RIs
db.r6i.24xlarge	15	15 * 1.3 = 19.5	4.5 db.r6i.24xlarge	4 db.r6i.24xlarge RIs + 1 db.r6i.12xlarge RI (0.5 db.r6i.24xlarge = 1 db.r6i.12xlarge)
db.r6g.12xlarge	5	5 * 1.3 = 6.5	1.5 db.r6g.12xlarge	1 db.r6g.12xlarge RIs + 1 db.r6g.4xlarge RI + 1 db.r6g.2xlarge RI (0.5 db.r6g.12xlarge = 1 db.r6g.4xlarge + 1 db.r6g.2xlarge)

Database storage and I/O operations examples

Let's assume you have an Aurora database and you start with a 1,000 GB database at the beginning of the month (30-day period) that is growing 2% or 20 GB daily. Let's also assume the same database reads 350 data pages from storage per second to satisfy the queries running on it. Finally, let's assume your application makes changes to the database affecting an average of 100 data pages per second. We want to calculate the storage and I/O cost of running this workload with Aurora Standard and Aurora I/O-Optimized storage in US East (N. Virginia).

Example 1: Pricing with Aurora Standard

Usage	Calculation	Cost
Storage for 1,000 GB growing 2% daily for 30 days	(1,000 GB * 30 days + 20 GB * 29 days + 20 GB * 28 days + …. + 20 GB * 1 day) * $0.10 per GB-month	$129.00
Read I/Os for 350 data pages per second for 30 days (assuming 16 KB page size)	(350 data pages * 30 days * 24 hours * 60 minutes * 60 seconds) * $0.20 per 1 million I/Os	$181.44
Write I/Os for 100 data pages per second for 30 days (assuming 4 KB page size)	(100 data pages * 30 days * 24 hours * 60 minutes * 60 seconds) * $0.20 per 1 million I/Os	$51.84
Storage and I/O cost with Aurora Standard		$362.28

Example 2: Pricing with Aurora I/O-Optimized

Usage	Calculation	Cost
Storage for 1,000 GB growing 2% daily for 30 days	(1,000 GB * 30 days + 20 GB * 29 days + 20 GB * 28 days + …. + 20 GB * 1 day) * $0.225 per GB-month	$290.25
Read I/Os for 350 data pages per second for 30 days (assuming 16 KB page size)	(350 data pages * 30 days * 24 hours * 60 minutes * 60 seconds) * $0 for read I/Os	-
Write I/Os for 100 data pages per second for 30 days (assuming 4 KB page size)	(100 data pages * 30 days * 24 hours * 60 minutes * 60 seconds) * $0 for write I/Os -	-
Storage and I/O cost with Aurora I/O-Optimized		$290.25

For this workload, you will get 19.8% storage **cost savings** with Aurora I/O-Optimized.

Aurora Global Database examples

Let's assume you have an Aurora Global Database where the primary Region has two instances, consuming 80 GB of storage and performing 45 million write I/O operations and 5 million read I/O operations per month. Say the secondary Region has one instance serving 5 million read I/O operations per month. We want to calculate the cost of running a Global Database with US East (N. Virginia) as the primary Region and US West (Oregon) as the secondary Region.

Example 1: Aurora Global Database with Aurora Standard

Primary Region - US East (N. Virginia)

Usage	Calculation	Cost
Database instance	2 * db.r6i.large (at $0.29 per hour) * 30 days * 24 hours	$417.60
Storage	80 GB * $0.10 per GB-month (30 days)	$8.00
Replicated write I/Os	None	-
Read and write I/Os	50 million I/Os * $0.20 per 1 million I/Os	$10.00
Data transfers	45 million write I/Os * 2.75 KB per I/O * $0.02 per GB	$2.48
Primary Region cost with Aurora Standard		**$438.08**

Secondary Region - US West (Oregon)

Usage	Calculation	Cost
Database instance	1 * db.r6i.large (at $0.29 per hour) * 30 days * 24 hours	$208.80
Storage	80 GB * $0.10 per GB-month (30 days)	$8.00
Replicated write I/Os	45 million replicated write I/Os * $0.20 per 1 million I/Os	$9.00
Read and write I/Os	50 million I/Os * $0.20 per 1 million I/Os	$10.00
Data transfers	None	-
Secondary Region cost with Aurora Standard		**235.80**

Example 2: Aurora Global Database with Aurora I/O-Optimized

Primary Region - US East (N. Virginia)

Usage	Calculation	Cost
Database instance	2 * db.r6i.large (at $0.377 per hour) * 30 days * 24 hours	$542.88
Storage	80 GB * $0.225 per GB-month (30 days)	$18.00
Replicated write I/Os	None	-
Read and write I/Os	50 million I/Os * $0 per 1 million I/Os	-
Data transfers	45 million write I/Os * 2.75 KB per I/O * $0.02 per GB	$2.48
Primary Region cost with Aurora I/O-Optimized		**$563.36**

Secondary Region - US West (Oregon)

Usage	Calculation	Cost
Database instance	1 * db.r6i.large (at $0.377 per hour) * 30 days * 24 hours	$271.44
Storage	80 GB * $0.225 per GB-month (30 days)	$18.00
Replicated write I/Os	45 million replicated write I/Os * $0.20 per 1 million I/Os	$9.00
Read and write I/Os	50 million I/Os * $0 per 1 million I/Os	-
Data transfers	None	-
Secondary Region cost with Aurora I/O-Optimized		**$298.44**

Data API examples

Example 1: Data API pricing with mixed INSERTS and read requests

Data API is called to query Aurora PostgreSQL-Compatible with the ExecuteStatement API call. There are 4.1 million API requests a month in total. 2.1 million of the of the 4.1M requests are inserts into Aurora. Each has data in an API payload size of 64 KB/request. 2 million of the 4.1M total requests read Aurora data. Of these, 1 million had a data returned payload of less than 32KB and 1 million had a payload size between 32.01KB up to 64KB.

Usage	Calculation	Cost
2.1 million API requests/month with a payload of 64 KB/request, where each API requires 2 requests (64 KB requires 64 KB/32 KB, or 2 requests per API)	2.1 million * 2 * $0.35/million	$1.47
2 million API requests/month where 1 million have a payload of 32 KB and 1 million with a payload between 32.01 KB to 64 KB that requires 2 requests (32.01 KB to 64 KB/32 KB, or 2 requests per API)	(1 million API * $0.35/million) + (1 million API * 2 * $0.35/million APIs)	$1.05
Free tier 1 million API requests for one year	-$0.35 for 1 million API requests for one year	-$0.35
Data API costs		**$2.17**

Example 2: Data API pricing with free tier

Data API is called to query Aurora PostgreSQL-Compatible with the ExecuteStatement API call. There are 300,000 API requests a month. All of the API requests return 64 KB of data payload so each request will be charged as 64 KB / 32 KB = 2 requests.

Usage	Calculation	Cost
300,000 API requests with a payload of 64 KB where each API requires 2 requests (64 KB requires 64 KB / 32 KB, or 2 API requests)	.3 million APIs * 2 * $0.35/million APIs	$0.21
Free tier 1 million API requests for one year	-$0.35 for 1 million API requests for one year	-$0.35
Data API costs		**$0.00/month in first year**

Example 3: Data API pricing using all request types

Data API is called to query Aurora PostgreSQL-Compatible using all five API Calls (BeginTransaction. CommitTransaction, RollbackTransaction, ExecuteStatement, BatchExecuteStatement. There are 1.2 billion API requests a month. None of the requests return or send more than 32KB of data payload.

Usage	Calculation	Cost
1 billion API requests/month with a payload less than 32 KB	1 billion API requests * $0.35/million	$350
200 million API requests/month with a payload less than 32 KB	200 million API requests * $0.20/million	$40
Free tier 1 million API requests for one year	-$0.35 for 1 million API requests for one year	-$0.35
Data API costs		**$389.65/month in first year**

RDS Extended Support example

If you have an instance on Aurora PostgreSQL-Compatible 12, this version reaches end of standard support on February 28, 2025. If you deploy in US East (Ohio), you will be charged $0.100 per vCPU-hr between March 1, 2025 to February 28, 2027. Starting March 1, 2027, you will be charged $0.200 per vCPU-hr.

Snapshot or cluster export example

Let's assume you have a 100 GB snapshot and you use filtering to select a 10 GB table from this snapshot to export to Amazon S3.

Usage	Calculation	Cost
Export 10 GB table from 100 GB snapshot (Subsequent exports of data from the same snapshot are not incremental)	100 GB * $0.010 per GB	$1.00
Export cost		**$1.00**

Next Steps

Resources

Learn with hands-on training →

Documentation

Dive deep on Aurora →

Free tier

Sign up for a free account →

Console

Start building in the console →

Learn
- What Is AWS?
- What Is Cloud Computing?
- What Is Agentic AI?
- Cloud Computing Concepts Hub
- AWS Cloud Security
- What's New
- Blogs
- Press Releases

Resources
- Getting Started
- Training
- AWS Trust Center
- AWS Solutions Library
- Architecture Center
- Product and Technical FAQs
- Analyst Reports
- AWS Partners

Developers
- Builder Center
- SDKs & Tools
- .NET on AWS
- Python on AWS
- Java on AWS
- PHP on AWS
- JavaScript on AWS

Help
- Contact Us
- File a Support Ticket
- AWS re:Post
- Knowledge Center
- AWS Support Overview
- Get Expert Help
- AWS Accessibility
- Legal

Back to top ↑

Amazon is an Equal Opportunity Employer: Minority / Women / Disability / Veteran / Gender Identity / Sexual Orientation / Age.

Privacy Site terms Cookie Preferences © 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.