

What is Amazon Aurora?

[Amazon Aurora](#) is a modern [relational database](#) service offering performance and high availability at scale, fully open-source MySQL- and PostgreSQL-compatible editions, and a range of developer tools for building serverless and machine learning (ML)-driven applications.

Aurora features a distributed, fault-tolerant, and self-healing storage system that is decoupled from compute resources and auto-scales up to 128 TiB per database instance. It delivers high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon Simple Storage Service (Amazon S3), and replication across three Availability Zones (AZs).

Aurora is also a fully managed service that automates time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups while providing the security, availability, and reliability of commercial databases at one-tenth of the cost.

Is Amazon Aurora MySQL compatible?

Amazon Aurora is drop-in [compatible with existing MySQL](#) open-source databases and adds support for new releases regularly. This means you can easily migrate MySQL databases to and from Aurora using standard import/export tools or snapshots. It also means that most of the code, applications, drivers, and tools you already use with MySQL databases today can be used with Aurora with little or no change. This makes it easy to move applications between the two engines.

You can see the current Amazon Aurora MySQL release compatibility information in the [documentation](#).

Is Amazon Aurora PostgreSQL compatible?

Amazon Aurora is drop-in [compatible with existing PostgreSQL](#) open-source databases and adds support for new releases regularly. This means you can easily [migrate PostgreSQL databases to and from Aurora using standard import/export tools or snapshots](#). It also means that most of the code, applications, drivers, and tools you already use with PostgreSQL databases today can be used with Aurora with [little or no change](#).

You can see the current Amazon Aurora PostgreSQL release compatibility information in the [documentation](#).

How is Aurora PostgreSQL supported for issues related to PostgreSQL extensions?

Amazon fully supports Aurora PostgreSQL and all extensions available with Aurora. If you need support for Aurora PostgreSQL, reach out to AWS Support. If you have an active AWS Premium Support account, you can contact AWS Premium Support for Aurora specific issues.

How do I get started with Aurora?

To try Aurora, sign in to the [AWS Management Console](#), select RDS under the Database category, and choose Amazon Aurora as your database engine. For detailed guidance and resources, check out our [Getting started with Aurora](#) page.

In which AWS Regions is Aurora available?

You can see Region availability for Aurora [here](#).

How can I migrate from MySQL to Aurora and the other way around?

If you want to migrate from MySQL to Aurora and the other way around, you have several options:

- You can use the standard mysqldump utility to export data from MySQL and mysqlimport utility to import data to Aurora, and the other way around.
- You can also use [Amazon RDS DB Snapshot](#) migration feature to migrate an Amazon RDS for MySQL DB Snapshot to Aurora using the AWS Management Console.

Migration to Aurora completes for most customers in under an hour, though the duration depends on format and dataset size. For more information see [Best Practices for Migrating MySQL Databases to Amazon Aurora](#).

How can I migrate from PostgreSQL to Aurora and the other way around?

If you want to migrate from PostgreSQL to Aurora and the other way around, you have several options:

- You can use the standard pg_dump utility to export data from PostgreSQL and pg_restore utility to import data to Aurora, and the other way around.
- You can also use [RDS DB Snapshot](#) migration feature to migrate an Amazon RDS for PostgreSQL DB Snapshot to Aurora using the AWS Management Console.

Migration to Aurora completes for most customers in under an hour, though the duration depends on format and dataset size.

To migrate SQL Server databases to Amazon Aurora PostgreSQL-Compatible Edition, you can use [Babelfish for Aurora PostgreSQL](#). Your applications will work without any changes. See the [Babelfish documentation](#) for more information.

Do I need to change client drivers to use Amazon Aurora PostgreSQL-Compatible Edition?

No, Aurora works with standard PostgreSQL database drivers.

Performance

What does "five times the performance of MySQL" mean?

Amazon Aurora delivers significant increases over MySQL performance by tightly integrating the database engine with an SSD-based virtualized storage layer purpose-built for database workloads, reducing writes to the storage system, minimizing lock contention, and eliminating delays created by database process threads.

Our tests with SysBench on r3.8xlarge instances show that Amazon Aurora delivers over 500,000 SELECTs/sec and 100,000 UPDATEs/sec, five times higher than MySQL running the same benchmark on the same hardware. Detailed instructions on this benchmark and how to replicate it yourself are provided in the [Amazon Aurora MySQL-Compatible Edition Performance Benchmarking Guide](#).

What does "three times the performance of PostgreSQL" mean?

Amazon Aurora delivers significant increases over PostgreSQL performance by tightly integrating the database engine with an SSD-based virtualized storage layer purpose-built for database workloads, reducing writes to the storage system, minimizing lock contention, and eliminating delays created by database process threads.

Our tests with SysBench on r4.16xlarge instances show that Amazon Aurora delivers SELECTs/sec and UPDATEs/sec over three times higher than PostgreSQL running the same benchmark on the same hardware. Detailed instructions on this benchmark and how to replicate it yourself are provided in the [Amazon Aurora PostgreSQL-Compatible Edition Performance Benchmarking Guide](#).

How do I optimize my database workload for Amazon Aurora MySQL-Compatible Edition?

Amazon Aurora is designed to be compatible with MySQL so that existing MySQL applications and tools can run without requiring modification. However, [one area where Amazon Aurora improves upon MySQL is with highly concurrent workloads](#). In order to maximize your workload's throughput on Amazon Aurora, we recommend building your applications to drive a large number of concurrent queries and transactions.

How do I optimize my database workload for Amazon Aurora PostgreSQL-Compatible Edition?

Amazon Aurora is designed to be compatible with PostgreSQL so that existing PostgreSQL applications and tools can run without requiring modification. However, one area where Amazon Aurora improves upon PostgreSQL is with highly concurrent workloads. In order to maximize your workload's throughput on Amazon Aurora, we recommend building your applications to drive a large number of concurrent queries and transactions.

Billing

How much does Aurora cost?

See the [Aurora pricing page](#) for current pricing information.

Does Aurora participate in the AWS Free Tier?

There is no AWS Free Tier offering for Aurora at this time. However, Aurora durably stores your data across three Availability Zones in a Region and charges for only one copy of data. You are not charged for backups of up to 100% of the size of your database cluster. You are also not charged for snapshots during the backup retention period that you've configured for your database cluster.

Aurora replicates my data across three Availability Zones. Does that mean that my effective storage price will be three times what is shown on the pricing page?

No, Aurora replication is bundled into the price. You are charged based on the storage your database consumes at the database layer, not the storage consumed in the virtualized storage layer of Aurora.

What are I/O operations in Aurora and how are they calculated?

I/O operations are performed by the Aurora database engine against its SSD-based virtualized storage layer. Every database page read operation counts as one I/O.

The Aurora database engine issues reads against the storage layer to fetch database pages not present in memory in the cache:

- If your query traffic can be totally served from memory or the cache, you will not be charged for retrieving any data pages from memory.
- If your query traffic cannot be served entirely from memory, you will be charged for any data pages that need to be retrieved from storage.

Each database page is 16 KB in Amazon Aurora MySQL-Compatible Edition and 8 KB in Aurora PostgreSQL-Compatible Edition.

[Aurora was designed to remove unnecessary I/O operations](#) to reduce costs and ensure resources are available for serving read/write traffic. Write I/O operations are only consumed when persisting redo log records in Aurora MySQL-Compatible Edition or write ahead log records in Aurora PostgreSQL-Compatible Edition to the storage layer for the purpose of making writes durable.

Write I/O operations are counted in 4 KB units. For example, a log record that is 1,024 bytes counts as one write I/O operation. However, if the log record is larger than 4 KB, more than one write I/O operation is needed to persist it.

Concurrent write operations whose log records are less than 4 KB might be batched together by the Aurora database engine in order to optimize I/O consumption. Unlike traditional database engines, Aurora never flushes dirty data pages to storage.

You can see how many I/O requests your Aurora instance is consuming by checking the AWS Management Console. To find your I/O consumption, go to the Amazon RDS section of the console, look at your list of instances, select your Aurora instances, then look for the “VolumeReadIOPs” and “VolumeWriteIOPs” metrics in the monitoring section.

For more information on the pricing of I/O operations, visit the [Aurora pricing page](#). You are charged for read and write I/O operations when you configure your database clusters to the Aurora Standard configuration. You are not charged for read and write I/O operations when you configure your database clusters to Amazon Aurora I/O-Optimized.

What is Aurora Standard and Aurora I/O-Optimized?

Aurora offers you the flexibility to optimize your database spend by choosing between two configuration options based on your price-performance and price-predictability needs. The two configuration options are Aurora Standard and Aurora I/O-Optimized. Neither option requires upfront I/O or storage provisioning and both can scale I/O operations to support your most demanding applications.

Aurora Standard is a database cluster configuration that offers cost-effective pricing for the vast majority of applications with low to moderate I/O usage. With Aurora Standard, you pay for database instances, storage, and pay-per-request I/O.

Aurora I/O-Optimized is a database cluster configuration that delivers improved price performance for I/O-intensive applications such as payment processing systems, ecommerce systems, and financial applications. Also, if your I/O spend exceeds 25% of your total Aurora database spend, you can save up to 40% on costs for I/O-intensive workloads with Aurora I/O-Optimized. Aurora I/O-Optimized offers predictable pricing for all applications as there are no charges for read and write I/O operations, making this configuration ideal for workloads with high I/O variability.

When should I use Aurora I/O-Optimized?

Aurora I/O-Optimized is the ideal choice when you need predictable costs for any application. It delivers improved price performance for I/O-intensive applications, which require a high write throughput or run analytical queries processing large amounts of data. For customers with an I/O spend that exceeds 25% of their Aurora bill, you can save up to 40% on costs for I/O-intensive workloads with Aurora I/O-Optimized.

How do I migrate my existing database cluster to use Aurora I/O-Optimized?

You can use the one-click experience available in the AWS Management Console to change the storage type of your existing database clusters to be Aurora I/O-Optimized. You can also invoke the AWS Command Line Interface (AWS CLI) or AWS SDK to make this change.

Can I switch back and forth between Aurora I/O-Optimized and Aurora Standard configuration?

You can switch your existing database clusters once every 30 days to Aurora I/O-Optimized. You can switch back to Aurora Standard at any time.

Does Aurora I/O-Optimized work with Reserved Instances?

Yes, Aurora I/O-Optimized works with existing Aurora Reserved Instances. Aurora automatically accounts for the price difference between Aurora Standard and Aurora I/O-Optimized with Reserved Instances. With Reserved Instance discounts with Aurora I/O-Optimized, you can gain even more savings on your I/O spend.

Does the price of backtrack, snapshot, export, or continuous backup change with Aurora I/O-Optimized?

There are no changes to the price of backtrack, snapshot, export, or continuous backup with Aurora I/O-Optimized.

Do I continue paying for the I/O operations required for replicating data across Regions with Aurora Global Database with Aurora I/O-Optimized?

Yes, the charges for the I/O operations required to replicate data across Regions continue to apply. Aurora I/O-Optimized does not charge for read and write I/O operations, which is different from data replication.

Hardware and scaling

What are the minimum and maximum storage limits of an Amazon Aurora database?

The minimum storage is 10 GB. Based on your database usage, your Amazon Aurora storage will automatically grow, up to 128 TiB, in 10 GB increments with no impact to database performance. [There is no need to provision storage in advance.](#)

How do I scale the compute resources associated with my Amazon Aurora DB Instance?

There are two ways to scale the compute resources associated with my Amazon Aurora DB Instance – via Aurora Serverless and via manual adjustment.

You can use Aurora Serverless, an on-demand, autoscaling configuration for Amazon Aurora to scale database compute resources based on application demand. It enables you to run your database in the cloud without worrying about database capacity management. You can specify the desired database capacity range and your database will scale based on your application's needs. Read more in the [Aurora Serverless User Guide](#).

You can also [manually scale](#) your compute resources associated with your database by selecting the desired DB instance type in the AWS Management Console. Your requested change will be applied during your specified maintenance window or you can use the "Apply Immediately" flag to change the DB instance type immediately.

Both of these options will have an availability impact for a few minutes as the scaling operation is performed. Note that any other pending system changes will also be applied.

Backup and restore

How do I enable backups for my DB Instance?

Automated continuous [backups](#) are always enabled on Amazon Aurora DB Instances. Backups do not impact database performance.

Can I take DB Snapshots and keep them around as long as I want?

Yes, and there is no performance impact when taking snapshots. Note that restoring data from [DB Snapshots](#) requires the creation of a new DB Instance.

If my database fails, what is my recovery path?

Amazon Aurora automatically makes your data durable across three Availability Zones (AZs) in a Region and will automatically attempt to recover your database in a healthy AZ with no data loss. In the unlikely event your data is unavailable within Amazon Aurora storage, you can restore from a DB Snapshot or perform a point-in-time restore operation to a new instance. Note that the latest restorable time for a point-in-time restore operation can be up to five minutes in the past.

What happens to my automated backups and DB Snapshots if I delete my DB Instance?

You can choose to create a final DB Snapshot when deleting your DB Instance. If you do, you can use this DB Snapshot to restore the deleted DB Instance at a later date. Amazon Aurora retains this final user-created DB Snapshot along with all other manually created DB Snapshots after the DB Instance is deleted. Only DB Snapshots are retained after the DB Instance is deleted (i.e., automated backups created for point-in-time restore are not kept).

Can I share my snapshots with another AWS account?

Yes. Aurora gives you the ability to create snapshots of your databases, which you can use later to restore a database. You can share a snapshot with a different AWS account, and the owner of the recipient account can use your snapshot to restore a DB that contains your data. You can even choose to make your snapshots public – that is, anybody can restore a DB containing your (public) data.

You can use this feature to share data between your various environments (production, dev/test, staging, etc.) that have different AWS accounts, as well as keep backups of all your data secure in a separate account in case your main AWS account is ever compromised.

Will I be billed for shared snapshots?

There is no charge for sharing snapshots between accounts. However, you may be charged for the snapshots themselves, as well as any databases you restore from shared snapshots. Learn more about [Aurora pricing](#).

Can I automatically share snapshots?

We do not support automatic sharing of DB snapshots. To share a snapshot, you must manually create a copy of the snapshot, and then share the copy.

How many accounts can I share snapshots with?

You may share manual snapshots with up to 20 AWS account IDs. If you want to share the snapshot with more than 20 accounts, you can either share the snapshot as public, or contact support for increasing your quota.

In which regions can I share my Aurora snapshots?

You can share your Aurora snapshots within each AWS region where Aurora is available.

Can I share my Aurora snapshots across different regions?

No. Your shared Aurora snapshots will only be accessible by accounts in the same region as the account that shares them.

Can I share an encrypted Aurora snapshot?

Yes, you can share encrypted Aurora snapshots.

High availability and replication

How does Amazon Aurora improve my database's fault tolerance to disk failures?

Amazon Aurora automatically divides your database volume into 10 GB segments spread across many disks. Each 10 GB chunk of your database volume is replicated six ways, across three AZs. Amazon Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.

[Amazon Aurora storage is also self-healing](#). Data blocks and disks are continuously scanned for errors and repaired automatically.

How does Aurora improve recovery time after a database crash?

Unlike other databases, after a database crash Amazon Aurora does not need to replay the redo log from the last database checkpoint (typically five minutes) and confirm that all changes have been applied before making the database available for operations. This reduces database restart times to less than 60 seconds in most cases.

Amazon Aurora moves the buffer cache out of the database process and makes it available immediately at restart time. This prevents you from having to throttle access until the cache is repopulated to avoid brownouts.

What kind of replicas does Aurora support?

Amazon Aurora MySQL-Compatible Edition and Amazon Aurora PostgreSQL-Compatible Edition support [Amazon Aurora replicas](#), which share the same underlying volume as the primary instance in the same AWS region. Updates made by the primary are visible to all Amazon Aurora Replicas.

With Amazon Aurora MySQL-Compatible Edition, you can also create cross-region MySQL Read Replicas based on MySQL's binlog-based replication engine. In MySQL [Read Replicas](#), data from your primary instance is replayed on your replica as transactions. For most use cases, including read scaling and high availability, we recommend using Amazon Aurora Replicas.

You have the flexibility to mix and match these two replica types based on your application needs:

Feature	Amazon Aurora Replicas	MySQL Replicas
Number of replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High

Replica location	In-region	Cross-region
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes

You have two additional replication options in addition to the ones listed above. You can use [Amazon Global Database](#) for much faster physical replication between Aurora clusters in different regions. And for replication between Aurora and non-Aurora MySQL-Compatible Edition databases (even outside of AWS), you can set up your own, self-managed binlog replication.

Can I have cross-region replicas with Amazon Aurora?

Yes, you can set up cross-region Aurora replicas using either physical or logical replication. Physical replication, called [Amazon Aurora Global Database](#), uses dedicated infrastructure that leaves your databases entirely available to serve your application, and can replicate up to five secondary regions with typical latency of under a second. It's available for both Aurora MySQL-Compatible Edition and Aurora PostgreSQL-Compatible Edition.

For low-latency global reads and disaster recovery, we recommend using Amazon Aurora Global Database.

Aurora supports native logical replication in each database engine (binlog for MySQL and PostgreSQL replication slots for PostgreSQL), so you can replicate to Aurora and non-Aurora databases, even across Regions.

Aurora MySQL-Compatible Edition also offers an easy-to-use logical cross-region read replica feature that supports up to five secondary AWS regions. It is based on single threaded MySQL binlog replication, so the replication lag will be influenced by the change/apply rate and delays in network communication between the specific regions selected.

Can I create Aurora Replicas on the cross-region replica cluster?

Yes, you can add up to 15 Aurora Replicas on each cross-region cluster, and they will share the same underlying storage as the cross-region replica. A cross-region replica acts as the primary on

the cluster and the Aurora Replicas on the cluster will typically lag behind the primary by tens of milliseconds.

Can I fail over my application from my current primary to the cross-region replica?

Yes, you can promote your cross-region replica to be the new primary from the Amazon RDS console. For logical (binlog) replication, the promotion process typically takes a few minutes depending on your workload. The cross-region replication will stop once you initiate the promotion process.

With Amazon Aurora Global Database, you can promote a secondary region to take full read/write workloads in under a minute.

Can I prioritize certain replicas as failover targets over others?

Yes. You can assign a promotion priority tier to each instance on your cluster. When the primary instance fails, Amazon RDS will promote the replica with the highest priority to primary. If two or more Aurora Replicas share the same priority, then Amazon RDS promotes the replica that is largest in size. If two or more Aurora Replicas share the same priority and size, then Amazon RDS promotes an arbitrary replica in the same promotion tier.

For more information on failover logic, read the [Amazon Aurora User Guide](#).

Can I modify priority tiers for instances after they have been created?

Yes, you can modify the priority tier for an instance at any time. Simply modifying priority tiers will not trigger a failover.

Can I prevent certain replicas from being promoted to the primary instance?

You can assign lower priority tiers to replicas that you don't want promoted to the primary instance. However, if the higher priority replicas on the cluster are unhealthy or unavailable for some reason, then Amazon RDS will promote the lower priority replica.

How can I improve upon the availability of a single Amazon Aurora database?

You can add Amazon Aurora Replicas. Aurora Replicas in the same AWS Region share the same underlying storage as the primary instance. Any Aurora Replica can be promoted to primary without any data loss, and therefore can be used to enhance fault tolerance in the event of a primary DB Instance failure.

To increase database availability, simply create one to 15 replicas, in any of three AZs, and Amazon RDS will automatically include them in failover primary selection in the event of a database outage. You can use Amazon Aurora Global Database if you want your database to

span multiple AWS Regions. This will replicate your data with no impact on database performance and provide disaster recovery from region-wide outages.

What happens during failover and how long does it take?

Failover is handled automatically by Amazon Aurora so your applications can resume database operations as quickly as possible without manual administrative intervention.

- If you have an Aurora Replica in the same or a different AZ when failing over, Aurora flips the canonical name record (CNAME) for your DB Instance to point at the healthy replica, which is promoted to become the new primary. Start-to-finish, failover typically completes within 30 seconds. For improved resiliency and faster failovers, consider using [Amazon RDS Proxy](#) which automatically connects to the failover DB instance while preserving application connections. Proxy makes failovers transparent to your applications and reduces failover times by up to 66%.
- If you are running Aurora Serverless v1 and the DB instance or AZ become unavailable, Aurora will automatically recreate the DB instance in a different AZ. Aurora Serverless v2 works like provisioned for failover and other high availability features. For more information, see [Aurora Serverless v2 and high availability](#).
- If you do not have an Aurora Replica (i.e., single instance) and are not running Aurora Serverless, Aurora will attempt to create a new DB Instance in the same Availability Zone as the original instance. This replacement of the original instance is done on a best-effort basis and may not succeed, for example, if there is an issue that is broadly affecting the Availability Zone.

Your application should retry database connections in the event of connection loss. Disaster recovery across regions is a manual process, where you promote a secondary region to take read/write workloads.

If I have a primary database and an Amazon Aurora Replica actively taking read traffic and a failover occurs, what happens?

Amazon Aurora will automatically detect a problem with your primary instance and trigger a failover. If you are using the Cluster Endpoint, your read/write connections will be automatically redirected to an Amazon Aurora Replica that will be promoted to primary.

In addition, the read traffic that your Aurora Replicas were serving will be briefly interrupted. If you are using the Cluster Reader Endpoint to direct your read traffic to the Aurora Replica, the read only connections will be directed to the newly promoted Aurora Replica until the old primary node is recovered as a replica.

How far behind the primary will my replicas be?

Since Amazon Aurora Replicas share the same data volume as the primary instance in the same AWS Region, there is virtually no replication lag. We typically observe lag times in the tens of milliseconds.

For cross-region replication, binlog-based logical replication lag can grow indefinitely based on change/apply rate as well as delays in network communication. However, under typical conditions, under a minute of replication lag is common. Cross-region replicas using Amazon Aurora Global Database's physical replication will have a typical lag of under a second.

Can I set up replication between my Aurora MySQL-Compatible Edition database and an external MySQL database?

Yes, you can set up binlog replication between an Aurora MySQL-Compatible Edition instance and an external MySQL database. The other database can run on Amazon RDS, or as a self-managed database on AWS, or completely outside of AWS.

If you're running Aurora MySQL-Compatible Edition 5.7, consider setting up GTID-based binlog replication. This will provide complete consistency so your replication won't miss transactions or generate conflicts, even after failover or downtime.

What is Amazon Aurora Global Database?

[Amazon Aurora Global Database](#) is a feature that allows a single Amazon Aurora database to span multiple AWS regions. It replicates your data with no impact on database performance, enables fast local reads in each Region with typical latency of less than a second, and provides disaster recovery from region-wide outages. In the unlikely event of a regional degradation or outage, a secondary region can be promoted to full read/write capabilities in less than one minute. This feature is available for both Aurora MySQL-Compatible Edition and Aurora PostgreSQL-Compatible Edition.

How do I create an Amazon Aurora Global Database?

You can create an Aurora Global Database with just a few clicks in the Amazon RDS console. Alternatively, you can use the AWS Software Development Kit (SDK) or AWS Command-Line Interface (CLI). You need to provision at least one instance per region in your Amazon Aurora Global Database.

How many secondary regions can an Amazon Aurora Global Database have?

You can create up to five secondary regions for an Amazon Aurora Global Database.

If I use Amazon Aurora Global Database, can I also use logical replication (binlog) on the primary database?

Yes. If your goal is to analyze database activity, consider using Aurora advanced auditing, general logs, and slow query logs instead, to avoid impacting the performance of your database.

Will Aurora automatically fail over to a secondary region of an Amazon Aurora Global Database?

No. If your primary region becomes unavailable, you can manually remove a secondary region from an Amazon Aurora Global Database and promote it to take full reads and writes. You will also need to point your application to the newly promoted region.

Security

Can I use Amazon Aurora in Amazon Virtual Private Cloud (Amazon VPC)?

Yes, all Amazon Aurora DB Instances must be created in a VPC. With Amazon VPC, you can define a virtual network topology that closely resembles a traditional network you might operate in your own datacenter. This gives you complete control over who can access your Amazon Aurora databases.

Does Amazon Aurora encrypt my data in transit and at rest?

Yes. Amazon Aurora uses SSL (AES-256) to secure the connection between the database instance and the application. Amazon Aurora allows you to encrypt your databases using keys you manage through [AWS Key Management Service](#) (AWS KMS).

On a database instance running with Amazon Aurora encryption, data stored at rest in the underlying storage is encrypted, as are its automated backups, snapshots, and replicas in the same cluster. Encryption and decryption are handled seamlessly. For more information about the use of AWS KMS with Amazon Aurora, see the [Amazon RDS User's Guide](#).

Can I encrypt an existing unencrypted database?

Currently, encrypting an existing unencrypted Aurora instance is not supported. To use Amazon Aurora encryption for an existing unencrypted database, create a new DB Instance with encryption enabled and migrate your data into it.

How do I access my Amazon Aurora database?

Aurora databases must be accessed through the database port entered on database creation. This provides an additional layer of security for your data. Step-by-step instructions on how to connect to your Amazon Aurora database are provided in the [Amazon Aurora Connectivity Guide](#).

Can I use Amazon Aurora with applications that require HIPAA compliance?

Yes, the MySQL- and PostgreSQL-compatible editions of Aurora are [HIPAA-eligible](#). You can use them to build HIPAA-compliant applications and store healthcare-related information, including protected health information (PHI) under an executed Business Associate Addendum (BAA) with AWS. If you have already entered into a BAA with AWS, no further action is necessary to begin using these services in the account(s) covered by your BAA. For more information about using AWS to build compliant applications, see [Healthcare Providers](#).

Where can I access a list of Common Vulnerabilities and Exposures (CVE) entries for publicly known cybersecurity vulnerabilities for Amazon Aurora releases?

You can currently find a list of CVEs at [Amazon Aurora Security Updates](#).

How can I detect security threats to my Aurora database?

Aurora is integrated with [Amazon GuardDuty](#) to help you identify potential threats to data stored in Aurora databases. GuardDuty RDS Protection profiles and monitors login activity and new databases in your account, and uses tailored ML models to detect suspicious logins to Aurora databases. For more information, see [Monitoring threats with GuardDuty RDS Protection](#) and the [GuardDuty RDS Protection User Guide](#).

Serverless

What is Amazon Aurora Serverless?

Aurora Serverless is an on-demand, auto-scaling configuration for [Amazon Aurora](#). With Aurora Serverless, you can run your database in the cloud without managing database capacity. Manually managing database capacity can be time consuming and lead to inefficient use of database resources. With Aurora Serverless, you create a database, specify the desired database capacity range, and connect your application. Aurora automatically adjusts the capacity within the range specified based on your application's needs.

You pay on a per-second basis for the database capacity you use when the database is active. Learn more about [Aurora Serverless](#) and get started in a few steps in the [Amazon RDS Management Console](#).

What is the difference between Aurora Serverless v2 and v1?

[Aurora Serverless v2](#) supports every type of database workload, from development and test environments, websites, and applications that have infrequent, intermittent, or unpredictable workloads to the most demanding, business critical applications that require high scale and high availability. It scales in place by adding more CPU and memory without having to failover the database to a larger or smaller database instance. As a result, it can scale even when there are long running transactions, table locks, and more.

In addition, it scales database capacity in increments as small as 0.5 Aurora Capacity Units (ACUs) so your database capacity closely matches your application's needs.

[Aurora Serverless v1](#) is a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads. It automatically starts up, scales compute capacity to match your application's usage, and shuts down when it's not in use. Visit the [Aurora User Guide](#) to learn more.

Which Aurora features does Aurora Serverless v2 support?

[Aurora Serverless v2](#) supports all features of provisioned Aurora, including [read replica](#), [Multi-AZ configuration](#), [Aurora Global Database](#), [RDS Proxy](#), and [Performance Insights](#).

Can I start using Aurora Serverless v2 with provisioned instances in my existing Aurora DB cluster?

Yes, you can start using Aurora Serverless v2 to manage database compute capacity in your existing Aurora DB cluster. A cluster containing both provisioned instances as well as Aurora Serverless v2 is referred to as a mixed-configuration cluster. You can choose to have any combination of provisioned instances and Aurora Serverless v2 in your cluster.

To test Aurora Serverless v2, you add a reader to your Aurora DB cluster and select Serverless v2 as the instance type. Once the reader is created and available, you can start using it for read-only workloads. Once you confirm that the reader is working as expected, you can initiate a failover to start using Aurora Serverless v2 for both reads and writes. This option provides a minimal downtime experience to get started with Aurora Serverless v2.

Can I migrate from Aurora Serverless v1 to Aurora Serverless v2?

Yes, you can migrate from Aurora Serverless v1 to Aurora Serverless v2. Refer to the [Aurora User Guide](#) to learn more.

Which versions of Amazon Aurora are supported for Aurora Serverless?

[Aurora Serverless v1 compatibility information can be seen here.](#) [Aurora Serverless v2 compatibility information can be seen here.](#)

Can I migrate an existing Aurora DB cluster to Aurora Serverless?

Yes, you can restore a snapshot taken from an existing Aurora provisioned cluster into an Aurora Serverless DB Cluster and the other way around.

How do I connect to an Aurora Serverless DB cluster?

You access an Aurora Serverless DB cluster from within a client application running in the same VPC. You can't give a public IP address to an Aurora Serverless DB.

Can I explicitly set the capacity of an Aurora Serverless cluster?

While Aurora Serverless automatically scales based on the active database workload, in some cases, capacity might not scale fast enough to meet a sudden workload change, such as a large number of new transactions. In these cases, you can set the capacity explicitly to a specific value with the AWS Management Console, the AWS CLI, or the Amazon RDS API.

Why isn't my Aurora Serverless DB Cluster automatically scaling?

Once a scaling operation is initiated, Aurora Serverless attempts to find a scaling point, which is a point in time at which the database can safely complete scaling. Aurora Serverless might not be able to find a scaling point if you have long-running queries or transactions in progress, or temporary tables or table locks in use.

How am I billed for Aurora Serverless?

In Aurora Serverless, database capacity is measured in ACUs. You pay a flat rate per second of ACU usage. Compute costs for running your workloads on Aurora Serverless will depend on the database cluster configuration that you choose: Aurora Standard or Aurora I/O-Optimized. Visit the [Aurora pricing page](#) for information about pricing and Regional availability.

Parallel Query

What is Amazon Aurora Parallel Query?

[Amazon Aurora Parallel Query](#) refers to the ability to push down and distribute the computational load of a single query across thousands of CPUs in Aurora's storage layer. Without Parallel Query, a query issued against an Amazon Aurora database would be executed wholly within one instance of the database cluster; this would be similar to how most databases operate.

What's the target use case?

Parallel Query is a good fit for analytical workloads requiring fresh data and good query performance, even on large tables. Workloads of this type are often operational in nature.

What benefits does Parallel Query provide?

Parallel Query results in faster performance, speeding up analytical queries by up to two orders of magnitude. It also delivers operational simplicity and data freshness as you can issue a query directly over the current transactional data in your Aurora cluster. And, Parallel Query enables

transactional and analytical workloads on the same database by allowing Aurora to maintain high transaction throughput alongside concurrent analytical queries.

What specific queries improve under Parallel Query?

Most queries over large data sets that are not already in the buffer pool can expect to benefit. The initial version of Parallel Query can push down and scale out of the processing of more than 200 SQL functions, equijoins, and projections.

What performance improvement can I expect?

The improvement to a specific query's performance depends on how much of the query plan can be pushed down to the Aurora storage layer. Customers have reported more than an order of magnitude improvement to query latency.

Is there any chance that performance will be slower?

Yes, but we expect such cases to be rare.

What changes do I need to make to my query to take advantage of Parallel Query?

Changes in query syntax are not required. The query optimizer will automatically decide whether to use Parallel Query for your specific query. To check if a query is using Parallel Query, you can view the query execution plan by running the EXPLAIN command. If you wish to bypass the heuristics and force Parallel Query for test purposes, use the `aurora_pq_force` session variable.

How do I turn Parallel Query feature on or off?

Parallel Query can be enabled and disabled dynamically at both the global and session level using the `aurora_pq` parameter.

Are there any additional charges associated with using Parallel Query?

No. You aren't charged for anything other than what you already pay for instances, I/O, and storage.

Since Parallel Query reduces I/O, will turning it on reduce my Aurora IO charges?

No, Parallel Query I/O costs for your query are metered at the storage layer, and will be the same or larger with Parallel Query turned on. Your benefit is the improvement in query performance.

There are two reasons for potentially higher I/O costs with Parallel Query. First, even if some of the data in a table is in the buffer pool, Parallel Query requires all data to be scanned at the storage layer, incurring I/O. Second, a side effect of avoiding contention in the buffer pool is that running a Parallel Query does not warm up the buffer pool. As a result, consecutive runs of the same Parallel Query will incur the full I/O cost.

Learn more about [Parallel Query in the Documentation](#).

Is Parallel Query available with all instance types?

No. At this time, you can use Parallel Query with instances in the R* instance family.

What versions of Amazon Aurora support Parallel Query?

Parallel Query is available for the MySQL 5.7 and MySQL 8.0 compatible version of Amazon Aurora.

Is Parallel Query compatible with all other Aurora features?

Parallel Query is compatible with Aurora Serverless v2 and Backtrack.

If Parallel Query speeds up queries with only rare performance losses, should I simply turn it on all the time?

No. While we expect Parallel Query to improve query latency in most cases, you may incur higher I/O costs. We recommend that you thoroughly test your workload with the feature enabled and disabled. Once you're convinced that Parallel Query is the right choice, you can rely on the query optimizer to automatically decide which queries will use Parallel Query. In the rare case when the optimizer doesn't make the optimal decision, you can override the setting.

Can Aurora Parallel Query replace my data warehouse?

Aurora Parallel Query is not a data warehouse and doesn't provide the functionality typically found in such products. It's designed to speed up query performance on your relational database and is suitable for use cases such as operational analytics, when you need to perform fast analytical queries on fresh data in your database.

For an exabyte scale cloud data warehouse, please consider [Amazon Redshift](#).

Amazon DevOps Guru for RDS

What is Amazon DevOps Guru for RDS?

[Amazon DevOps Guru for RDS](#) is a new ML-powered capability for [Amazon RDS](#) (which includes Amazon Aurora) that is designed to automatically detect and diagnose database performance and operational issues, enabling you to resolve issues in minutes rather than days.

Amazon DevOps Guru for RDS is a feature of [Amazon DevOps Guru](#), which is designed to detect operational and performance issues for all Amazon RDS engines and dozens of other resource types. DevOps Guru for RDS expands the capabilities of DevOps Guru to detect, diagnose, and remediate a wide variety of database-related issues in Amazon RDS (e.g. resource over-utilization, and misbehavior of certain SQL queries).

When an issue occurs, Amazon DevOps Guru for RDS is designed to immediately notify developers and DevOps engineers and provides diagnostic information, details on the extent of the problem, and intelligent remediation recommendations to help customers quickly resolve database-related performance bottlenecks and operational issues.

Why should I use DevOps Guru for RDS?

Amazon DevOps Guru for RDS is designed to remove manual effort and shorten time (from hours and days to minutes) to detect and resolve hard to find performance bottlenecks in your relational database workload.

You can enable DevOps Guru for RDS for every Amazon Aurora database, and it will automatically detect performance issues for your workloads, send alerts to you on each issue, explain findings, and recommend actions to resolve.

DevOps Guru for RDS helps make database administration more accessible to non-experts and assists database experts so that they can manage even more databases.

How does Amazon DevOps Guru for RDS work?

Amazon DevOps Guru for RDS uses ML to analyze telemetry data collected by [Amazon RDS Performance Insights](#) (PI). DevOps Guru for RDS does not use any of your data stored in the database in its analysis. [PI measures database load](#), a metric that characterizes how an application spends time in the database and selected metrics generated by the database, such as server status variables in MySQL and pg_stat tables in PostgreSQL.

How can I get started with Amazon DevOps Guru for RDS?

To get started with DevOps Guru for RDS, ensure [Performance Insights](#) is enabled through the RDS console, and then simply enable DevOps Guru for your Amazon Aurora databases. With DevOps Guru, you can choose your analysis coverage boundary to be your entire AWS account, prescribe the specific [AWS CloudFormation](#) stacks that you want DevOps Guru to analyze, or use AWS tags to create the resource grouping you want DevOps Guru to analyze.

What types of issues can Amazon DevOps Guru for RDS detect?

Amazon DevOps Guru for RDS helps identify a wide range of performance issues that may affect application service quality, such as lock pile-ups, connection storms, SQL regressions, CPU and I/O contention, and memory issues.

How is DevOps Guru for RDS different from Amazon RDS Performance insights?

[Amazon RDS Performance Insights](#) is a database performance tuning and monitoring feature that collects and visualizes Amazon RDS database performance metrics, helping you quickly assess the load on your database, and determine when and where to take action. Amazon DevOps Guru for RDS is designed to monitor those metrics, detect when your database is experiencing performance issues, analyze the metrics, and then tell you what's wrong and what you can do about it.

Amazon RDS Blue/Green Deployments

What versions does Amazon RDS Blue/Green Deployments support?

[Amazon RDS Blue/Green Deployments](#) are available in Amazon Aurora MySQL-Compatible Edition versions 5.6. Learn more about available versions in the Aurora documentation.

What Regions does Amazon RDS Blue/Green Deployments support?

[Amazon RDS Blue/Green Deployments](#) are available in all AWS Regions (excluding AWS China Regions) and the AWS GovCloud Regions.

What is the cost of using Amazon RDS Blue/Green Deployments?

You will incur the same price for running your workloads on green instances as you do for blue instances. The cost of running on blue and green instances include our [current standard pricing](#) for db.instances, cost of storage, cost of read/write I/Os, and any enabled features, such as cost of backups and [Amazon RDS Performance Insights](#). Effectively, you are paying approximately 2x the cost of running workloads on db.instance for the lifespan of the blue-green-deployment.

For example: You have Aurora MySQL-Compatible Edition 5.7 cluster running on two r5.2xlarge db.instances, a primary writer instance and a reader instance, in us-east-1 AWS region. Each of the r5.2xlarge db.instances are configured for 40 GiB Storage and have 25 Million I/Os per month. You create a clone of the blue instance topology using [Amazon RDS Blue/Green Deployments](#), run it for 15 days (360 hours) and each green instance has 3 million I/O reads during that time. You then delete the blue instances after a successful switchover. The blue instances (writer and reader) cost \$849.2 for 15 days at an on-demand rate of \$1.179/hr (Instance + Storage+ I/O). The green instances (writer and reader) cost \$840.40 for 15 days at an on-demand rate of \$1.167/hr (Instance +Storage+ I/O). The total cost to you for using Blue/Green Deployments for those 15 days is \$1689.60, which is approximately 2x the cost of running blue instances for that time period.

What kind of changes can I make with Amazon RDS Blue/Green Deployments?

[Amazon RDS Blue/Green Deployments](#) help you make safer, simpler, and faster database changes, such as major or minor version upgrades, schema changes, instance scaling, engine parameter changes, and maintenance updates.

What is the “blue environment” in Amazon RDS Blue/Green Deployments? What is the “green environment”?”

In [Amazon RDS Blue/Green Deployments](#), the blue environment is your current production environment. The green environment is your staging environment that will become your new production environment after switchover.

How do switchovers work with Amazon RDS Blue/Green Deployments?

When [Amazon RDS Blue/Green Deployments](#) initiate a switchover, they block writes to both the blue and green environments, until switchover is complete. During switchover, the staging environment, or green environment, catches up with the production system, ensuring data is consistent between the staging and production environment. Once the production and staging environment are in complete sync, Blue/Green Deployments promote the staging environment as the new production environment by redirecting traffic to the newly promoted production environment. Amazon RDS Blue/Green Deployments are designed to enable writes on the green environment after switchover is complete, ensuring zero data loss during the switchover process.

After Amazon RDS Blue/Green Deployments switches over, what happens to my old production environment?

[Amazon RDS Blue/Green Deployments](#) do not delete your old production environment. If needed, you can access it for additional validations and performance/regression testing. If you no longer need the old production environment, you can delete it. Standard billing charges apply on old production instances until you delete them.

What do Amazon RDS Blue/Green Deployments switchover guardrails check for?

[Amazon RDS Blue/Green Deployments](#) switchover guardrails block writes on your blue and green environments until your green environment catches up before switching over. Blue/Green Deployments also perform health checks of your primary and replicas in your blue and green environments. They also perform replication health checks, for example, to see if replication has stopped or if there are errors. They detect long running transactions between your blue and green environments. You can specify your maximum tolerable downtime, as low as 30 seconds, and if you have an ongoing transaction that exceeds this your switchover will time out.

Do Amazon RDS Blue/Green Deployments support Amazon Aurora Global Databases?

No, [Amazon RDS Blue/Green Deployments](#) do not support Amazon Aurora Global Databases.

Can I use Amazon RDS Blue/Green Deployments to rollback changes?

No, at this time you cannot use [Amazon RDS Blue/Green Deployments](#) to rollback changes.

[Trusted Language Extensions for PostgreSQL](#)

Why should I use Trusted Language Extensions for PostgreSQL?

[Trusted Language Extensions \(TLE\) for PostgreSQL](#) enables developers to build high performance PostgreSQL extensions and run them safely on [Amazon Aurora](#). In doing so, TLE improves your time to market and removes the burden placed on database administrators to certify custom and third-party code for use in production database workloads. You can move forward as soon as you decide an extension meets your needs. With TLE, independent software vendors (ISVs) can provide new PostgreSQL extensions to customers running on Aurora.

What are traditional risks of running extensions in PostgreSQL and how does TLE for PostgreSQL mitigate those risks?

PostgreSQL extensions are executed in the same process space for high performance. However, extensions might have software defects that can crash the database.

[TLE for PostgreSQL](#) offers multiple layers of protection to mitigate this risk. TLE is designed to limit access to system resources. The **rds_superuser** role can determine who is permitted to install specific extensions. However, these changes can only be made through the TLE API. TLE is designed to limit the impact of an extension defect to a single database connection. In addition to these safeguards, TLE is designed to provide DBAs in the **rds_superuser** role fine-grained, online control over who can install extensions and they can create a permissions model for running them. Only users with sufficient privileges will be able to run and create using the “CREATE EXTENSION” command on a TLE extension. DBAs can also allow-list “PostgreSQL hooks” required for more sophisticated extensions that modify the database’s internal behavior and typically require elevated privilege.

How does TLE for PostgreSQL relate to/work with other AWS services?

[TLE for PostgreSQL](#) is available for [Amazon Aurora PostgreSQL-Compatible Edition](#) on versions 14.5 and higher. TLE is implemented as a PostgreSQL extension itself and you can activate it from the **rds_superuser** role similar to other extensions supported on Aurora.

In what versions of PostgreSQL can I run TLE for PostgreSQL?

You can run [TLE for PostgreSQL](#) in PostgreSQL 14.5 or higher in [Amazon Aurora](#).

In what Regions is Trusted Language Extensions for PostgreSQL available?

[TLE for PostgreSQL](#) is currently available in all AWS Regions (excluding AWS China Regions) and the AWS GovCloud Regions.

How much does it cost to run TLE?

[TLE for PostgreSQL](#) is available to [Aurora](#) customers at no additional cost.

How is TLE for PostgreSQL different from extensions available on Amazon Aurora and Amazon RDS today?

[Aurora](#) and [Amazon RDS](#) support a curated set of over [85 PostgreSQL extensions](#). AWS manages the security risks for each of these extensions under the [AWS shared responsibility model](#). The extension that implements [TLE for PostgreSQL](#) is included in this set. Extensions that you write or that you obtain from third-party sources and install in TLE are considered part of your application code. You are responsible for the security of your applications that use TLE extensions.

What are some examples of extensions I could run with TLE for PostgreSQL?

You can build developer functions, such as bitmap compression and differential privacy (such as publicly accessible statistical queries that protect privacy of individuals).

What programming languages can I use to develop TLE for PostgreSQL?

[TLE for PostgreSQL](#) currently supports JavaScript, PL/pgSQL, Perl, and SQL.

How do I deploy a TLE for PostgreSQL extension?

Once the `rds_superuser` role activates [TLE for PostgreSQL](#), you can deploy TLE extensions using the SQL `CREATE EXTENSION` command from any PostgreSQL client, such as `psql`. This is similar to how you would create a user-defined function written in a procedural language, such as PL/pgSQL or PL/Perl. You can control which users have permission to deploy TLE extensions and use specific extensions.

How do TLE for PostgreSQL extensions communicate with the PostgreSQL database?

[TLE for PostgreSQL](#) access your PostgreSQL database exclusively through the TLE API. The TLE supported trusted languages include all functions of the PostgreSQL server programming interface (SPI) and support for PostgreSQL hooks, including the check password hook.

Where can I learn more about the TLE for PostgreSQL open-source project?

You can learn more about the [TLE for PostgreSQL](#) project on the official [TLE GitHub page](#).

Generative AI

What is pgvector?

pgvector is an open-source extension for PostgreSQL supported by Amazon Aurora PostgreSQL-Compatible Edition.

What capabilities does pgvector enable for Aurora PostgreSQL?

You can use pgvector to store, search, index, and query billions of embeddings that are generated from machine learning (ML) and artificial intelligence (AI) models in your database, such as those from Amazon Bedrock (limited preview) or Amazon SageMaker. A vector embedding is a numerical representation that represents the semantic meaning of content such as text, images, and video.

With pgvector, you can query embeddings in your Aurora PostgreSQL database to perform efficient semantic similarity searches of these data types, represented as vectors, combined with other tabular data in Aurora. This enables the use of generative AI and other AI/ML systems for new types of applications such as personalized recommendations based on similar text descriptions or images, candidate match based on interview notes, customer service next best action recommendations based on successful transcripts or chat session dialogs, and more.

Read our [blog on vector database capabilities](#) and learn how to store embeddings using the pgvector extension in an Aurora PostgreSQL database, create an interactive question answering chatbot, and use the native integration between pgvector and [Aurora machine learning](#) for sentiment analysis.

Does pgvector work with Aurora machine learning?

Yes. [Aurora machine learning](#) (ML) exposes ML models as SQL functions, allowing you to use standard SQL to call ML models, pass data to them, and return predictions as query results. pgvector requires vector embeddings to be stored in the database, which requires running the ML model on source text or image data to generate embeddings and then moving the embeddings in batch into Aurora PostgreSQL.

Aurora ML can make this a real-time process enabling embeddings to be kept up-to-date in Aurora PostgreSQL by making periodic calls to [Amazon SageMaker](#) which returns the most recent embeddings from your model.

Amazon RDS Extended Support

Can I use RDS Extended Support with any minor version?

No, [Amazon RDS Extended Support](#) is only available on certain minor versions. See [Aurora User Guide](#) for details.

How can I estimate my RDS Extended Support charges?

Amazon RDS Extended Support charges depend on three factors: 1. number of vCPUs or ACUs running on the instance, 2. AWS Region, and 3. number of years past end of standard support.

To estimate your charges, [determine the number of vCPUs on your instance](#) and the appropriate calendar year pricing for your engine version. If your version is within the year 1-2 pricing and you are using provisioned instances, you will be charged **#vCPUs x Year 1 and Year 2 pricing per hour of usage for your chosen Region**. If your version is on year 3 pricing and you are using provisioned instances, you will be charged **#vCPUs x Year 3 pricing per hour of usage for your chosen Region**.

For example, if you are running a Aurora MySQL-Compatible 2 db.r5.large instance in N. Virginia on December 30, 2024, which is within the first year of RDS Extended Support, you will be charged \$0.200 per hour, or 2 vCPUs x \$0.100 per vCPU-hr.

When does Amazon Aurora start charging for RDS Extended Support?

You will begin to receive charges for Amazon RDS Extended Support the day after the Aurora MySQL-Compatible Edition's major version end of standard support date. This will be in addition to the instance, storage, backup, and/or data transfer charges incurred for the life of the instance.

For example, Aurora MySQL-Compatible 2 standard support ends on November 30, 2024. If you run an Aurora MySQL-Compatible 2 instance after November, 30, 2024 you will be charged for RDS Extended Support on that instance.

Do I have to pay for RDS Extended Support on my DB snapshots?

No, Amazon RDS Extended Support pricing does not apply to DB snapshots. However, when you restore a snapshot to a new DB instance that uses a version on RDS Extended Support, the instance will be charged RDS Extended Support pricing until you upgrade it to a standard support version or delete the instance.

When do I stop receiving charges for RDS Extended Support?

Upgrading your instance to a newer engine version that's available in standard support will prevent your instance from being charged RDS Extended Support pricing. RDS Extended

Support charges automatically stop when you shut down or delete an instance that is running a major engine version beyond its end of standard support date.

There are two different prices listed for each engine version. How do I know which of those I'm being charged?

The RDS Extended Support price you are charged depends on the engine version, AWS Region, and the number of calendar years since standard support expired for that version. You will be charged the year 1 and year 2 pricing in your chosen Region per vCPU-hr for the first two years after the end of standard support. If RDS Extended Support is offered for a third year, you will be charged the year 3 pricing in your chosen Region per vCPU-hr starting on the first day of the third year.

For example, Aurora PostgreSQL-Compatible 11 reaches end of standard support on February 29, 2024. If you are deployed in US East (Ohio), you will be charged \$0.100 per vCPU-hr between April 1, 2024 to March 31, 2026. Starting April 1, 2026, you will be charged \$0.200 per vCPU-hr.

How can I avoid being charged for RDS Extended Support?

We recommend upgrading your instance as early as possible to a major engine version that is within its standard support term. This will help avoid incurring RDS Extended Support charges.

Can I use Amazon RDS Blue/Green Deployments to migrate from a RDS Extended Support version to a standard support version?

You can use Amazon RDS Blue/Green Deployments to migrate your instances using RDS Extended Support, so long as Blue/Green Deployments supports your instance's engine, Region, and major version type. Blue/Green Deployments is available for Aurora MySQL-Compatible Edition. For information on available versions, see the [Blue/Green Deployments documentation](#).

Do Reserved Instance discounts apply to RDS Extended Support?

No, RDS Extended Support charges are independent of instance charges. Therefore, Reserved Instance discounts are not applicable to RDS Extended Support charges.

Will I get charged for RDS Extended Support even if I move from RDS for MySQL 5.7 to Aurora MySQL 2 (based on MySQL 5.7)?

If you migrate from RDS for MySQL 5.7 to Aurora MySQL 2 before February 29, 2024, you will not be charged for RDS Extended Support. If you migrate after February 29, 2024 and before November 30, 2024, you will be charged for RDS Extended Support for the number of hours you were running MySQL 5.7 on Amazon RDS.

If you migrate after November 30, 2024 or use Aurora MySQL-Compatible 2 after November 30, 2024, you will also be charged for RDS Extended Support on your Aurora database. For additional details, please refer to the [Amazon Aurora](#) and [Amazon RDS documentation](#).

What happens to DB snapshots I created on a version that is no longer on standard support? Will I have to pay RDS Extended Support price for them?

No, you will not be charged RDS Extended Support pricing on DB snapshots. However, when you restore a DB snapshot to a new DB instance after end of standard support, you will be charged RDS Extended Support pricing for that instance.

For example, if you restore a DB snapshot to a new DB instance on Aurora MySQL-Compatible 2 after November 30, 2024, the instance will be charged the Aurora MySQL-Compatible 2 RDS Extended Support pricing until you upgrade it to Aurora MySQL-Compatible version 3 or newer or delete the instance.

If I create a new instance on a major version engine after it reaches end of standard support, will I be charged for RDS Extended Support?

Yes, if you create an instance or restore a DB snapshot to an instance running on a version that has reached its end of standard support date, you will be charged for RDS Extended Support pricing in addition to the instance, storage, backup, and data transfer charges.