



Disaster recovery strategy for databases on AWS

AWS Prescriptive Guidance



AWS Prescriptive Guidance: Disaster recovery strategy for databases on AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Introduction	1
Targeted business outcomes	3
Curb financial losses	3
Reduce the impact to critical business processes	3
Reduce manual coordination during a real event (automation)	3
Retain your customers	3
Enhance security	3
Increase employee productivity	4
Defining your DR strategy	5
Choosing a database service	9
Driving adoption of your DR strategy	13
Automating your DR strategy	14
Disaster event detection	14
Failover	14
Testing to achieve confidence	16
Considerations	16
Testing frequency	16
Drift detection	17
Observability	17
Next steps and resources	18
Document history	19

Disaster recovery strategy for databases on AWS

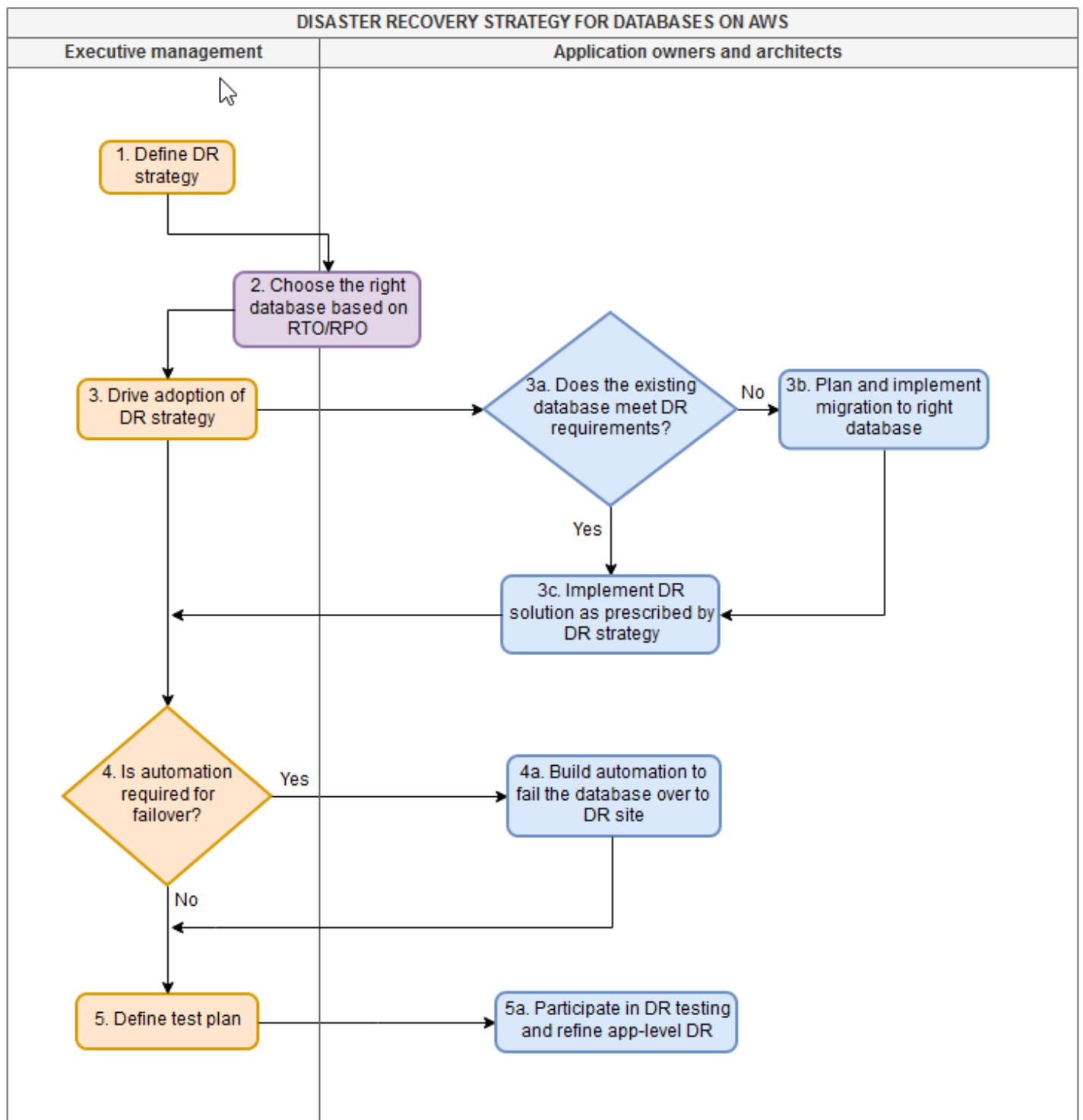
Oliver Francis, Suhas Basavaraj, Balaji Desikachari, and Jitendra Kumar, Amazon Web Services

In October 2021, Forbes Technology Council published an article titled [Why Disaster Recovery Is No Longer Optional For Today's Businesses](#). The article quotes a U.S. Federal Emergency Management Agency (FEMA) study that states that 40 percent of small and medium businesses in the United States go out of business after a disaster. Furthermore, 25 percent of the businesses that reopen close down again within a year. Given the gravity of the subject, a well-prepared business continuity plan that includes a disaster recovery (DR) strategy for IT assets is crucial to ensure the long-term growth and success of your business. Cloud computing has made disaster recovery viable even for small and medium businesses by lowering the barrier to entry. Given the risks posed by the lack of an adequate solution in the case of a disaster, a business continuity plan that includes a DR strategy should be at the top of your priority list as a business leader.

If your workloads are running on AWS, undoubtedly a bulk of your data resides in AWS databases. Whether it is data stored in a relational database system (RDBMS) such as Amazon Aurora MySQL-Compatible Edition or a NoSQL database such as Amazon DynamoDB, this data is valuable to your business. Developing a DR strategy that is specific to databases on AWS helps you understand what AWS offers in terms of disaster recovery for your databases and how you can leverage this information in your DR plan for the AWS Cloud.

The following swim lane diagram presents high-level tasks associated with defining and implementing a DR strategy. The two columns show the tasks owned by the executive management team and the tasks owned by application owners and architects. The executive management team owns decision-making. This team defines the DR strategy to be followed by the organization, drives DR adoption across the organization, decides if automation will be required to accomplish DR goals, and defines the DR test plan.

When the DR strategy is in place, the executive management team works with the application owners and architects to choose the right databases that fit the recovery time objective (RTO) and recovery point objective (RPO) expectations that are derived from the DR strategy. As the executive management team drives the adoption of the DR strategy, application owners and architects evaluate the databases that are used within their applications and migrate to the right database if the current ones don't achieve the RTO and RPO set for their application. The application owners and architects also build automation if required and participate in DR testing to refine and improve the application-level DR.



This article explains these steps at a high level for business leaders and decision makers who want to formulate a cross-Region DR strategy that helps meet their business objectives. The article assumes some technical knowledge and familiarity with disaster recovery terminology but no AWS expertise.

Targeted business outcomes

This section discusses the expected outcomes associated with defining and building a disaster recovery (DR) plan for your databases on AWS.

Curb financial losses

If you have a well-designed DR solution for databases that store your application data, you can recover from catastrophic events faster and with minimal or no data loss. This minimizes financial losses that can potentially result from your application being unavailable for prolonged periods, or, in the worst case, from permanently losing your data.

Reduce the impact to critical business processes

When you identify your tier 0 services and processes, and plan your DR strategy to recover these quickly, you will be able to reduce significant impact to these services and processes, and handle the recovery of dependent processes as well.

Reduce manual coordination during a real event (automation)

If you choose to automate your DR solution, you can reduce the manual coordination required to run the DR solution when there is an event. Automation is beneficial even when you're testing your [business continuity plan \(BCP\)](#), which specifies how you sustain standard business operations during an unplanned disruption.

Retain your customers

In a competitive market, your DR strategy affects how you earn customer loyalty and retain customers. If your organization has the ability to recover quickly from disasters, you can build confidence with your customers.

Enhance security

A well-planned cross-Region DR strategy can help restrict the impact of a ransomware attack to one AWS Region and enable you to use your data in another Region freely without losing your data.

Increase employee productivity

If your employees understand the DR process well and are comfortable with it, a DR event won't cause panic. You can follow well-defined runbooks to implement the DR solution, and your employees can make the best use of their time to get your business back to normal.

Defining your disaster recovery strategy

Depending on how critical the applications in your organization are to your business, you might decide on a uniform strategy for all applications or develop a more complex DR strategy based on the criticality of each application. Your organization might tolerate a downtime of several hours before all applications are brought up in the DR site. In this case, you can opt for a cost-effective DR strategy based on backup and restore for all databases. On the other hand, your organization's business might depend on some critical services or applications becoming available rapidly, with more aggressive RPO and RTO requirements, whereas other applications might tolerate less stringent RPO and RTO needs. In this case, you will need to assign the right DR strategy for each tier of applications and databases.

The following table describes four DR options for workloads that run in the AWS Cloud, to help you determine and define your organization's DR strategy. The RPO and RTO documented in this table are for a full stack that includes both application and database components. For more information, see [Disaster recovery options in the cloud](#) in the AWS Well-Architected Framework documentation. The next section covers RPO and RTO options that are specific to databases.

Recovery option	RPO	RTO	Infrastructure tasks in DR Region	Cost
Backup and restore	Hours	Less than 24 hours	Provision all required application resources in the DR Region and restore the database from a copied snapshot.	Low
Pilot light	Tens of minutes	Tens of minutes	Provision a copy of your application infrastructure and switch the resources in the application stack	Medium

off. Replicate your data from one Region to another. Keep the databases always on and synchronized with primary databases. Provision the resources on demand during the failover and testing event. You also need to deploy infrastructure changes and application changes to both Regions simultaneously. You can simplify this by building automation pipelines that can synchronize code and infrastructure in both primary and DR Regions.

Warm standby	Minutes	Minutes	Provision a copy of the entire application infrastructure in the DR Region, but keep the copy scaled down compared with the primary Region. The DR Region will be able to accept traffic at a smaller volume compared with the primary Region.	High
---------------------	---------	---------	--	------

Multi-site or active/active

Near zero

Zero or near zero

Provision a complete copy of your infrastructure into the DR Region. All resources in the DR Region will be equivalent to the resources in the primary Region and will be able to serve traffic at the same scale as the primary Region. Because there is no break in the traffic flow, this option doesn't require a failover task as part of your DR plan.

Higher

Choosing the right database for your RTO and RPO requirements

When you have defined the DR strategy for your organization based on the table in the previous section, choose the databases that are the right fit for your RPO and RTO requirements. By default, all AWS database services provide backup and restore functionality. Use the following table to understand more advanced DR features provided by AWS database services and map them to your DR requirements.

AWS database service	Replication method to DR Region	Possible standby classification	RTO	RPO
Amazon RDS for MySQL	Cross-Region read replica	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes. Automation can minimize delays.	Typically under 1 second
Amazon RDS for PostgreSQL	Cross-Region read replica	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes. Automation can minimize delays.	Typically under 1 second
Amazon RDS for MariaDB	Cross-Region read replica	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes. Automation can minimize delays.	Typically under 1 second
Amazon Aurora MySQL-Compatible Edition	Global database has secondary cluster(s) in a different Region	1. Hot standby with capability to support read traffic 2. Warm standby with capability to support read traffic	Typically under 1 minute.	Typically under 1 second

Amazon Aurora PostgreSQL-Compatible Edition	Global database has secondary cluster(s) in a different Region	1. Hot standby with capability to support read traffic 2. Warm standby with capability to support read traffic	Typically under 1 minute.	Typically under 1 second
Amazon DocumentDB (with MongoDB compatibility)	Global cluster has secondary cluster(s) in a different Region	1. Hot standby with capability to support read traffic 2. Warm standby with capability to support read traffic	Typically under 1 minute.	In seconds
Amazon ElastiCache (Redis OSS)	Global datastore has a secondary cluster in a different Region	1. Hot standby with capability to support read traffic 2. Warm standby with capability to support read traffic	Usually minutes. Automation can minimize delays.	In seconds
Amazon DynamoDB	DynamoDB global tables	Active/active configuration accepts write operations in secondary Region	Zero or near zero.	Sub-second

Amazon RDS for Oracle	Read replica promotion across Regions (Oracle Enterprise Edition and Active Data Guard)	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes.	Typically under 1 second
Amazon RDS for Oracle (alternative)	Mounted read replica promotion across Regions (Oracle Enterprise Edition)	1. Pilot light 2. Warm standby (doesn't allow read queries)	Usually minutes.	Typically under 1 second
Amazon RDS for SQL Server	Cross-Region read replica (Microsoft SQL Server Enterprise Edition 2016-2022)	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes. Automation can minimize delays.	Typically under 1 second
Amazon Neptune	Global database has secondary clusters in a different Region	1. Hot standby with capability to support read traffic 2. Warm standby with capability to support read traffic	Typically under 1 minute.	Typically 1 second
Amazon RDS for Db2	Cross-Region read replica (Amazon RDS for Db2 11.5.9 and 12.1.4)	1. Pilot light 2. Warm standby with capability for read traffic	Usually minutes. Automation can minimize delays.	Typically under 1 second

Amazon MemoryDB Multi-Region	Multi-Region cluster with active-active replication	Active/active configuration accepts write operations in secondary Region	Zero or near zero.	Typically under 1 second
---	---	---	-----------------------	-----------------------------

Driving adoption of your disaster recovery strategy

When your DR strategy is in place, you can drive adoption of the strategy throughout your organization. When your strategy reaches 100 percent adoption, all applications and their databases would be in a position to handle an adverse DR event and can pursue DR maturity through testing.

You can work with application owners and architects to drive adoption by communicating the DR strategy throughout your organization and asking employees to evaluate the current AWS databases used in their applications. This ensures that all currently used databases can meet the expectations set in the DR strategy. If application owners and architects determine that the database they chose for their applications cannot meet the requirements of the DR strategy, they will have to plan a migration to a suitable AWS database that can achieve the chosen RTO and RPO expectations. For example, if a critical application in your business that requires an RPO in seconds and an RTO in minutes currently uses an Amazon RDS for Oracle DB instance, Amazon RDS will not be able to meet these expectations. In this case, you could consider migrating the workload to [Amazon Aurora PostgreSQL-Compatible](#) and use a global database to meet the expectations set by your DR strategy.

Automating your disaster recovery strategy

You can optionally choose to implement full or partial automation to gain better control of disaster recovery. If you're using the backup and restore DR option, you can automate your backups by using [AWS Backup](#), which supports all Amazon RDS databases as well as DynamoDB, Amazon DocumentDB, and Amazon Neptune tables.

Disaster event detection

To shorten the recovery time, you can consider automating the detection of a Regionwide event, which can then initiate failover to the DR Region. To implement automated detection to achieve an aggressive RTO, you can build a solution based on [health checks](#). These health checks do not stop at heartbeats (which check whether the control plane and data plane modules within a network can communicate with one another), but go deeper to evaluate the interrelated nature of application components to achieve an accurate prediction. However, an automated solution can carry the risk of false alarms, which can lead to unnecessary failovers. You should exercise caution in this case, because unnecessary failovers introduce availability issues for your business. You also can build manual overrides in the workflow to confirm that the failover has been carried out. You can subscribe to the [Service Health Dashboard](#) RSS feed to stay informed of service-level disruptions. Additionally, you can use the [AWS Health Dashboard](#) (requires an AWS account) within your primary Region and account to stay aware of events that can affect your account. These can help you make an informed decision to fail over in the case of a Region-wide event.

Failover

Regardless of which DR strategy you choose, you can build custom DR automation solutions to perform the failover to the DR Region. This automation can minimize the need for manual intervention and provide greater control in testing your DR solution. You can choose from the [AWS service APIs](#), which AWS provides in multiple languages such as JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js and C++, based on your organization's preference. To build automation that uses these AWS service APIs, you should first focus on turning the database infrastructure to code in the form of AWS CloudFormation or Terraform templates. These templates can help you automate the failover of several databases and also maintain the order in which application and database components are brought back up in the DR Region.

For DR purposes, we recommend that you focus on these two goals:

- Existing [CloudFormation stacks should export](#) pertinent information about your databases, including instance names and endpoints. Your automation processes can refer to these export values within a Region and perform operations that will help with your DR operations.
- If you have resources that are in production but don't have an associated CloudFormation stack, you should focus on creating stacks for those resources. Also ensure that these stacks cover the right export values, as mentioned in the previous point.

When you have met these two goals, you can build automation solutions in the language of your organization's choice to take advantage of CloudFormation exports and automatically perform the cutover actions required in the event of a disaster. For example, if you have an ElastiCache (Redis OSS) global datastore that is deployed as a CloudFormation template, the automation code has access to the CloudFormation exports that provide details about the global datastore. In the event of a disaster, the code can automatically promote the secondary datastore to the primary datastore without any manual intervention by using the ElastiCache (Redis OSS) service APIs.

In a typical scenario, automation should be scalable for multiple databases within your organization. You can scale your automation solutions for several databases by using [AWS Step Functions](#) or [AWS Batch](#).

Testing to achieve confidence

The best DR solution for databases is one that is tested frequently and passes the following checks:

- Proper data recovery that meets the RPO expectations for each database
- Complete restoration of a functioning database within the expected RTO time frame, which allows applications to connect to the database and resume full functionality

DR testing should be part of your business strategy so that backups work when they are most needed. DR testing should also address cases where:

- The size of a database grew significantly, and your current DR strategy no longer meets the service-level agreement (SLA) for the business.
- A backup file is corrupted, which could cause problems during recovery.

What to consider when testing your DR strategy

- Have clear business continuity goals regarding RPO and RTO, and make sure that test results align with your goals.
- Create a detailed DR test plan that takes financial and human resource requirements for testing into account.
- Assign resources to document potential issues and learnings.
- Update your DR strategy based on learnings, and find the solution that supports the optimal processes and automation that works for your organization.

Testing frequency for DR solutions

There are no set recommendations for the DR test cycles, unless they are explicitly prescribed by regulations. For example, the Payment Card Industry Data Security Standards (PCI DSS) compliance audit requires organizations to test their DR plan at least once a year. (See [PCI DSS Disaster Recovery Requirements](#) on the PCI DSS website.)

Application teams can also perform continuous testing of their individual DR solutions when their application or infrastructure changes.

Drift detection

Your DR solution should also manage [drift detection](#). This will ensure that the primary Region and DR Region are at the right level of synchronization and will ensure smooth progress during testing. [AWS Config](#) provides configuration management and history tracking of configurations in your infrastructure, and can help you manage drift effectively.

Observability

Improving observability positively impacts your preparation for testing. All DR solutions move data in the primary Region to the secondary (DR) Region. You can set up alerts for replication lag and backups, or put a process in place to perform daily checks that ensure that your data was copied to the DR Region successfully.

Next steps and resources

- For more information about DR on AWS, read the white paper [Disaster Recovery of Workloads on AWS: Recovery in the Cloud](#).
- Review the [reliability pillar](#) in the AWS Well-Architected Framework.
- Explore [AWS Resilience Hub](#) to track and validate the resilience of your AWS workloads.
- Review the following additional resources:
 - [Using failover in an Amazon Aurora global database](#) (Aurora documentation)
 - [Replication across AWS Regions using global datastores](#) (Amazon ElastiCache (Redis OSS) documentation)
 - [Disaster Recovery and Amazon DocumentDB Global Clusters](#) (Amazon DocumentDB documentation)
 - [Prepare for faster disaster recovery: Deploy an Amazon Aurora global database with Terraform \(Part 1\)](#) (AWS blog post)
 - [Amazon DynamoDB global tables](#) (AWS website)
 - [Disaster Recovery on AWS, Cross-Region Replication](#) (Amazon RDS lab)
 - [Cross-Region disaster recovery of Amazon RDS for SQL Server](#) (AWS blog post)
 - [Managed disaster recovery with Amazon RDS for Oracle cross-Region automated backups – Part 1](#) (AWS blog post)

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Updated details in the comparison table	Updated the Choosing the right database for your RTO and RPO requirements section.	June 29, 2026
Updated details in the comparison table	In the Choosing the right database for your RTO and RPO requirements section, updated the information about Amazon RDS for Db2, and corrected RPO details for Amazon RDS for Oracle and Amazon RDS for Oracle (alternative).	June 11, 2025
Added Amazon RDS for Db2	In the Choosing the right database for your RTO and RPO requirements section, added information about Amazon RDS for Db2.	March 4, 2024
Updated RTO/RPO details	In the Choosing the right database for your RTO and RPO requirements section, updated the information about Amazon DocumentDB, Amazon ElastiCache (Redis OSS), and Amazon RDS for SQL Server.	April 19, 2023
Initial publication	—	October 26, 2022