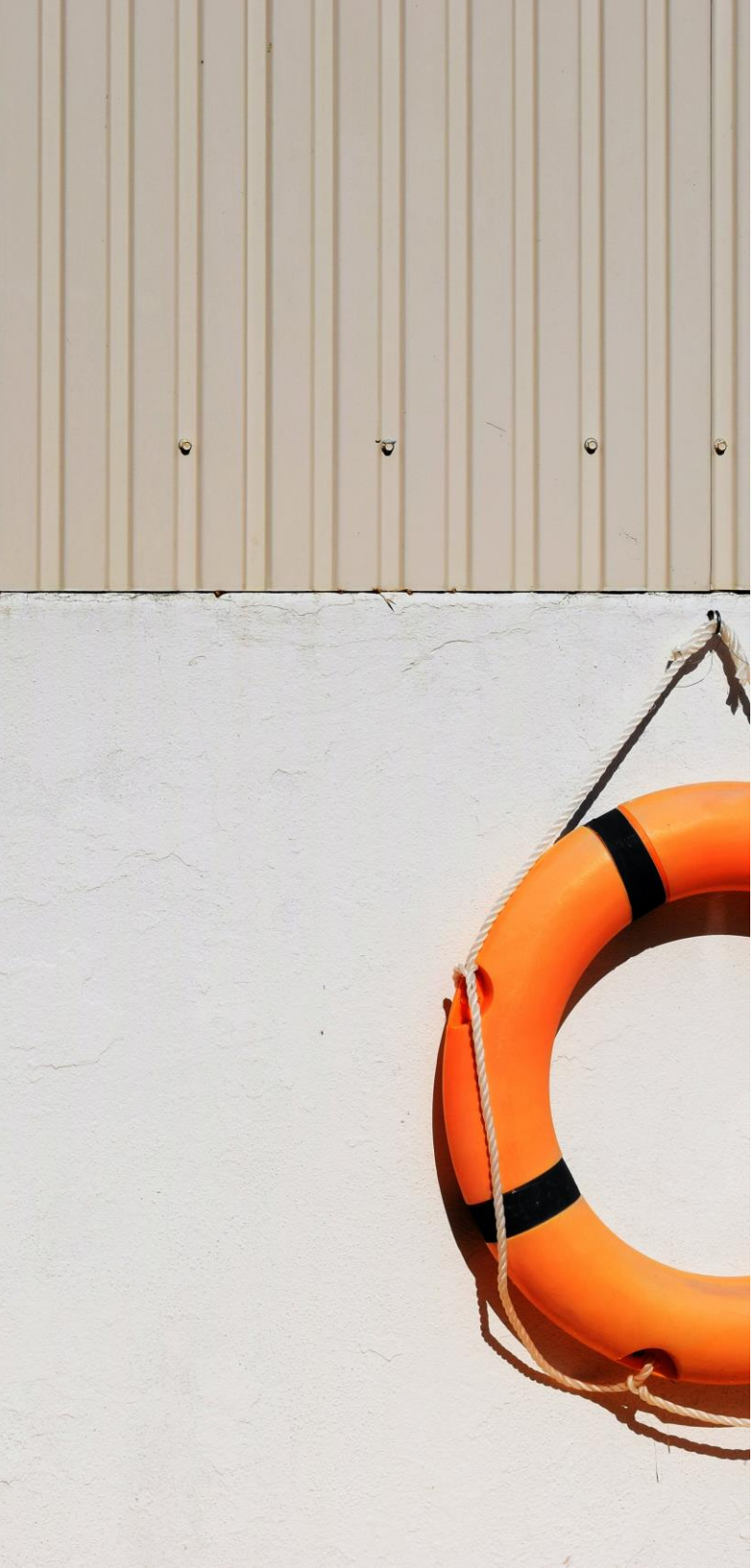
Best practices for implementing disaster recovery in the cloud

No matter how robust an organization's IT systems may seem, disruptions are inevitable. Whether caused by human error, natural disasters, or malicious cyberattacks, an event that disrupts the operation of one or more of an organization's applications can result in financial loss, reputation damage, and non-compliance.

According to IDC, in 2020 the cost of annual system downtime ranged between $1.25B to $2.5B, so it's not surprising that disaster recovery has become an essential component of an organization's business continuity strategy.

Given the high stakes, IT teams need to understand the core concepts of disaster recovery, the differences between disaster recovery and backup, and best practices for planning and implementing disaster recovery. They should also explore the benefits of using public clouds such as Amazon Web Services (AWS) as recovery sites for on-premises and cloud-based applications. In many cases, cloud-based recovery both lowers costs and improves resilience.



What is the difference between backup and disaster recovery?

Before developing a disaster recovery plan, it's important to understand what disaster recovery is, what it isn't, and how it differs from backup. Some organizations mistake backup for disaster recovery, but as they may discover after a serious IT disruption, simply having copies of data (backup) doesn't mean you can keep your business running (disaster recovery).

Backup

Backup is the process of making an extra copy (or multiple copies) of data to protect against data loss. Backup solutions keep extra copies of data locally or in a remote location or both so that if any data is lost or corrupted, it can be recovered. Local backups can be restored more quickly than remote backups, but remote backups have the advantage of increased resilience due to geographic redundancy.

Backup solutions often have relatively low total cost of ownership (TCO) as the only infrastructure needed is storage, and the performance requirements for that storage are low. Some companies even use tape-based backup because of the low cost.

Disaster recovery

Disaster recovery refers to the plan and processes for quickly reestablishing access to applications, data, and IT resources when a disruption occurs. This plan might involve switching over to a redundant set of servers and storage systems (also known as failing over to a secondary or recovery site) until your source servers are functional again. Switching to a secondary site is not done automatically, but is instead performed on the basis of an explicit decision of an authorized team.

Disaster recovery usually has a higher TCO than backup, as the secondary site needs to be maintained and tested at all times and must be advanced enough to support the functionality of the entire application in case of a disaster.

How to choose between backup and disaster recovery

Backup and disaster recovery are not mutually exclusive. Business requirements may dictate that organizations apply a combination of these solutions, depending on the resilience requirements of each application.

In terms of similarities, both backup and disaster recovery solutions maintain copies of historical data that may have changed in the source server (often referred to as “snapshots” or “point-in-time copies”). Backup solutions are able to restore a previous version of data if it was incorrectly modified or corrupted on the source server. Disaster recovery solutions are able to launch copies of applications in an operational state.

In addition, disaster recovery solutions should provide the option of launching applications from a previous point in time, which allows for successful recovery if the latest state of the source application prevents normal operation. Database corruptions, ransomware data encryption, and incorrect software configuration all fall under this category.

Backup

- Restores data and files
- Recovery objectives of hours or days
- Longer retention period

Disaster recovery

- Recovers entire application and system state
- RPO of seconds, RTO of minutes
- Change-based, continuous replication

When deciding whether an application needs a backup or disaster recovery solution, consider the following differences:

- **Purpose —** Backups work best when you need to access a lost or damaged file or object, such as an email, PowerPoint presentation, or database. Backups are also useful for long-term data archiving and data retention. If you need to restore a single piece of data, it is much easier to use a backup copy rather than recover an entire application where that single piece of data is stored.

  However, if you want your organization to quickly restore its functions after an unexpected IT disruption, disaster recovery is the right solution. With a disaster recovery solution in place, you can perform a failover to launch replicated applications to your secondary or recovery site, and your business can continue to function as normal even if the production site is unavailable.

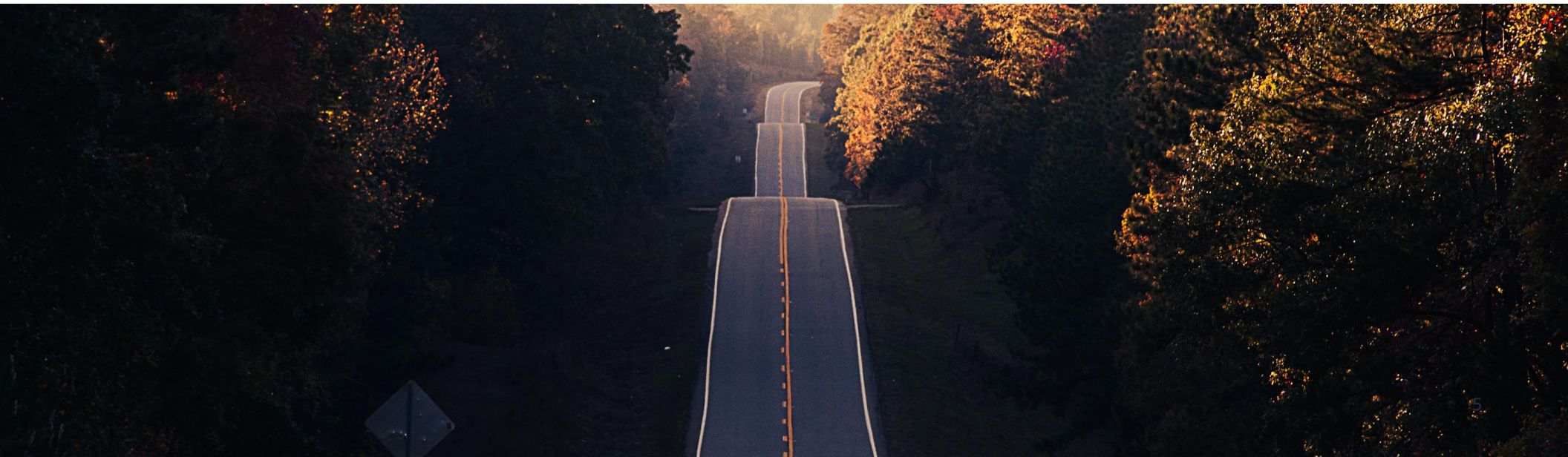
- **Speed of recovery —** Restoring data from backups usually does not help with business continuity or quick recovery. Disaster recovery, however, replicates your critical applications with the aim of quickly performing failover if necessary to assure the business continuity of the affected applications.

  Whereas disaster recovery solutions can achieve a recovery time objective (RTO) of minutes and a recovery point objective (RPO) of seconds, backup solutions are usually unable to meet these aggressive recovery objectives.

- **Resource allocation —** Backups are usually stored in a compressed state and do not need to be restored quickly. Therefore, organizations normally use low-cost, low-performance storage (often off site) for backup.

  Disaster recovery requires a separate site with operational IT infrastructure that should be ready for a possible failover at any time. This requires additional resources.

TIP:
Select a disaster recovery solution that offers point-in-time recovery—a feature that facilitates failover using earlier versions of replicated servers. In the case of ransomware or data corruption, this will help you launch an unencrypted or uncorrupted version of your applications.



Business impact and risk analysis

A properly planned and implemented disaster recovery solution helps mitigate the damage that can be caused by a disaster, including:

- **Direct and indirect financial loss —** The impact of direct financial loss is most relevant for applications that are critical for revenue-generating processes. These can include external-facing IT systems that are provided to your customers for a fee or internal IT systems that process data required for revenue generation. Indirect financial loss may occur as a result of customers switching to a competing product due to your service not being available or from the cost of additional work needed to resume normal operation after the disaster is over.

- **Reputational damage —** In addition to financial loss, downtime caused by unexpected IT disruptions can significantly harm your organization's reputation. A short recovery period aided by an effective disaster recovery solution can help avoid irreversible damage to a company's corporate image.

- **Failure to abide by compliance standards —** Multiple compliance standards, including System and Organization Controls (SOC), payment card industry (PCI), and the Health Insurance Portability and Accountability Act (HIPAA), require that a disaster recovery solution be in place. Some even add very specific requirements, such as minimal physical distance between the source site and the recovery site.

  The negative impact of a disaster, however, is not equivalent for all applications. Conducting a business impact analysis for each of your applications will help you quantify the business impact of a disruption to each application.

  When calculating the business impact, consider the consequences of application downtime on both internal and external customers—and the effect that will have on your business with regards to cost, reputation, and compliance.

The business impact of a disaster may not be constant. For example, disruption to your customer payment system is likely to have a higher impact on the business during the fourth quarter (October - December) when retail sales tend to increase.

Risk analysis for <APPLICATION NAME> per quarter for Q1, Q2, and Q3			
Disaster type	**Likelihood** (per quarter)	**Consequences**	**Risk** (likelihood x consequences)
Weather-related outage	3%	$200,000	$6,000
Power outage	1.5%	$300,000	$4,500
Ransomware attack	1%	$2,000,000	$20,000
Total			**$30,500**

Risk analysis for <APPLICATION NAME> per quarter for Q4 (includes holiday shopping season)			
Disaster type	**Likelihood** (per quarter)	**Consequences**	**Risk** (likelihood x consequences)
Weather-related outage	3%	$260,000	$7,800
Power outage	1.5%	$390,000	$5,850
Ransomware attack	1%	$2,600,000	$26,000
Total			**$39,650**

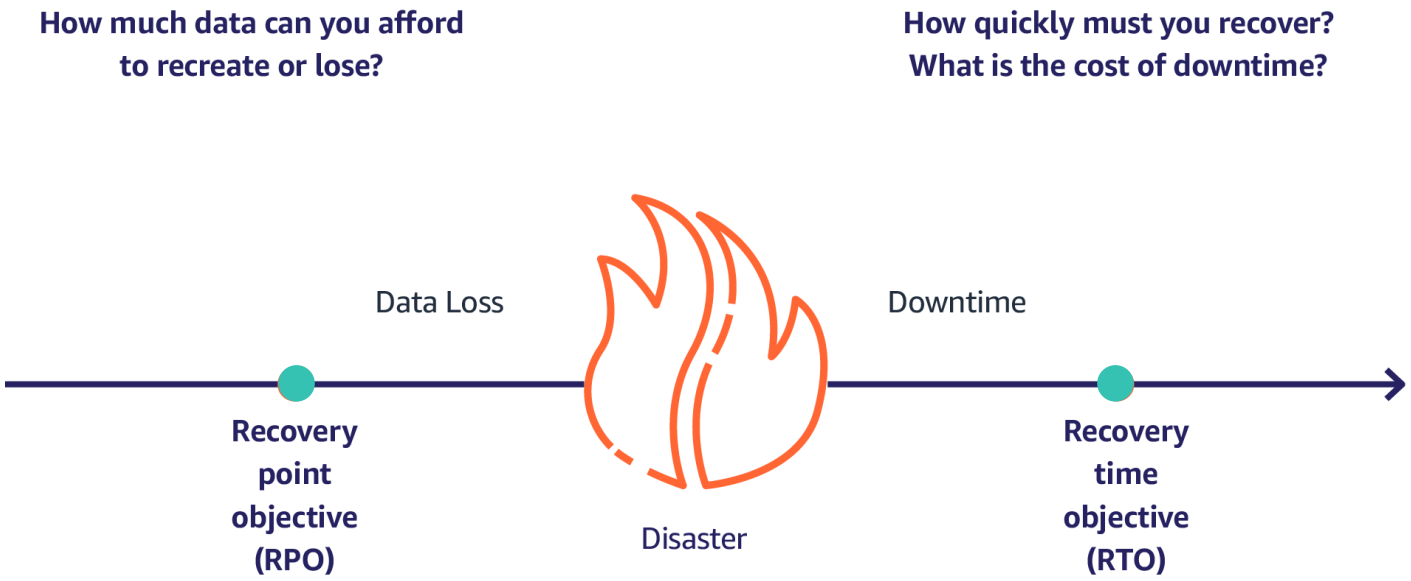
TIP:
Conduct a business impact analysis for each application to help you define your RTO and RPO in the most cost-effective manner.

Determining recovery objectives

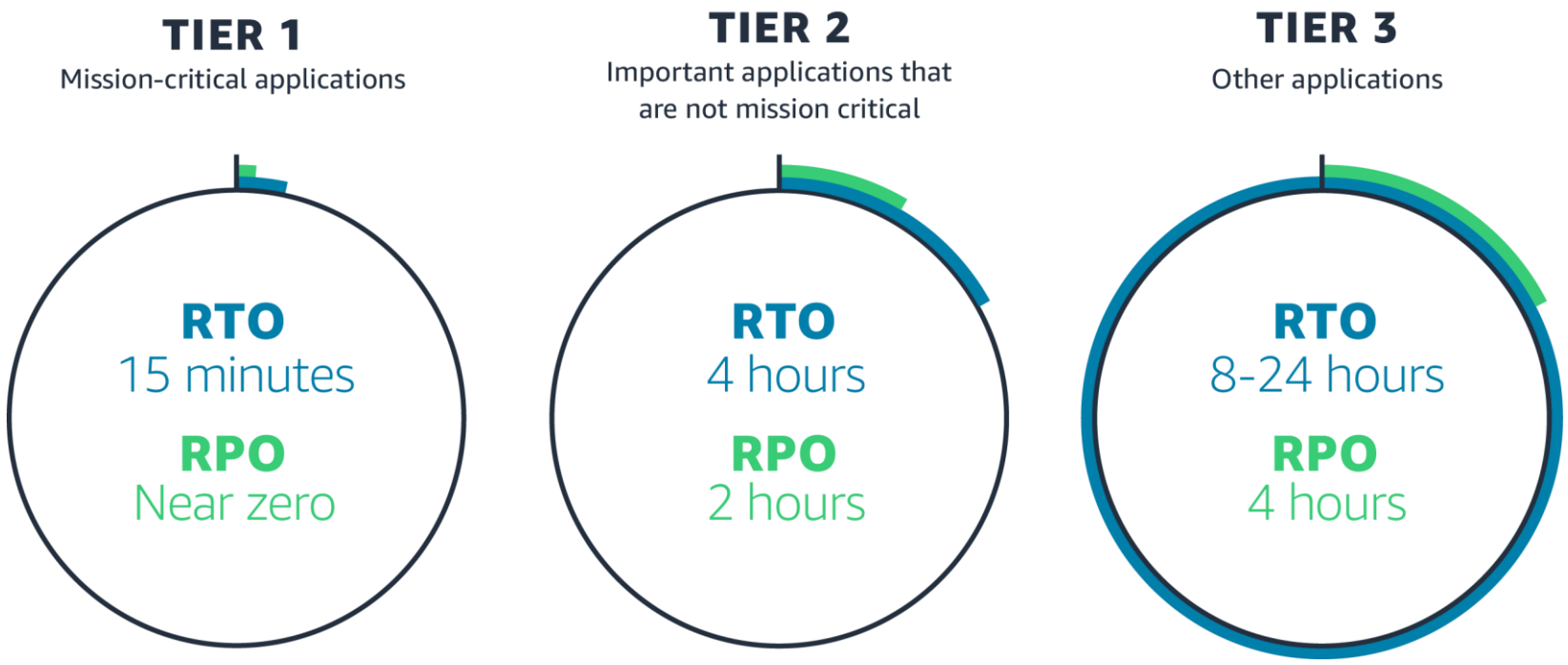
As part of disaster recovery planning, you should define a recovery time objective (RTO) and recovery point objective (RPO) for each application. Use the risk analysis to help you determine how quickly the application needs to be made available (RTO) and how much data loss can be tolerated (RPO).

RTO is the maximum acceptable delay between the interruption of an application and the restoration of its service. This objective determines what is considered an acceptable time window when an application is unavailable.

RPO is the maximum acceptable gap between the data in the recovery site and the latest data stored in the source application when the disaster strikes. This objective determines what is considered an acceptable loss of data (measured in time units) that can be caused by a disaster.



Although RTO and RPO for each application depend on many factors, such as business impact, service level agreements (SLAs), and external compliance requirements, there are some common standards. Common figures for mission-critical applications ("tier-1 applications") are an RTO of 15 minutes and a near-zero RPO. For important applications that are not mission critical ("tier-2 applications"), the RTO is typically 4 hours and the RPO is 2 hours. For all other applications ("tier-3 applications"), a typical RTO is 8 to 24 hours and RPO is 4 hours.



Use the risk analysis and recovery objectives to choose the most appropriate disaster recovery strategy and tools for each application—with the aim of meeting your RTO and RPO while reducing TCO.

Using the cloud to reduce disaster recovery costs

The disaster recovery industry has changed dramatically in recent years due to the advancement of cloud technology. Cloud providers such as AWS allow you to pay only for the resources you use, which is not the case with most on-premises data centers. The elasticity, scalability, and security benefits of AWS make it an ideal disaster recovery site that reduces disaster recovery TCO by decreasing both capital expenditures (CapEx) and ongoing IT operating expenses (OpEx):

- **Hardware —** When using AWS as your target recovery site, no hardware is needed, and you pay for a fully provisioned recovery site only when required, such as during a disaster or drill. This means no CapEx investment or unnecessary duplicate provisioning of resources.
- **Software licenses —** If you use AWS as your recovery site along with an appropriate replication tool, you can minimize the need for duplicate software licenses since there usually is no need for duplicate standby systems or standby licenses. The disaster recovery tool can keep servers continuously in sync on AWS without running most operating system or application licenses. In the event of a disaster or a test, you can launch your servers within minutes and then pay for third-party licenses as needed.
- **Disaster recovery infrastructure and services —** Whereas traditional disaster recovery solutions require duplicate compute and storage infrastructure provisioned in the recovery site, the flexibility of AWS allows you to replicate your applications into a low-cost storage area. This virtually eliminates the need to pay for expensive compute during regular disaster recovery maintenance. During a disaster or drill, you can launch fully provisioned workloads, and only then do you need to pay for more comprehensive compute resources.
- **Management and monitoring —** AWS disaster recovery solutions can provide more advanced automation than traditional solutions, which means fewer IT resources are required to maintain and launch your applications. Automated conversion minimizes the heavy lifting typically involved in converting servers from one infrastructure to another. As a result, servers can boot natively on AWS, even if they originated from non-cloud infrastructure.

Olli Salumeria saves 80% on disaster recovery by leveraging AWS

Olli Salumeria set up a disaster recovery site on AWS for its SAP ERP infrastructure. In addition to achieving an RTO of 15 minutes and an RPO of 5 minutes, Olli Salumeria lowered its disaster recovery costs by 80% because it no longer had to pay for duplicate compute resources. It pays for fully provisioned Amazon Elastic Compute Cloud (EC2) instances only in the case of an actual disaster or drill.

Setting up a recovery site on AWS

AWS Elastic Disaster Recovery is the recommended service for setting up a recovery site on AWS. It minimizes downtime and data loss with fast, reliable recovery of on-premises and cloud-based applications using affordable storage and minimal compute.

One of the benefits of Elastic Disaster Recovery is that you can recover your applications on AWS from physical infrastructure, VMware vSphere, Microsoft Hyper-V, and cloud infrastructure. You can also use it to recover Amazon EC2 instances into a different AWS Region or Availability Zone.

Elastic Disaster Recovery continuously replicates applications and databases from any supported source into a staging area subnet in your AWS account in the AWS Region you select. The staging area design reduces costs by using low-cost storage and minimal compute resources to maintain ongoing replication.

You can perform non-disruptive tests to confirm that implementation is complete. During normal operation, maintain readiness by monitoring replication and periodically performing non-disruptive recovery and failback drills.

If you need to recover applications, you can launch recovery instances on AWS within minutes, using the most up-to-date server state or a previous point in time.

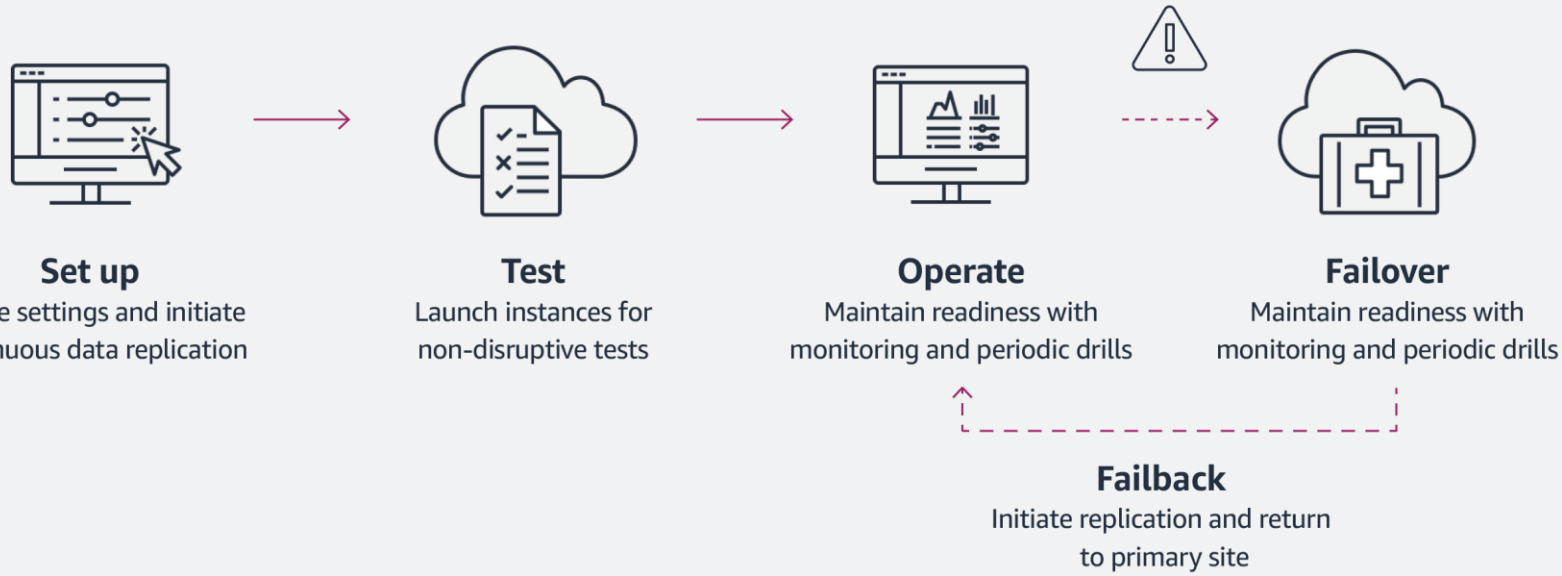
After your recovery instances are running on AWS, you can choose to keep them there, or you can initiate data replication back to your primary site when the issue is resolved.

In addition to reducing TCO, using AWS as a recovery site for your applications with Elastic Disaster Recovery provides the following benefits:

- **RTO and RPO —** Ability to achieve RTOs of minutes by launching the disaster recovery site on demand and RPOs of seconds using continuous data replication.
- **Source infrastructure support —** Supports applications running on x86 architecture (including physical and virtual).
- **Hypervisor support —** Supports any hypervisor (including physical servers, when there is no hypervisor at all).
- **Wide OS support —** Supports a large variety of Linux distributions and Windows versions.
- **Environment isolation —** Ability to set up the recovery site in an isolated environment so that drills do not impact the source site.
- **Application support —** ACID (atomicity, consistency, isolation, durability)-compliant applications are supported, including any ACID-compliant database, such as Microsoft SQL Server, Oracle Database, and SAP HANA.
- **Automation —** Ability to fully automate disaster recovery-related operations, such as failback.
- **Ease of use —** Ability to add disaster recovery capabilities to working applications with no need for redesign or re-architecture.
- **Point-in-time recovery —** Ability to launch recovery instances using uncorrupted, unencrypted versions of your applications from a previous point in time.

Implementation best practices

The process of setting up, testing, and implementing Elastic Disaster Recovery on AWS is the same for all applications. The high-level process looks like this:



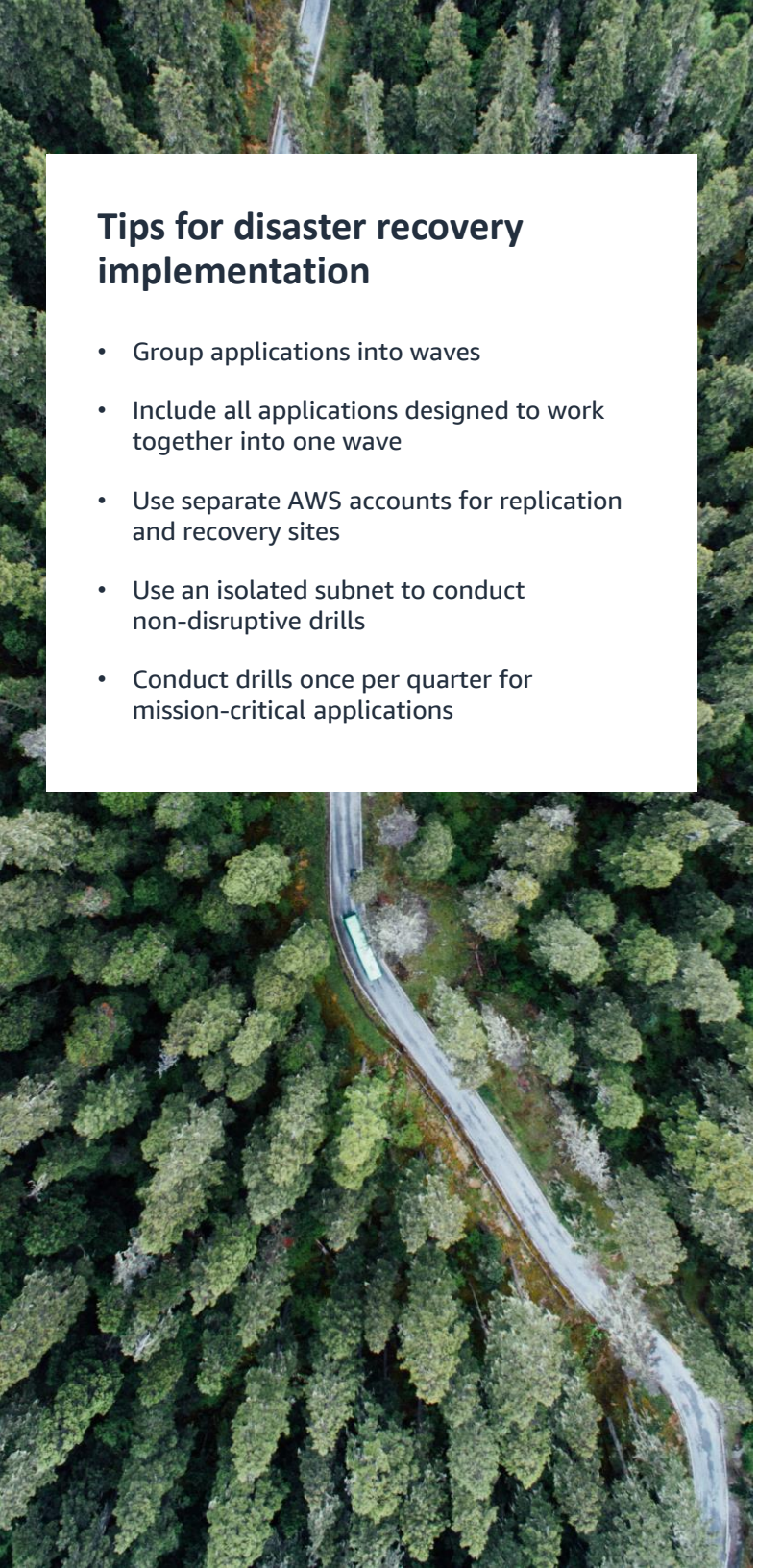
There are many resources you can turn to for in-depth, step-by-step instructions, including a proof of concept (POC) checklist and user guide, online training, and technical documentation. The following best practices will help you implement Elastic Disaster Recovery successfully.

Group your applications into waves — Each wave should include applications that are designed to work together so that every wave can be tested individually without being obstructed by dependencies on resources that are allocated to future waves. By implementing disaster recovery in waves, any issues that arise will have limited impact, and the implementation project will be easier to manage.

Make sure that the first several waves are smaller in size (10-20 servers per wave) as the implementation of these waves may take more effort and be slower due to lack of familiarity with the solutions. Once your team becomes proficient with the methodology and tools, the size of the waves can be increased according to the desired rate of progress and available resources. Large-scale disaster recovery projects usually include 100-200 servers in each wave.

Configure dedicated subnets for staging areas — Before the implementation of the first wave, the staging areas should be created for all of the waves. It's most effective to use dedicated subnets for the staging areas. This allows setting up the needed connectivity without interfering with any other processes. This also helps ensure that existing resources don't compete over private IP addresses with the resources automatically provisioned by the disaster recovery solutions.

Use multiple AWS accounts — Use different AWS accounts for the staging areas and the recovery sites (used for failover). AWS accounts have API throttling, so replicating more than 300 servers into a staging area in a single AWS account is not recommended. In these cases, multiple AWS accounts should be used.



Tips for disaster recovery implementation

- Group applications into waves
- Include all applications designed to work together into one wave
- Use separate AWS accounts for replication and recovery sites
- Use an isolated subnet to conduct non-disruptive drills
- Conduct drills once per quarter for mission-critical applications

You can fail over servers that are using different AWS accounts for staging areas into the same target AWS account (and vice versa). In general, the replication subnets (staging area) and the recovery subnets (target site) of an application don't have to be in the same AWS account. This provides flexibility when designing the AWS account architecture for disaster recovery.

Conduct an initial disaster recovery drill after setup — Once implementation of all the waves is finished, run a comprehensive disaster recovery drill to make sure nothing was missed. Finding and correcting issues is more expensive and labor intensive when done after your disaster recovery solution is up and running.

Isolate target applications before drills — Isolate the target instance subnets before performing any drill as doing so can prevent potential conflicts with the source environment. Isolating the target application allows you to continue operating the production site uninterrupted during drills, which in turn can remove risks in case the drill does not go as expected.

Perform periodic drills — After your initial drill, it's important to perform periodic drills as part of maintaining your disaster recovery solution. This helps to gain confidence that a disaster recovery solution will provide business continuity within the desired RTO and RPO for each application and that the disaster run books are accurate. The more frequently drills are performed, the higher the level of confidence.

However, every drill has costs associated with it, including workforce costs and the cost of launched resources on AWS. Therefore, every organization needs to decide the frequency of drills for each application. Industry best practices are to conduct a disaster recovery drill at least once per year and for the more business-critical applications, no less than once per quarter.

Summary

When it comes to business continuity, all applications should not be treated equally. Conducting a proper business impact analysis for each application will help you determine whether it requires a lower-cost backup solution that focuses on data retention or a more robust disaster recovery solution that can minimize downtime during an IT disruption.

Once you have determined which applications require disaster recovery and what their recovery objectives are, consider leveraging AWS to set up a flexible, cost-effective, and reliable recovery site. Moving from an on-premises recovery site to a cloud-based recovery site can reduce your disaster recovery TCO while offering additional benefits, such as automation, isolated testing environments, and point-in-time recovery.

Visit AWS Elastic Disaster Recovery to learn more. For in-depth technical information, please see the Getting Started Guide.