Disaster Recovery in the Cloud: effective techniques for Business Continuity

26 May 2023 - 10 min. read

Disaster Recovery (DR) *DR Stratagy*

It's no longer just a quote, but a mantra:

"Everything fails, all the time" - Werner Vogels

This principle has influenced the methodology behind designing cloud infrastructures. It is necessary to assess different types of failures before designing. However, not all disasters are the same. Let's try to classify them:

- Small-scale events: for example, when **a server or a VM** goes offline.
- Large-scale events: in the AWS world, we can imagine a failure of **multiple resources in a single Availability Zone** within a single Region.
- Catastrophic events: the failure affects the operation of many systems and impacts all users. The disaster affects **the entire Region**.

To effectively respond to these three types of disasters, different design and planning approaches are required:

- **High Availability**: High availability provides redundancy and fault tolerance. A system is highly available when **it can withstand the failure of one or more individual components** (e.g., hard drives, servers, or network connectivity). Production systems usually have defined uptime requirements.

- **Backup**: Backup is crucial for protecting data and **ensuring business continuity**. However, it can be challenging to implement. It is essential to keep critical data saved in case of a disaster.
- **Disaster Recovery (DR)**: A disaster is any **event that negatively impacts business continuity** or a company's finances. Such events include hardware or software failures, network disruptions, power outages, or physical damage to a building (such as a fire or flood). The cause can be human error or any other significant event. Recovery after a disaster involves a set of policies and procedures that enable the recovery or continuation of technological infrastructure and vital systems after any disaster.

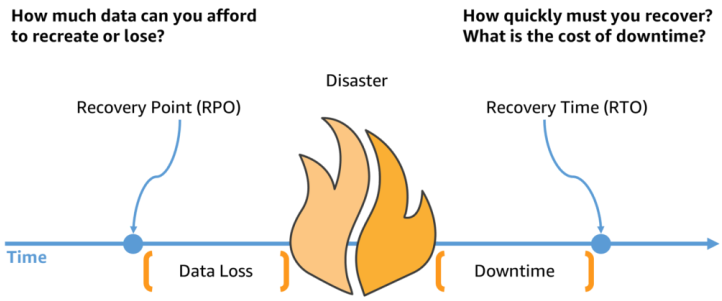
In this series of articles, we will focus specifically on Disaster Recovery, from its definition to recovery techniques and how the AWS service ecosystem can help ensure business continuity in any situation.

Disaster Recovery (DR) is a fundamental process to ensure the continuity of business operations in the event of IT disasters. There are different DR techniques, and all aim to restore applications, data, and IT infrastructure as quickly as possible after an adverse event.

In the Cloud context, DR is even more critical because applications and data are hosted remotely on servers and infrastructures that can experience service disruptions due to connectivity issues, hardware failures, cyberattacks, or natural disasters.

Whenever this topic is addressed and to ensure effective DR, it is necessary to start with two key acronyms that are essential for selecting the best DR strategy:

- RPO **(Recovery Point Objective)** - To define this key performance indicator (KPI), we need to answer a simple question: how much data can we afford to lose? RPO defines how often data should be replicated to ensure the right time window between the last replication and the disastrous event. The more frequent the replication, the lower the potential maximum data loss.
- RTO **(Recovery Time Objective)** - In this case, the question to answer is: how long can our systems remain non-functional before becoming operational again? The concept of RTO refers to the time between the disaster and the complete restoration of systems.



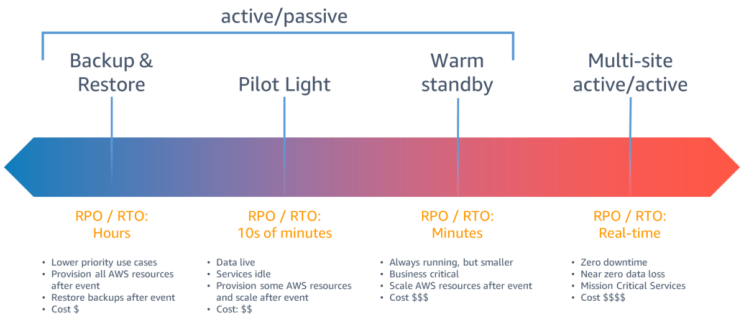
Courtesy of AWS (https://aws.amazon.com/it/blogs/architecture/disaster-recovery-dr-architecture-on-aws-part-i-strategies-for-recovery-in-the-cloud/)

In summary, DR is a fundamental process to ensure the continuity of business operations in the event of IT disasters, and its effectiveness depends on the RTO/RPO objective and the right balance between consistency, availability, and fault tolerance.

There are several approaches to Disaster Recovery, and the choice depends on factors such as budget, data and application criticality, acceptable recovery time (RTO), and acceptable recovery point (RPO). Among the most common methods for DR, we have:

- Backup and Restore: This is one of the most common techniques in disaster recovery. It involves regular backup of critical data and systems, which are later restored in case of disruption. Copies of data and applications are created on separate storage media, such as external hard drives or cloud storage services. In the event of a disaster, the data is retrieved from the backup and restored on repaired or replaced systems.
- Pilot Light: The "Pilot Light" technique involves creating a basic infrastructure essential to run critical applications. It entails having a partially configured backup environment that can be quickly scaled and activated in an emergency. Usually, only key components of the system are replicated, while the rest of the infrastructure is created at the time of restoration.
- Warm Standby: The "Warm Standby" technique involves creating a replica of the production environment, ready to be activated in case of an emergency. The standby environment is kept up-to-date with the latest data and configurations, but services are not running in real time. When a disaster occurs, the production system is halted, and the standby system is activated to resume operations.
- Multi-site: The "Multi-site" technique involves distributing systems and data across geographically separate locations. The sites can be located in separate areas, avoiding a single event from damaging all the infrastructures simultaneously. In the

event of a disaster, traffic, and services can be redirected to alternative sites to maintain operational continuity.



Courtesy of AWS (https://aws.amazon.com/it/blogs/architecture/disaster-recovery-dr-architecture-on-aws-part-i-strategies-for-recovery-in-the-cloud/)

Regarding the application of DR techniques in different-sized realities, it can be said that each organization must choose the most suitable method according to its needs and resources. Larger companies often have higher budgets and can afford to invest in expensive infrastructures like active replicas that are constantly synchronized. On the other hand, small and medium-sized enterprises can opt for more cost-effective and flexible solutions, such as data replication in the cloud.

In summary, the choice of Disaster Recovery method depends on the organization's needs, budget, and the criticality of data and applications. While the process of replicating data for disaster recovery may seem clear and straightforward, there are several technical factors to consider that can affect RPO/RTO (Recovery Point Objective/Recovery Time Objective). Some of the main factors include:

- Latency: Latency refers to the time delay between data being written to the production system and its replication at the recovery site. It is important to evaluate latency to ensure timely replication of data and to avoid significant data loss during an interruption.
- Consistency: Ensuring that replicated data is **consistent** and **intact** is essential. This means that data should be replicated in a consistent order, and write operations should be applied consistently on both sites. Inconsistency in data can lead to unexpected results or data loss.
- Scalability: During data replication, it is important to consider the ability to handle increasing data volumes over time. The replication solution should be capable of

efficiently scaling to handle larger data sizes without compromising overall performance.

- Security: Data security is crucial during the data replication process for disaster recovery. Replicated data must be protected from unauthorized access and security threats. This may require implementing encryption mechanisms, adequate access controls, and physical protection measures in both the production and recovery sites.
- Testing and Verification: Regular testing and verification are important to ensure that the data replication process functions correctly and that data can be effectively restored. Data recovery tests can help identify any issues or gaps in the process and allow for necessary corrections.
- Automation: Automating data replication can simplify the process and reduce human errors. Implementing automated tools and processes can efficiently manage data replication, improving overall consistency and reliability.
- Recovery capability: During data replication, it is important to evaluate the ability to restore them at the destination site. This may require implementing appropriate restoration procedures, having suitable hardware and software resources available, and the ability to test and verify the restoration process.

Of course, the blanket is always too short, and we must be aware that disaster recovery must necessarily be a compromise. In support of this, I would like to mention a fundamental theorem for distributed systems: the CAP theorem.

The CAP Theorem

The **CAP theorem (Consistency, Availability, Partition tolerance)**, which normally applies to distributed database systems, can be similarly applied in the context of Disaster Recovery (DR) with some additional considerations. Also known as Brewer's theorem, it states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:

- Consistency: Data consistency in the context of DR refers to the assurance that replicated data is consistent between the production site and the recovery site. In other words, the replicated data should reflect the latest changes made in the production system. However, during a disruption event and during the recovery process, a brief period of data inconsistency between the two sites may be

acceptable to ensure operational continuity. Therefore, DR may temporarily sacrifice short-term consistency to ensure availability and partition tolerance.

- Availability: The primary goal of DR is to ensure the availability of critical services and data in the event of a disruption. This means that the recovery system should be able to provide essential services in a timely manner, even if the production site is compromised. Availability in the context of DR can be achieved by sacrificing short-term data consistency or by using techniques such as restoring from backups or using dedicated recovery resources.
- Partition tolerance: Partition tolerance is crucial in the context of DR. It refers to the system's ability to continue functioning even when disruptions or partitions occur in the communication between the production site and the recovery site. Replicating data and resources in geographically separate sites contributes to ensuring partition tolerance. In case of communication disruption between the sites, the recovery system should be able to operate autonomously until communication is restored.

When applying the CAP theorem to DR, a conscious choice is often made to balance consistency, availability, and partition tolerance based on specific business needs and the consequences of disruptions. For example, in a prioritized recovery environment, availability may be given priority, temporarily sacrificing data consistency. Conversely, in an environment highly sensitive to consistency, consistency may be prioritized, temporarily sacrificing availability.

All this is just to understand how disaster recovery is done, and how we transition from production to our secondary site. But there is a crucial point in the matter that is essential and often overlooked by everyone, which is **designing how to return to production after the disastrous event has subsided**. In short, we always forget to define how to "go back."

In this case, it is important to follow a well-planned process to ensure a safe and smooth transition.

First and foremost, a recovery assessment must be conducted. Before returning to normal production, it is necessary to **carefully assess the state of the system and the production environment**. Verify that the problem or disruption event has been completely resolved and that the environment is ready for restoration. In this case, having good monitoring systems is crucial to detect any anomalies or issues. The use of monitoring tools and alerts is helpful in identifying and promptly resolving any

problems that may occur during the transition back to production. Repetition helps: monitoring is also crucial post-restoration to identify any residual issues or side effects that may arise and to intervene promptly to resolve them.

It is not enough for the production environment to be finally available; testing and validation must be performed. This may include functional testing, load testing, and other appropriate tests to ensure that everything functions correctly as expected. Verify that the data is consistent and intact.

In some cases, it may be necessary to perform a rollback of the changes made during the DR process. This may involve restoring configurations, application changes, or previous versions of data. It is important to plan the rollback in order to minimize the impact on operations and ensure data consistency.

Communication with users and stakeholders involved should not be underestimated in emergency situations. The individuals involved must be made aware of the changes made and the necessary steps to return to normal production.

To inform the involved individuals, it is necessary to document our procedures, recording all the steps, changes made, and actions taken during the DR process. This will help reconstruct the disruption event and analyze it later to improve future DR strategies.

Additionally, we conclude with a connection to the next article (on-prem to cloud and then cloud to cloud).

Conclusions

We have reached the end of the first stage of our journey in Cloud Disaster Recovery. After defining the macro concepts and understanding the dynamics, it is time to delve into the implementation of DR techniques in complex business scenarios.

We will start by exploring the best DR techniques for hybrid on-prem to cloud contexts in the next article, and then discuss DR for Business Continuity in the Cloud-to-Cloud context in the third article of our mini-series.

Are you ready? See you in 14 days on Proud2beCloud!

About Proud2beCloud

Proud2beCloud is a blog by beSharp, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!

Nicola Ferrari

Cloud Infrastructure Line Manager @ beSharp and AWS authorized instructor champion.I live my life one level at a time getting superpowers by collecting caffeine hidden here and there in my daily map. I'm a hardened internet surfer (yes, I surfed the whole internet... twice!) and tech-addicted with a passion for computers and networking. Building great IT things all nice and tidy contribute to achieving my main goal: the pursuit of perfection!

Simone Merlini

CEO and co-founder of beSharp, Cloud Ninja and early adopter of any type of * aaS solution. I divide myself between the PC keyboard and the one with black and white keys; I specialize in deploying gargantuan dinners and testing vintage bottles.
