Hybrid Disaster Recovery: strategic guidelines

9 June 2023 - 8 min. read

Disaster Recovery (DR) DR Strategy

With this article, we continue our series on Disaster Recovery by examining the key technical aspects of DR in a scenario where companies are adopting a hybrid approach that combines existing on-premise infrastructure with AWS Cloud resources.

(Did you miss part 1? Read it now!)

Both planning and implementing a robust Disaster Recovery (DR) strategy are essential to ensure business continuity in case of catastrophic events or unexpected system disruptions. In recent years, Cloud Computing has provided new opportunities to improve the efficiency and reliability of DR solutions.

This hybrid combination offers a range of benefits that allow organizations to effectively address business continuity objectives.

Why does the Cloud change the DR scenario?

Let's start by analyzing the advantages of the Cloud model, which enables us to undertake projects that would otherwise be unsustainable.

First and foremost there is the **on-demand** nature of the Cloud and AWS, which provides scalability and flexibility. In an on-premise context, resources are limited by hardware capacity and existing physical infrastructures. In the Cloud, it is possible to scale DR resources vertically or horizontally based on needs, allowing greater flexibility in managing load spikes or emergency situations.

Other essential advantages include **reliability and availability**, as there are managed services that allow us to leverage geographic redundancy, automatic load distribution, data replication across multiple availability zones, and failover management.

Certainly, thanks to the **programmatic nature of the Cloud**, we can automate, streamline, and simplify recovery processes, making them faster and more efficient.

Last, but not least, **costs** must be considered. A comparison of operational costs between the on-premise approach and AWS Cloud highlights the economic benefits of the latter. In the on-premise model, companies have to invest in hardware, software, physical space, maintenance, and specialized personnel to implement and manage a DR solution. With AWS Cloud, costs are based on the actual consumption of resources: you pay for what you truly need and use. This consumption-based approach allows companies to reduce operational costs related to DR infrastructure, eliminating the need for significant upfront investments and reducing maintenance costs.

Of course, this last concept is directly related to the type of strategy being adopted.

The main goal of DR is to maintain business continuity, which means determining which applications are the most critical to the organization and what RPO and RTO are required. A good practice is to classify workloads based on their business continuity requirements, trying to standardize them as much as possible.

Based on the RPO and RTO requirements, we can generally consider three main categories:

1. **Low criticality**: the RPO for these workloads is a few hours or even several days. The RTO is several days or more. Typically, these workloads require a less complex and expensive DR architecture, allowing for a longer recovery period.
2. **Medium criticality**: the RPO for these workloads is in the range of tens of minutes, making acceptable a minor data loss. The RTO varies from a few hours to a day. It is still important to ensure the availability of updated data to limit information loss and restore the application within a reasonable timeframe.
3. **High criticality**: this group includes workloads with an RPO of a few seconds or even zero, meaning no data loss is acceptable. The RTO could be in the range of minutes or tens of minutes, in order to quickly restore the application in case of a disruption.

Each group requires a different strategy. For example, critical workloads require a **Multi-site Active-Active** DR site, while medium criticality workloads may employ a **Pilot Light** strategy, and low criticality workloads may rely on **Backup and Restore**.

If the concepts of RPO/RTO and the most common DR strategies are not clear, you can refer to the first article of our series.

Three DR strategies and how to apply them

Combining the concepts expressed so far, it is evident that the Cloud maximizes its advantages in the Backup and Restore strategy. Let's see how it is technically possible to implement the backup and restore from on-premise to the Cloud through some considerations.

Let's focus on the data and therefore the RPO. How do we move our data to the Cloud? And even before that, what constitutes data?

Data is not only the information saved in databases or file servers but also the configurations of our environment. It is important to have systems that perform regular backups and are capable of transferring them to the secondary site, but we also need tools to keep the configurations aligned.

In this case, services like Storage Gateway and EBS come to our rescue. Storage Gateway can be configured in different modes: File Gateway, Tape, and Volume Gateway. For Backup and Restore, it makes more sense to focus on the Volume Gateway mode.

With Volume Gateway, on-premise systems mount iSCSI volumes, and applications interact with them as if they were normal block storage. Data written to these volumes is compressed and can be asynchronously copied as point-in-time snapshots and stored in the Cloud as EBS Snapshots.

Storage Gateway allows us to leverage low-cost and highly retained object storage (AWS S3) to store our data. Since it integrates with EBS, it enables us to create volumes for our EC2 instances to be restored. Simultaneously, thanks to integration with AWS Backup, we can automatically configure the retention and automatic decommissioning of our backups, making it easier to maintain control and governance over our process.

For workloads with medium and high criticality, we need to rely on tools that can create continuous replicas of our environments or portions of them.

In the case of **medium-criticality workloads**, low-latency replication of databases and files is essential, but not for the computational nodes that deliver our services. We are dealing with an RPO of tens of minutes and an RTO of a few hours. The best strategy is the Pilot Light, where some infrastructure components are always active and synchronized in the DR site, while other components are regularly replicated and ready to come into play when needed.

Once again, we can rely on Storage Gateway, but in synchronous File Gateway mode, which allows us to replicate our files to S3 or directly to the managed FSx for Windows File Server service. Alternatively, AWS DataSync replicates data by copying it to any class of S3 storage and is also compatible with any AWS managed file storage service (EFS and FSx). However, DataSync involves periodic synchronization rather than ongoing synchronization.

For our databases, AWS Database Migration Service (DMS) is the key. With DMS, we can keep our on-premise databases synchronized using Change Data Capture (CDC), which synchronizes data modification events (DML) by reading transaction logs. This allows us to keep the two databases aligned in near-real-time without overloading the production nodes.

For **high-criticality workloads**, AWS provides Elastic Disaster Recovery (AWS DRS), which allows us to restore applications on AWS from physical infrastructures, VMware vSphere, Microsoft Hyper-V, and even other cloud providers.

In the event of a disaster, a failover towards AWS is necessary with the help of AWS Elastic Disaster Recovery (AWS DRS). Once the disaster is mitigated, a failback to the original infrastructure needs to be executed. The recovery instances with AWS Elastic Disaster Recovery can be brought to the latest available version or to a specific point in time (Point-in-time - PIT).

For most organizations, the DR site is not designed to handle daily operations, and significant effort may be required to move data and business services back to the primary environment once the disaster is over. Planning for downtime or a partial disruption of activities may be necessary during the failback process to the primary site.

Once ready to resume operations on the primary system, **failback replication** will be necessary. During the usage of the recovery system on AWS, new data is written to the secondary system, and this data needs to be brought back. Failback to the source servers can be performed by installing the AWS Elastic Disaster Recovery Failback Client.

Performing recovery tests (or drills) is a fundamental aspect of being prepared for a disaster in all the cases listed above. Conducting regular recovery tests to ensure that backup data is intact and that workloads can be restored correctly is crucial for evaluating the effectiveness of the restoration process and identifying any issues or areas for improvement. For example, AWS DRS also allows us to automate drills.

Conclusions

Many of the points mentioned are based on the assumption that on-premises infrastructures are based on traditional architectures, primarily VMs. However, the cloud remains a valid choice even in more modern scenarios where containers, for example, play a prominent role. Different techniques may be required, but the concepts remain the same.

Implementing DR in the Cloud should not deter us from innovating our workloads as much as possible, striving to leverage high-level managed services to the fullest extent. This helps reduce the effort spent on infrastructure maintenance and allows us to focus on providing a higher level of service and achieving better cost efficiency.

In the next article, we will explore the techniques to be adopted in a "Cloud-to-Cloud" scenario. We will specifically examine how disaster scenarios change in such a context with a focus on security and automation.

See you in 14 days!

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.