Business continuity in the Cloud: the importance of Fault Tolerance

10 November 2023 - 9 min. read

Business Continuity | Disaster Recovery (DR) | Fault tolerance | High Availability (HA)

Introduction

One of the principles that we strive to convey to our partners every day is that the Cloud is not a magical place where I put resources, pay a bill, and automatically obtain all the advantages commonly listed in articles, white papers, and provider presentations pages.

The Cloud involves a shift in mindset and, consequently, a new way of making investments and viewing the "digital" aspects of companies. Especially for those who provide services to consumers, business continuity is becoming increasingly crucial, ensuring that we are ready to fulfill requests at any time or condition.

It's not only about understanding the financial impact of unexpected events but also about implementing all the procedures and practices that lead to maximizing our uptime.

I would like to start with some data: here are some examples of service interruptions from big corporations, which demonstrate that the issue generated is not only a matter of customer trust but also a financial one.

In 2015, a 12-hour Apple Store outage cost the company $25 million. In 2016, a five-hour power outage at a Delta Airlines operations center resulted in an estimated loss of $150 million. In 2019, a 14-hour outage cost Facebook approximately $90 million. In addition to this data, the twelfth Annual Global Data Center Survey (2022) from the Uptime Institute provides a critical overview and an idea of the industry's future trajectory. Among the key findings of the report, I would like to focus on two specific points:

- **Service outages are becoming more costly and still too frequent**. Outages costing over $1 million have increased to 25%, a significant rise from 15% in 2021. While the data indicates an improving trend, the frequency of outages is still too high. More than two-thirds of these outages cost operators over $100,000, so the consequences are worsening.
- **Operator confidence in public cloud services is on the rise, despite ongoing disruption risks**. Organizations are more inclined to trust the cloud. In 2022, only 63% of operators are not placing critical workloads in a public cloud, a significant drop from nearly 75% in 2019. However, more than a third of respondents report that disruptions in public cloud availability zones (which are relatively common) would cause significant performance issues.

When it comes to service outages, I believe no one was surprised by the results, but it's certainly interesting to note that the bar of expectations is continually rising, and the consequences of downtime are having an increasingly significant impact on earnings and user trust.

The second bullet point, in my view, relates to using Cloud technology without fully understanding its fundamentals. As we mentioned, the Cloud is not magical; simply moving a workload to the Cloud doesn't automatically increase uptime.

This article aims to be an attempt to explain how the Cloud empowers us and what we need to do to achieve **fault tolerance**, improve **uptime**, and address some aspects of a structured **business continuity** plan.

What is Fault Tolerance?

If we were in school, we should start with the definition:

fault tolerance refers to the ability of a system or application to continue functioning reliably even when failures or malfunctions occur in one or more of its components.

Being able to be fault-tolerant is important for addressing many of the objectives defined in business continuity plans. A well-designed fault-tolerant infrastructure brings us benefits such as reduced downtime, improved reliability, risk reduction, and maintaining service quality.

It's important to clarify the concept of fault tolerance and how it differs from concepts like HA (High Availability), redundancy, and DR (Disaster Recovery), often mistakenly used as interchangeable synonyms:

High Availability (HA) refers to a system's ability to remain operational, without significant interruptions, for continuous periods of time. This goal can be achieved through continuous monitoring, redundancy, and other measures. A high-availability system is designed to avoid failures and ensure continuous service availability.

Redundancy is used to enhance a system's reliability. Designing infrastructures that include redundant components means that if one component fails, another can take its place to prevent interruptions. Redundancy can be applied at both hardware and software levels.

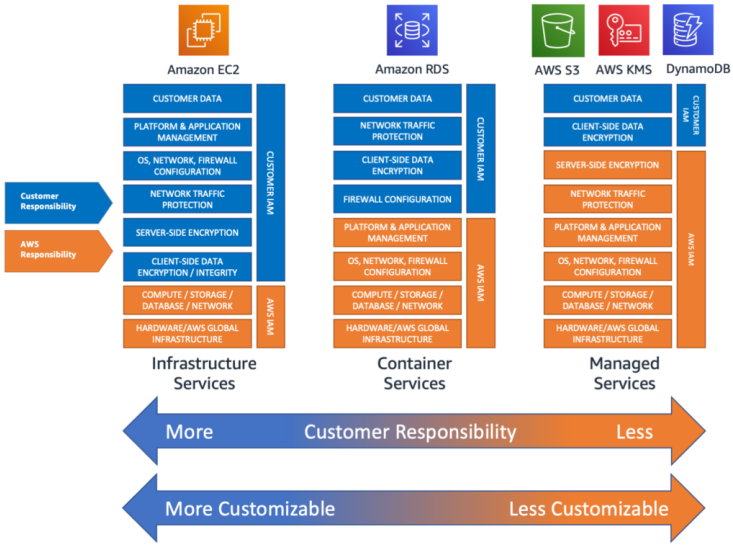
Disaster Recovery (DR) is a plan aimed at restoring a system or application in the event of catastrophic events or large-scale failures. This usually involves data backup, business continuity planning, and specific procedures to restore the system at an alternate location or with alternative resources.

High availability, redundancy, and DR are **all important components to achieve fault tolerance, but they alone may not be sufficient to guarantee it**. Fault tolerance requires a thoughtful design that considers a wide range of scenarios and provides adequate measures to manage them without significant disruptions.

Fault tolerance is, therefore, a design and implementation strategy that significantly contributes to continued reliable operation, thus protecting a company's reputation and maintaining customer trust.

Shared Responsibility Model

Those approaching the Cloud should keep in mind that providers are not, do not want to be, and cannot be responsible for our applications in their entirety. Furthermore, depending on the service or building block we use for our infrastructure, the provider shifts the responsibility bar. Taking AWS as an example, as shown in the image, more IaaS services shift the responsibility towards the Customer, while managed services shift more of the responsibility towards the provider.



https://aws.amazon.com/blogs/industries/applying-the-aws-shared-responsibility-model-to-your-gxp-solution/

This means that despite the provider implementing every form of hardware and geographical redundancy, part of the responsibility still rests with the customer.

AWS highlights these two sides of the coin by distinguishing the provider, responsible for the resilience **of** the Cloud, and the customer, responsible for resilience **in** the cloud.

AWS is responsible for protecting the infrastructure on which all services offered in the AWS Cloud run. This infrastructure includes the hardware, software, networks, and facilities on which AWS Cloud services operate. The customer's responsibility, instead, is determined by the AWS Cloud services they choose, based on the configurations they need to make and their management and design activities.

Fault Tolerance on AWS

Let's try to ground all the discussions on Fault Tolerance.

In the case of AWS, we need to start by understanding how the physical infrastructure of the data centers has been implemented. It is necessary to reiterate that AWS provides us with the tools to create resilient infrastructures, but it's up to us to design and implement them correctly.

Let's start with the Availability Zones (AZ), which are individual data centers or groups of them, logically grouped in Regions.

Redundancy is achieved by leveraging a single AZ but duplicating the instances of the service in use within it. Let's provide two practical examples: I can create two EC2 instances within which I can install MySql, set them up in a cluster, active-standby, and be resilient to provider hardware failures.

High Availability is the ability to keep services running without interruptions. In a SaaS context, it means having redundant systems so that if one component fails, another immediately takes its place without causing downtime. HA is crucial to ensure that SaaS services are always accessible to users.

To achieve high availability, you can distribute your database across two different Availability Zones. It is essential to have designed, set up and configured the Cloud environment correctly. If you are not sure what I'm talking about, I must refer you to our series of articles on the Landing Zone.

Disaster Recovery is the plan that a company implements to restore its operations after a major incident or catastrophe. In a SaaS context, this means having backup copies of critical data and services in a safe and easily accessible location. In case of an emergency, DR allows you to quickly restore services, minimizing downtime.

Disaster Recovery is achieved by utilizing two different AWS Regions worldwide. There are not many automatic mechanisms or services that allow us to achieve DR with low implementation effort. DR needs to be carefully thought out, and there are numerous aspects to keep in mind. Please check our series of articles on these aspects.

Fault Tolerance during deployments

So far, we have addressed the issue only from an infrastructure perspective, but from a software development standpoint, this topic also has an impact.

For example, it is a good practice to leverage the Cloud, which is a fully programmable environment, to create multiple development environments for testing changes before making them effective in production. The use of multiple environments for experimenting, validating, and testing workloads reduces the risk of human errors by creating an automated, controlled, and reversible process.

Even in the phases of releasing updates to our products or public portals, fault tolerance needs to be considered. For example, with **Blue-Green Deployment**, two identical environments (blue and green) are maintained in parallel. During an update, user traffic is redirected from the "blue" environment to the "green" one. This technique allows updates to be performed without downtime, and in case of issues, it is possible to quickly revert to the "blue" environment.

Blue-Green deployment can be complemented by practices like those defined in **Canary Deployment** or **Rolling Deployment**. With these techniques, the update is rolled out to a small subset of users only. These users serve as indicators to detect any issues, and if no

problems are encountered, the update is gradually extended to a broader audience; otherwise, the update is halted by redirecting the few "tester" users to the old environment.

All of these are gradual and controlled deployment approaches that reduce the risk of production failures and ensure that changes are stable and reliable before being exposed to end-users.

(Check these articles for more on Blue-Green Deployment, Canary Deployment, and Rolling Deployment: How to setup a Continuous Deployment Pipeline on AWS for ECS Blue/Green Deployments - ECS deployment strategies: reduce downtime and risk with blue/green deployment).

Chaos Engineering

At this point, the question that naturally arises is: how do I verify my fault tolerance?

Chaos Engineering is a practice designed to test the resilience and fault tolerance of a computer system in a controlled and planned manner. The main goal of chaos engineering is to identify weaknesses and vulnerabilities in the system before they can cause real production problems.

One of the examples of tools created to test the robustness of environments is Chaos Monkey developed by Netflix. You can check out the project here. This tool has this peculiar name because it aims to simulate the unpredictable behaviors of a monkey left loose inside a data center. What could happen? Unplugged cables, pressed buttons, destroyed racks, and so on.

If the infrastructure holds up and there are no downtime incidents, then you can finally say that fault tolerance has been achieved.

Conclusion

Even redundancy, high availability, and disaster recovery are not enough to improve resilience. There are other principles that should be followed, such as horizontal scalability to accommodate the load curve, automatic recovery in case of failure (self-healing), automation, and verification of all recovery procedures.

I don't want to sound cliché, but as Spiderman wisely reminds us: “With great power comes great responsibility”. The Cloud gives us the power that we must learn to harness in the right way, through study and experience.

I hope this article helped in clarifying things around Fault Tolerance, Business Continuity and all the related concepts.

We will be back soon with more on this. Stay tuned!

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!
