ARPIO

THE SEVEN SECRETS OF DISASTER RESILIENCE IN AWS

THE SUBTLETIES YOU NEED TO GET RIGHT TO BE TRULY RESILIENT.

If you're like most organizations who are journeying to Amazon Web Services (AWS), you're constantly relearning and reapplying past lessons in a new context. In the cloud, best practices continually evolve, architectures look different and you're always solving old problems in new ways.

This is especially true for how you think about the resilience of your workloads, because the cloud is no stranger to outages, and disasters still happen. The risk profile has changed, though, and the strategies to mitigate those risks have evolved.

You have plenty of resilience questions you need to answer, and the answers aren't always intuitive. Here, we'll share 7 less-obvious insights that will change the ways you see this problem, ensuring you're making the right decisions to achieve resilience in the cloud.

SECRET #1:
CLOUD DISASTERS ARE LOGICAL, NOT PHYSICAL

Ask any IT veteran about IT disasters, and they'll inevitably start talking about the weather.

Hurricanes, tornados, torrential floods, ice storms—they all have a reputation for causing bad things to happen in data centers.

And they do this in 1 of 2 ways. They either cut off essential services to a data center (like network and power), or they damage the systems inside the data center. These are physical disasters, failures of things you can touch.

AWS is susceptible to these same disasters, but these probably aren't the disasters you should focus on. Amazon builds its data centers to an incredible standard of reliability, and if you're architecting your workloads in accordance with cloud best practices, you're leveraging "availability zones" in AWS to add even more resilience. For all but the most mission-critical workloads, you've probably got weather disasters covered.

Yet disasters still happen in AWS, even to workloads that are built in accordance with cloud best practices. So how are they happening?

Most disasters in AWS happen above the physical layer, at a "logical" layer. They aren't caused by hardware failure or data center outages. Instead, they're caused by cyber attacks, human error, redundancy failures, insider threats, regional infrastructure outages and the like.

If a ransomware attack encrypts your data and deletes your backups, the incredible physical durability of EBS snapshots hasn't provided you any protection. If a disgruntled employee exacts revenge by terminating a bunch of EC2 instances, your application is down regardless of how many availability zones in which you've deployed it. And if your workload depends on a region-wide AWS service that—as happens a few times each year—is suffering a logical failure itself, you get to explain to senior executives that business can resume as soon as Amazon resolves their issue.

To mitigate the risks of these logical failures, you have to build logical redundancy yourself. In AWS, this means maintaining multiple copies of data in a protected manner. And it requires the ability to run on completely independent infrastructure in case your primary infrastructure fails.

CAUTIONARY TALE: THE CASE OF UNITED STATES V. RAMESH

Sudhish Kasaba Ramesh was once an employee of Cisco, working on the WebEx product.

He left Cisco in April of 2018, and it appears he wasn't very happy upon his departure.

Five months later, Sudhish's production AWS access had not been terminated, so he used his access to terminate 456 EC2 instances that were responsible for the WebEx Teams application. This criminal act is currently the subject of a case in U.S. federal court.

16,000 WebEx Teams customers were impacted by what was ultimately a 2 week outage. Cisco spent $1.4 million on employee labor to recover the service and paid another $1 million in SLA penalties to their customers. That's a $2.4 million logical disaster in the cloud.

SECRET #2:
HIGH AVAILABILITY IS NOT DISASTER RECOVERY

Most resilience strategies for modern cloud workloads start by solving for high availability. In this approach, you leverage techniques like clustering, health checks and automated failover to introduce both redundancy and self-healing into your applications.

In AWS, the most common approach to high availability is to deploy workloads redundantly across "availability zones." Each availability zone in an AWS region consists of one or more independent data centers. By deploying workloads across multiple availability zones, your workloads can be made resilient to the types of failures that would impair components within a data center or an entire data center itself.

But how do you protect workloads from failures that aren't confined to a data center? As we've established, most cloud disasters happen above this physical layer.

Logical disasters from human error, bad actors, cyber-threats and region-wide outages span availability zones. A multi-zone approach to resilience is insufficient here.

All high-availability solutions have limits, and when your workload encounters a failure that extends beyond these limits, you have a disaster. This is where your separate disaster recovery strategy needs to take over.

SECRET #3: YOU NEED MULTI-REGION, BUT NOT FOR THE REASON YOU THINK

AWS operates in more than 20 locations around the world. They call these locations "regions," and most regions are separated by hundreds of miles.

A common consideration for redundant IT workloads is the physical separation between redundant components. The thought here is to ensure a single event won't undermine all of your redundancy.

Some businesses embrace a requirement where redundant components must be separated by hundreds of miles. This allows them to endure earthquakes, hurricanes and other regional catastrophes that could otherwise take them offline. To meet this requirement, these businesses deploy their to multiple AWS regions.

Meanwhile, other businesses don't adopt this distance requirement, and deploy their workloads to multiple availability zones within a single region.

But while physical distance is one motivation for multi-region redundancy, the most critical motivation is isolation. Each AWS region is architected to be completely self-contained, taking no dependencies on resources outside the region. Meanwhile, availability zones are not completely independent.

Impairments within the region can impact all availability zones in the region, and region-wide services suffer region-wide outages when they go down.

While it’s tempting to dismiss a multi-region approach to resilience if your business does not impose requirements for geographic separation, that decision is not fully informed. The primary reason for a multi-region approach is to ensure your workload is deployed across redundant and independent infrastructures. The only way to get truly independent infrastructures in AWS is to leverage multiple regions.

CAUTIONARY TALE: THE 7+ HOUR SYDNEY OUTAGE OF 2020

The Sydney, Australia region of AWS has had plenty of outages in recent years. Perhaps the most impactful was a 7.5-hour outage of the VPC service in January, 2020.

You’re probably aware that the VPC service underpins most of the networking in AWS. It’s a region-wide service, so it’s engineered to be resilient to many types of failures within the region.

But on this day, Amazon suffered a catastrophic failure of the primary database that supports the VPC service. Fixing the issue required a multi-hour rebuild of that database server, and during the rebuild the service was forced to operate from read-only replicas. No changes could be made to customer networking.

Some workloads operate just fine without making networking updates, but modern cloud applications are elastic and dynamic, continuously making networking updates throughout the day. For 7.5 hours this day, the most resilient workloads in Sydney were completely unavailable while the region-wide infrastructure was rebuilt.

SECRET #4: DURABILITY IS NOT AVAILABILITY

The storage services in AWS advertise impressive “durability” metrics. For example, io2 volumes in EBS are built for 99.999% durability, which means only one in 100,000 volumes fails annually. S3 storage offers an astounding 99.999999999% durability, which equates to losing one file out of 100 billion files in a year.

This impressive durability is achieved by redundantly storing data across multiple physical media, and these metrics are a reflection of the failure rate of that underlying media. S3 Standard, for example, stores data redundantly across multiple devices in at least 3 availability zones, continually revalidating that the data has not been lost or corrupted.

But those redundant copies of data are still a single logical entity. One human error, one malicious act or one software bug is all it takes for that data to disappear.

To ensure your data is always available despite the threat of logical failures, you need to store multiple logical copies of that data. You also need to secure at least one copy of the data where it can't be deleted or corrupted by an employee mistake or a bad actor.

SECRET #5: DATA REPLICATION IS NOT DATA PROTECTION

A common approach to achieving disaster resilience involves replicating data in real time to an alternate AWS region, where it's available in case of an outage in your production region. This is typically achieved using asynchronous replication that's built into most common database platforms. If a disaster strikes the production region, the alternate region replica is promoted to become the primary database, while new compute is launched in that region to service end users.

This can be a simple and elegant solution to protect against the possibility of a region-wide service outage, but this solution doesn't protect your data from other types of disasters.

If data is deleted from a production database via human error, that deletion is quickly replicated to the alternate region. If ransomware corrupts or encrypts data in the database, it becomes corrupted in the other region as well. And if a bad actor compromises your AWS account, he or she can find and destroy the databases in both regions.

backups, so that data corruption can be cured by rolling back to a time before the corruption. Those backups must be secured, so that no malicious actor can destroy both your data and the backups necessary to restore it.

CAUTIONARY TALE: MURDER IN THE AMAZON CLOUD

Codespaces was once an up-and-coming competitor to services like Microsoft's GitHub. Over the course of seven years, they built a healthy business by offering boutique source code hosting to companies in highly regulated industries. They built their brand around the strength of their business continuity program.

One morning, they woke up to find that their service was under a distributed denial of service attack. While troubleshooting, they logged into the AWS console where they found a note from the attacker demanding a ransom to stop it. The attacker had compromised their AWS account.

Codespaces did what we'd all do and began revoking the attacker's AWS access, but they didn't move quickly enough. When the attacker detected their defensive actions, he deleted everything in the account: servers, backups, buckets. All of their customers' critical data was lost, and they hadn't secured a copy of that data elsewhere.

Within days, and with no possibility to restore their service, Codespaces announced they would be shutting down their business.

SECRET #6: RAPID RECOVERY TIME ENABLES RECOVERY TESTING

Let's face it, most organizations never test their disaster recovery process. And the ones who do test this process do it rarely and find it to be unreliable. Consequently, most organizations don't trust that their DR plans will work when they need it.

But testing is fundamental in IT. So why aren't companies doing it? Because it takes too much time.

Disasters are supposed to be rare, and we all hope to never experience one. Given this rarity, many businesses are comfortable with a recovery time measured in several hours or perhaps days. Their recovery processes are engineered for this "slow recovery is okay" requirement.

But if recovery is slow, then testing recovery is also slow. And as all IT organizations are under-resourced and overcommitted, finding the time to perform this test becomes a challenge.

When considering business requirements for disaster recovery time (your “recovery time objective”), you need to consider more than the waiting period your business can endure if disaster strikes. You also need to consider how frequently you’ll need to verify your recovery process and how much time can be afforded to the verification process each time.

SECRET #7:
AUTOMATION IS THE ONLY TENABLE WAY

Amazon Web Services is huge, with well over 200 services depending on how you count them, and your organization benefits greatly from the increased agility and velocity that these options provide.

But embracing these options entails complexity. In an on-premises environment, you generally deal with networking, storage and compute. In AWS, you grapple with 27+ networking entities, 14+ types of compute and 17+ flavors of storage. And this is all before you start to use the numerous higher-level services available on the platform.

If you aren’t diligent about managing this complexity, things can quickly get out of hand. Before you know it, you have an environment that was assembled over time through a mix of manual setup and infrastructure automation, and no single team member understands how to rebuild it when needed.

The key to managing the complexity of recovering a cloud environment is automation. It eliminates the challenges of fragmented and lost knowledge about how your environment is built. It handles the numerous tiny details that humans are likely to get wrong during a stressful recovery. It enables frequent testability by greatly accelerating the recovery process. And it ensures that recovery is consistent and predictable when you need it most.

PUTTING IT ALL TOGETHER

As we established earlier, disaster resilience for AWS workloads is a complex architectural challenge that requires you to think broadly about risks that could impact your workloads. Risks in the cloud are different than risks on-premises, and they require different approaches to mitigation.

But there's a common pattern used to broadly eliminate these risks in AWS. This pattern involves creating the ability to failover your workloads to an alternate region and to an alternate AWS account.

Alternate region failover protects you from infrastructure failures and service outages in your production AWS region. It ensures that your workloads can continue operating in the face of any platform outage in your production region, including outages that impact multiple availability zones and region-wide services.

Alternate account failover protects you from cyber attacks and human errors. By storing your data in an alternate AWS account and minimizing access to that alternate account, you can eliminate the attack vectors that could be leveraged to destroy data and infrastructure in your production environment.

Together, these strategies enable you to comprehensively mitigate the risk of catastrophic downtime and data loss for your AWS workloads.

ABOUT ARPIO

At Arpio, we've created a SaaS solution that gives AWS customers nearly instant recovery protection from outages and data loss—whatever the cause. We recognize that your AWS environment is more than just EC2 instances, so we protect and recover all of your critical application resources—in addition to your data—ensuring you can always get back online quickly.

What's more, Arpio takes just minutes to set up and can be implemented at a fraction of the cost of traditional large-scale DR solutions. Arpio gives you best-in-class recoverability for your AWS workloads that you don't have to build yourself, allowing your team to focus their time on the more strategic initiatives that truly differentiate your business.

ARPIO