

# Pitfalls of an Unmanaged AWS Environment

Enterprises of all shapes and sizes are deploying a variety of workloads into the cloud, and taking advantage of all that cloud computing has to offer - **elastic scalability**, **agility**, and a **pay-as-you-go** model.

As cloud adoption rates pick up and cloud footprints increase for organizations, there are a set of unique but predictable issues that start emerging, which if left unchecked can lead to a lot of challenges down the road. It's not uncommon for a sense of disillusionment to creep in as you struggle with your deployment, especially while hearing stories of other businesses that are striking big wins in the cloud. In reality these stories are about a small subset of organizations that plan their growth and run a tight, managed environment with the help of a trained team of cloud professionals; however, a majority of organizations choose to strike out on their own initially, and have a completely different story to tell.

Over the years, we have seen numerous companies taking the same path to the cloud, making the same mistakes and running into the same set of issues. At the end of the day, the very benefits of cloud computing, if not properly managed, can lead an environment to quickly become **chaotic**, **insecure** and **costly**. So, what exactly are these pitfalls? Here's what we learned from our engagements with several large, high-traffic web properties that initially built and managed their own AWS infrastructure prior to engaging us as their AWS partner:

## **Pitfall 1: Getting Stuck With Sub-Optimal Architecture Choices**

The fast pace of innovation at AWS has resulted in new services being introduced at breakneck speed. This creates a large number of complementary, and sometimes confusing architecture choices for businesses, who invariably – due to limited planning and pressing deadlines – end up making technology decisions that are less than optimal.

The resulting environment if left unmanaged becomes needlessly complex, doesn't effectively use new services and features, and fails to meet all the business goals and expectations. In such cases, security, availability, and scalability are all compromised. Businesses that don't adapt quickly get sucked deeper into the rabbit hole. In the case of one particular client of ours, a large media property, not using RDS initially for their database implementation led to a significantly more complex setup of their failover configuration, and furthermore, delayed the introduction of features such as automated on-demand snapshots for their development environment.



## **Pitfall 2: Failure To Control The Environment Sprawl**

Most organizations (unless you are born-in-the-cloud) take a phased approach with their cloud adoption strategy, incrementally moving different workloads or environments into the cloud. Unless this process is closely managed and controlled through standardization, automation and configuration management tools, it can quickly lead to disparate hybrid environments, where everything from access control mechanisms to application stack versions are out of sync. Furthermore, ownership of resources and responsibilities start becoming fuzzy across the environments. The end result is a chaotic infrastructure that is difficult to manage, costly to maintain, and eventually requires a forklift to upgrade.

Another one of our clients had a multitude of web media properties under their corporate umbrella, running on a variety of different platforms. The infrastructure eventually spread out across multiple AWS accounts, third-party data centers, and on-premises, leading to inconsistencies across the different environments. Several new initiatives were delayed until core production environments went through a complete re-architecture and upgrade process (which we managed and led), a side benefit of which was the deprecation of many unused instances and databases, and eventual containment of the environment sprawl.

## **Pitfall 3: Lack of Production Systems Discipline**

Many IT organizations tend to take the “Infrastructure As Code” paradigm to heart. The Dev team is typically asked to take on new devOps responsibilities, while traditional operations functions are sidelined and critical systems-related tasks go ignored. It is not uncommon to find environments that use the latest development tools and technologies, but fail to articulate and implement a cogent backup or DR strategy.

The absence of a “production systems mindset” deals a tremendous setback to the overall solutions architecture, important initiatives and ongoing project execution. AWS provides a plethora of technologies to efficiently monitor and manage the environment from a holistic operations perspective, but without the right team to implement and take advantages of these tools, opportunities for automation, innovation and cost control start languishing.

### **How Do You Spot The Problem?**

Watch out for the early warning signs:

- Do you have an uncontrolled AWS environment with multiple accounts, mysterious instances with no owners, and unpredictable costs?
- Does it take an inordinate amount of time to initiate customized instances every time you need to scale, test, or deploy?
- Do you wonder if there are better ways to optimize, secure and manage your environment?





thaumatargix

THE WORKING OF MIRACLES™



Overcoming these pitfalls is essentially an exercise in creating an overall better architected and managed environment. Once you recognize that a problem exists, remediation is essentially employing the same bag of tricks, which includes optimizing the environment, instituting devOps best practices, and streamlining operations at all levels. And it is not difficult to implement as our clients have found out. Tgix has been working with them to, among other things, migrate their various web sites, databases and media properties into a standardized and secure VPC/VPN-based architecture in AWS, along with a variety of automation and devOps tools and procedures.

Have you seen the same pitfalls in your AWS environment? Contact us for a free evaluation and recommendations for a better architected and managed environment.



## Case Study:

### ***AWS Re-Architecture and Ops Support***

**“Yesterday’s launch was the smoothest in company history. When we moved to FreeBSD and new servers in 2011 the launch went on for 6 weeks! You and your team did an amazing job, and everyone here recognizes it! Thank you so much.”**

**-- Director of IT, CollegeHumor.com**

#### **CHALLENGES**

- Multiple sites operating under CollegeHumor umbrella on a variety of different platforms
- Distributed infrastructure: On-premises, Third-Party Data Center, Public Cloud
- Existing devOps team with minimal knowledge and documentation of legacy software and platforms
- No standardization or configuration management, leading to inconsistencies across different environments

#### **SOLUTION:**

- Architecture and migration of all media properties to a highly-available and secure infrastructure on the AWS cloud
- Streamlining of environments and infrastructure standardization using Chef
- Development of devOps standards, procedures, and web-based documentation / knowledge repository
- Round-the-clock cloudOps, devOps, prodOps support

#### **BENEFITS**

- Consistent environments running standardized tools, software, platforms
- Standardized, documented Ops/Dev/QA processes & procedures
- Significantly improved monitoring capabilities & uptime

*CollegeHumor, owned by IAC / InterActiveCorp, is a popular online destination that serves comedy videos and original content to millions of users daily through several media properties. <http://www.collegehumor.com>*