Five Cloud Deployment Mistakes That Will Cost You

A KENTIK WHITEPAPER

Authors:
Jim Meehan, Product Marketing Director
Crystal Li, Senior Product Marketing Manager

The rapid expansion of cloud-deployed applications is perhaps the most obvious news within the tech landscape. The pace of this expansion is no surprise either—cloud technology is laden with benefits and promises. Most often we hear about benefits like "faster," "lower cost," "at the speed of business," "operational efficiency," "high performance," "scale rapidly," and "improved ROI."

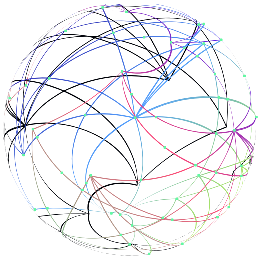
Cloud technology can be complex though, especially when it comes to networking. It's full of new concepts like VPCs, cloud interconnects, and multiple availability zones and regions. Combined with the rapid pace of deployment and lack of visibility into how cloud resources are being utilized, it's easy to make costly mistakes.

*It's **easy** to make*
costly mistakes.

In this whitepaper, we'll discuss five network-related cloud deployment mistakes that you might not be aware of, but that can negate the cloud benefits that you're hoping to achieve. The good news is: There's always an opportunity to learn from mistakes.

MISTAKE #1
DUPLICATE SERVICES AND UNKNOWN DEPENDENCIES

With teams running fast in parallel, it's not hard to see how they may end up repeatedly "reinventing the wheel."



This mistake happens most commonly when multiple, siloed teams jump into the cloud without much thought to shared architecture. Imagine separate teams building separate applications in the cloud. Each team spins up cloud resources such as compute, storage, network components, and so on. Many of them are actually reusable and shareable. Some of them are obvious standard services like DNS, databases, load-balancers, etc. But we also see duplication of custom-developed microservices that perform exactly the same function. With all the teams running fast in parallel, it's not hard to see how they may end up repeatedly "reinventing the wheel." Soon, the cloud environment becomes a tangled web of interdependencies. Without some kind of visibility, negative outcomes are likely, including brittle architecture, wasted development effort, and massive cloud overspend.

A few specific examples:

- Team A sets up a DNS service for their apps. Not knowing about Team A's DNS service, Team B creates a duplicate DNS service for their own app. Now their organization is paying twice for instances that could easily be collapsed into shared infrastructure.
- Now imagine that a third team (Team C) also needs a DNS service for their app. They begin using Team B's DNS without their knowledge. Some time later, Team B learns about Team A's DNS service, so they start using it and shut down their own. Now Team C's app has an outage because a service it was dependent on has disappeared without warning.

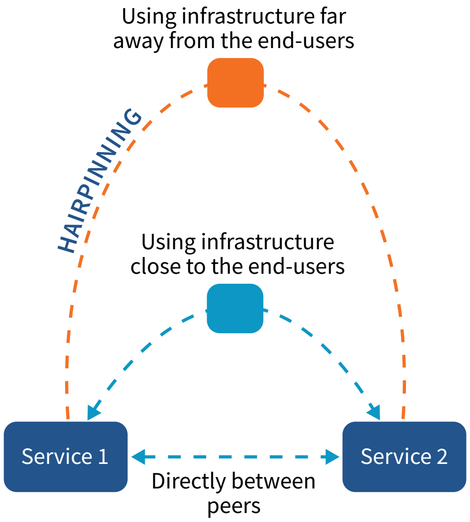
The flexibility and agility of cloud infrastructure is great, and individual teams running fast to develop their own apps is a true benefit. However, without visibility and processes to discover overlap, software development hygiene suffers and cost and/or reliability failures become much more likely.

MISTAKE #2:
TRAFFIC OR REQUEST HAIR-PINNING

Hairpinning happens when services that should communicate over short, fast, and cheap network paths end up communicating over long, expensive paths with lots of latency.

What is hair-pinning? It happens when services that should communicate over short, fast, and cheap network paths end up communicating over long, expensive paths with lots of latency. There are multiple causes and flavors of this problem. We'll discuss two common examples here.

For the first scenario, imagine two services sitting in separate zones within a physical data center. Through poor architecture choices or simple IP routing misconfiguration, the communication path between those services traverses cloud interconnects and VPCs instead of the local data center network fabric. Now every time those services communicate, they're experiencing much higher latency and also racking up expensive per-GB cloud data transfer charges. The inverse is also possible: two cloud-deployed services communicating via a network path that traverses a physical data center.


Using infrastructure far away from the end-users
HAIRPINNING
Using infrastructure close to the end-users
Service 1
Service 2
Directly between peers

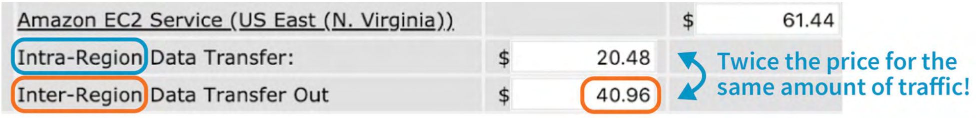
For the second scenario, imagine a service chain consisting of a web front-end, application server, and database backend. A DevOps team has migrated the application server to the cloud, but the web front-end and database are still in the legacy data center. Now every web request results in a chain of calls that traverse cloud interconnects twice, again resulting in poor performance and cost exposure.

Without visibility, scenarios like these can persist indefinitely.

MISTAKE #3
UNNECESSARY INTER-REGION TRAFFIC

It's relatively common knowledge that most cloud components are priced using a metered, pay-as-you-go model. But the pricing details of specific components often get lost in the weeds. Did you know that the per-GB cost of inter-region data transfer is (significantly) more expensive compared to intra-region data transfer? See the calculation below from the AWS pricing calculator. *Inter*-region data transfer is twice the price of *intra*-region data transfer:

Did you know that the per-GB cost of inter-region data transfer is (significantly) more expensive compared to intra-region data transfer?



Amazon EC2 Service (US East (N. Virginia))			$	61.44
Intra-Region Data Transfer:	$	20.48		
Inter-Region Data Transfer Out	$	40.96		

Twice the price for the same amount of traffic!

It's a similar story on Google Cloud Platform (GCP), according to the general info on GCP network pricing:

- Egress between zones **in the same region** costs $0.01 per GB
- Egress **between regions** varies in the range of $0.11 to $0.22 per GB

That's at least 10 times more expensive!

So here's another mistake we've seen many times that drives up cloud bills. Cloud architects and developers build the application or infrastructure in a way that generates significant, unnecessary traffic between regions. It's easy to do when you're unaware of the associated costs.

To streamline cloud network costs, we recommend following three steps:

- **Identify:** Get a view of major contributors to inter-region internet egress traffic. Answer questions such as: What applications are causing that big bandwidth bill? Which application teams are responsible for this traffic? And, are we exposed to data transfer charges that are unnecessary?
- **Relocate:** Evaluate workloads that could be possibly moved within the infrastructure. Optimize workload placement to minimize inter-region communication.
- **Consolidate:** Carefully examine all workloads, understand the network traffic they generate, and consolidate those that can be merged to save more resources.

Without knowledge of cost and the visibility to see how resources are utilized, it's easy to forget about optimization.

MISTAKE #4
MISSING COMPRESSION

People often assume a certain amount of "cloud magic" — i.e. that it's pre-configured to achieve a supreme level of efficiency. There's a kernel of truth in this because cloud providers do consolidate hardware layers to operate at a massive scale that brings down cost. However, higher up in the stack, when we think about building apps and services, proper configuration and optimization still depend mostly on the end user.

Cloud resources may be limitless, but they're definitely not free. Without knowledge of cost and the visibility to see how resources are utilized, it's easy to forget about optimization, or even overlook very basic configuration steps that lead to significant cost exposure. For example:

- Have you checked web server and proxy configurations (Apache / Nginx) to make sure that responses are compressed?
- Converting inter-service data streams from ASCII to binary can bring significant efficiency gains. For example, converting a data-heavy interface from JSON-over-HTTP to protobuf.

Understanding the network behavior of cloud services makes it easy to find these mistakes and opportunities.

MISTAKE #5
INTERNET TRAFFIC DELIVERY

Discovery becomes critical for cost control.

Using your cloud provider's default internet egress is definitely easy, but the costs can add up quickly, especially when your business needs to deliver tons of bits. Costs can be more than 10 times as expensive as traditional IP transit on a per-GB basis. Without considering and implementing other traffic delivery options, cloud migrations can result in a huge billing surprise.

A simple first step is to inventory which applications are delivering traffic to the internet, and how much. Teams may deploy new apps using default internet egress simply because they aren't aware of the cost impact. Discovery becomes critical for cost control.

Some traffic delivery options that can reduce internet egress charges:

- **Leverage CDNs (Content Delivery Networks)**: These services can cache frequently requested objects on nodes distributed all over the world, which can serve the traffic to end users at a substantially lower per-GB cost than serving those users directly from your origin servers, and with much better latency / performance as well. Cloud providers offer their own built-in CDN services, along with many third-party providers.
- **In-app packaging for mobile**: For cloud services that interact with mobile apps, consider packaging large, relatively static objects as part of the app instead of serving them over the network.
- **Private egress**: For services that generate a lot of traffic, it can be more economical to backhaul internet egress over cloud interconnects to a PoP that's well-connected to lots of relatively cheap IP transit. This option can be especially attractive for networks who already have PoPs and transit in place supporting traditional physical data centers.

Summary

With physical network infrastructure abstracted away in the cloud, it's easy to think that networks are always fast, reliable, limitless, and cheap. This fallacy is one of the biggest drivers of cloud deployment mistakes. Cloud deployments at scale are hugely complex. Without pervasive visibility, it becomes impossible to uncover mistakes that lead to poor performance and reliability, unnecessary spend, and ultimately, failed projects.

If you recognize any of the above as familiar experiences, check out Kentik's cloud visibility solution to set things straight. We'd love to help you undo (and learn from!) your cloud deployment mistakes.

If you need a Kentik account, you can sign up for a free trial.