AWS Wide Spread Outage

Learn how an AWS outage halted services for individual users and businesses across the globe.

We'll cover the following

- Introduction
- Sequence of events
- Analysis
- Lessons learned

Introduction

Several Amazon services and other services that depend on AWS were disrupted by an outage incident that spanned more than eight hours on Tuesday, December 7, 2021, at approximately 7:35 a.m. PST. The incident impacted everything from home consumer products to numerous commercial services.

This hours-long outage made headlines in the popular media, such as this one from the *Financial Times*: “From angry Adele fans to broken robot vacuums: AWS outage ripples through the US.” The outage affected millions of users worldwide, including individuals who were using the AWS online stores and other businesses that relied heavily on AWS for providing their services.

The disruption caused by AWS emphasized the need for a decentralized Internet where services don’t rely on a small number of giant companies. According to Gartner, 80% of the cloud market is handled by just five

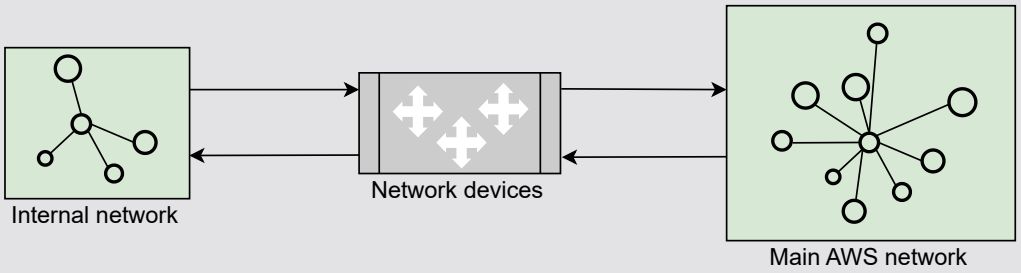
companies. Amazon, with a 41% share of the cloud computing market, is the largest.

Outages like the one above remind us of famous Lamport's quip: "A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."

Sequence of events

- An automated action to expand the capacity of one of the AWS services near the main AWS network elicited unusual behavior from a significant number of customers within the internal network.
- As a result, there was a significant increase in connection activity, which swamped the networking equipment that connected the internal network to the main AWS network.
- Communication between these networks got delayed. These delays enhanced latency and failures for services interacting between these networks, leading to a rise in retries and ping requests.
- As a result, the devices connecting the two networks experienced constant overload and performance difficulties.
- This overload instantly affected the availability of real-time monitoring data for AWS internal operations teams, hampering their ability to identify and remedy the cause of the congestion.
- Operators relied on logs to figure out what was going on and initially observed heightened internal DNS failures.

The following slides show the series of events that led to the outage.



The high-level infrastructure of Amazon. The main AWS network connects the internal network using networking device

1 of 7

Analysis

- **Hampered AWS services**: The networking difficulties affected a variety of AWS services, impacting customers that utilized these service capabilities. Since the primary AWS network remained unaffected, certain client applications that don't depend on these capabilities suffered relatively minor consequences as a result of this occurrence. AWS users, such as Amazon RDS, EMR, and Workspaces, were unable to generate new resources due to the inability of the system to launch new EC2 instances.
- **Impaired control plane**: Apart from the AWS services, the AWS control planes that are used for establishing and managing AWS resources were also impacted. These control planes take advantage of internal network-hosted services. For example, EC2 instances weren't affected by this event, but EC2 APIs suffered from increased latency and error rates.
- **Slow restoration**: Since DNS is the basis for all communication across th web, operators focused on moving the internal DNS traffic away from congested areas of the network in order to improve availability. However, since monitoring services were unavailable, operators had to identify and

disable major sources of traffic manually. This further improved the availability of services.

- **Elastic Load Balancers (ELB)**: Current Elastic Load Balancers were unaffected by the incident. However, the rising API error rates and latencies for the ELB APIs resulted in longer provisioning times for new load balancers.

Lessons learned

- **Independent communication system**: While the intention of having an internal network that's separate from the main network is the right idea, they weren't truly independent. A sequence of events highlighted their dependency. Finding such dependencies is crucial to truly benefit from independent networks for internal service use and external client use.
- **Contingency plan**: Although AWS takes measures to prepare its infrastructure for sudden surges in customer requests or power usage, the organization still found itself in a difficult situation due to the unusual severity of the failure. Investing in greater risk-based contingency planning benefits organizations during times of crisis.
- **Ready operations team**: A bug bringing an overall system to a halt is a single point of failure, which is possible in a complex system. The production team should be trained and ready for such events.
- **Multiple cloud computing providers**: Organizations can replicate their operations among many cloud computing providers so that no single failure knocks them out of action. However, this is easier said than done. An alternative approach is to employ different regions of the same provider for various purposes.
- **Testing**: Carrying out proper testing and identifying the potential bugs are both essential. In this case, overwhelming the network devices resulted in communication delays between these networks.

Point to Ponder

Question

What can we do to safeguard against the series of faults experienced by Amazon?