Summary of the December 24, 2012 Amazon ELB Service Event in the US-East Region

We would like to share more details with our customers about the event that occurred with the Amazon Elastic Load Balancing Service ("ELB") earlier this week in the US-East Region. While the service disruption only affected applications using the ELB service (and only a fraction of the ELB load balancers were affected), the impacted load balancers saw significant impact for a prolonged period of time.

The service disruption began at 12:24 PM PST on December 24th when a portion of the ELB state data was logically deleted. This data is used and maintained by the ELB control plane to manage the configuration of the ELB load balancers in the region (for example tracking all the backend hosts to which traffic should be routed by each load balancer). The data was deleted by a maintenance process that was inadvertently run against the production ELB state data. This process was run by one of a very small number of developers who have access to this production environment. Unfortunately, the developer did not realize the mistake at the time. After this data was deleted, the ELB control plane began experiencing high latency and error rates for API calls to manage ELB load balancers. In this initial part of the service disruption, there was no impact to the request handling functionality of running ELB load balancers because the missing ELB state data was not integral to the basic operation of running load balancers.

Over the next couple hours, our technical teams focused on the API errors. The team was puzzled as many APIs were succeeding (customers were able to create and manage new load balancers but not manage existing load balancers) and others were failing. As this continued, some customers began to experience performance issues with their running load balancers. These issues only occurred after the ELB control plane attempted to make changes to a running load balancer. When a user modifies a load balancer configuration or a load balancer needs to scale up or down, the ELB control plane makes changes to the load balancer configuration. During this event, because the ELB control plane lacked some of the necessary ELB state data to successfully make these changes, load balancers that were modified were improperly configured by the control plane. This resulted in degraded performance and errors for customer applications using these modified load balancers. It was when the ELB technical team started digging deeply into these degraded load balancers that the team identified the missing ELB state data as the root cause of the service disruption. At this point, the focus shifted to preventing additional service impact and recovering the missing ELB state data.

At 5:02 PM PST, the team disabled several of the ELB control plane workflows (including the scaling and descaling workflows) to prevent additional running load balancers from being affected by the missing ELB state data. At the peak of the event, 6.8% of running ELB load balancers were impacted. The rest of the load balancers in the system were unable to scale or be modified by customers, but were operating correctly. The team was able to manually recover some of the affected running load balancers on Monday night, and worked through the night to try to restore the missing ELB state data to allow the rest of the affected load balancers to recover (and to open all of the ELB APIs back up).

The team attempted to restore the ELB state data to a point-in-time just before 12:24 PM PST on December 24th (just before the event began). By restoring the data to this time, we would be able to merge in events that happened after this point to create an accurate state for each ELB load balancer. Unfortunately, the initial method used by the team to restore the ELB state data consumed several hours and failed to provide a usable snapshot of the data. This delayed recovery until an alternate recovery process was found. At 2:45 AM PST on December 25th, the team successfully restored a snapshot of the ELB state data to a time just before the data was deleted. The team then began merging this restored data with the system state changes that happened between this snapshot and the current time. By 5:40 AM PST, this data merge had been completed and the new ELB state data had been verified. The team then began slowly re-enabling the ELB service workflows and APIs. This process was done carefully to ensure that no impact was made to unaffected running load balancers and to ensure that each affected load balancer was correctly recovered. The system began recovering the remaining affected load balancers, and by 8:15 AM PST, the team had re-enabled the majority of APIs and backend workflows. By 10:30 AM PST, almost all affected load balancers had been restored to full operation. While the service was substantially recovered at this time, the team continued to closely monitor the service before communicating broadly that it was operating normally at 12:05 PM PST.

We have made a number of changes to protect the ELB service from this sort of disruption in the future. First, we have modified the access controls on our production ELB state data to prevent inadvertent modification without specific Change Management (CM) approval. Normally, we protect our production service data with non-permissive access control policies that prevent all access to production data. The ELB service had authorized additional access for a small number of developers to allow them to execute operational processes that are currently being automated. This access was incorrectly set to be persistent rather than requiring a per access approval. We have reverted this incorrect configuration and all access to production ELB data will require a per-incident CM approval. This would have prevented the ELB state data from being deleted in this event. This is a protection that we use across all of our services that has prevented this sort of problem in the past, but was not appropriately enabled for this ELB state data. We have also modified our data recovery process to reflect the learning we went through in this event. We are confident that we could recover ELB state data in a similar event significantly faster (if necessary) for any future operational event. We will also incorporate our learning from this event into our service architecture. We believe that we can reprogram our ELB control plane workflows to more thoughtfully reconcile the central service data with the current load balancer state. This would allow the service to recover automatically from logical data loss or corruption without needing manual data restoration.

Last, but certainly not least, we want to apologize. We know how critical our services are to our customers' businesses, and we know this disruption came at an inopportune time for some of our customers. We will do everything we can to learn from this event and use it to drive further improvement in the ELB service.

Sincerely,
The AWS Team