Dell Resilient Kubernetes

UC Santa Cruz Baskin Engineering

Ryan Ahrari, Nathanael Chow, Juan Lee, Ronald Chhua

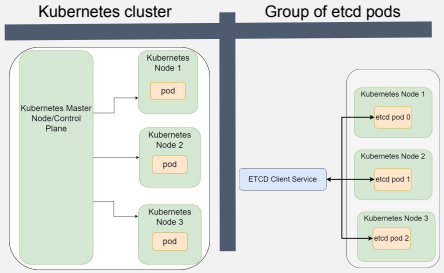
DELL

Abstract

The Resilient Kubernetes project works to enhance the protection of data within Kubernetes, a Linux container management software, for Dell. Kubernetes is a part of Dell's Infrastructure Solution Group which provides remote computer infrastructure resources to various corporations that would like to use a remote IT solution. We enhance this data protection by making etcd - a key-value store used by Kubernetes to hold essential cluster data - more resilient.

Approach

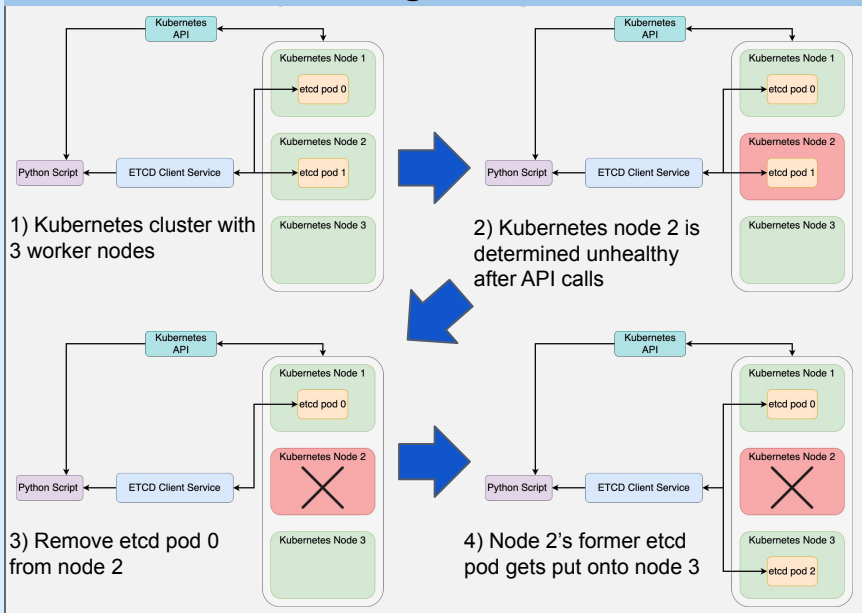
- Kubernetes is a cluster of servers (nodes in Kubernetes) where one server is the master node and acts as the control plane for the cluster, and the rest of the servers are worker nodes that rely on the master for scheduling and resource allocation.
- Etcd works as a group of pods (in Kubernetes we refer to container as pods) that runs on a Kubernetes cluster. Etcd works on the Raft consensus algorithm and as such has a leader-follower consensus protocol.
- We wrote a script that communicates with the etcd and Kubernetes client APIs to get several fields of information both about Raft metrics, and Kubernetes cluster metrics to determine if a particular node is unhealthy, or if a pod is unhealthy or unstable.



Overview

When we determine that a node is unhealthy - meaning its memory or CPU usage is too high - or a pod is unhealthy - meaning it is unresponsive when we try to communicate with it - or the etcd group is unstable - meaning the group is at risk of losing half its pods and the whole etcd group would be at risk of being unresponsive - we take the steps to do a migration. A migration involves the process taking an etcd pod that is running on a particular Kubernetes node, removing it from that node, and then finding a better node in the cluster to then place the pod onto.

Migration



Acknowledgments

We would like to thank Shefali Gautam, George Mathew, and Chegu Vinod from Dell for all their help and support during the project. Furthermore, we would like to thank our TAs Aidan Smith and Golam Muktadir, and our Professor, Richard Jullig.

Key Reasons/Ideas

Why do we care if a pod or a node is unstable/unhealthy? - Because of Raft, there are a certain number of etcd pods that need to be running in order for etcd to be able to get updates from Kubernetes. In particular if half or more of the etcd pods in a group fail, the etcd group will be in a disaster mode and no longer able to accept updates. This means that valuable client data can be lost if there are too many etcd pods that are unhealthy, or running on unhealthy nodes. As a result the migration that we perform works to avoid disaster scenarios by making sure that pods are put into the nodes that are believed to be the most healthy.

How do we determine where to place the new pod? - After deciding to migrate a pod, there is another set of decisions that need to be made on where to put a pod. The Kubernetes cluster is split onto several racks of servers, and we further partition these clusters into several domains. Then we decide to put the new etcd pod on a different rack and domain than the other pods currently running.

Results

We are very satisfied with the outcome of our project, as we learned a lot about Linux, Kubernetes, etcd, and Raft. Our script is a robust and complete solution for the issue we were presented to solve, but there are further enhancements that can be made by getting more information from Kubernetes if other groups decide to pursue this further.