

Abstract

Kubernetes is the leading open-source tool responsible for automating the deployment and management of containerized applications. One of the most important advantages of Kubernetes is "high availability" of applications. It increases software scalability and availability. Kubernetes can scale up and down applications and supporting infrastructure resources in response to changing organizational demands, enabling dynamic management of company resources. Kubernetes also enables flexibility in multi-cloud scenarios by ensuring that applications may run in either a public or private cloud.

The safest way to deploy a Kubernetes application across different regions for high-availability and disaster recovery is to create multiple-clusters in different regions. This would allow the same application to serve multiple geographical regions, improving availability & performance, disaster recovery and scaling application beyond a single cluster's limits. Multi-cluster deployment solves a range of difficulties; however, it increases the complexity of operation and maintenance as well. We will need to deploy resources and handle the clusters separately. These are the issues that Kubernetes Cluster Federation attempts to address for Multi-Cluster Kubernetes via a centralized administration interface. Kubernetes Cluster Federation is a mechanism for managing the configuration of several Kubernetes clusters using a centralized "federation control manager." Although a popular tool, there are almost no resources out there creating a Cluster Federation setup between a Central Cloud cluster acting as the host cluster and on-premises clusters running on a local machine acting as the member clusters.

This research proposes a POC where we will use a Cluster Federation tool called KubeFed to create a solution in which the functionalities and resources of a central cluster in the Cloud are automatically propagated to a local cluster on an independent machine in a separate location. These local clusters can live on small computing machines all over the world and get configurations and resources from a central cluster. End-users can make requests to local machines in nearby branches and receive the same services as those supplied by the central cluster situated further away. This way, we can propagate resources from a central cluster to local machines all around the world and decrease the end-user's latency and improve performance.