KubeEP: Aplikasi Prescale Berbasis Event untuk Kubernetes pada Lingkungan Cloud = KubeEP: Event-based Prescale Tool for Cloud-managed Kubernetes

Dipta Laksmana Baswara Dwiyantoro, author

Abstrak

Pada saat ini, pengadaan suatu event untuk menarik pengguna dilakukan oleh perusahaan. Namun, hal tersebut menyebabkan lonjakan HTTP Request pada pod yang ada pada kluster. Peristiwa tersebut menyebabkan thundering herd yang berdampak ke response time yang meningkat. Pada umumnya, terdapat Horizontal Pod Autoscaler (HPA) yang digunakan untuk mengatur jumlah pod berdasarkan kebutuhan namun waktu yang dibutuhkan cukup lama. Waktu yang lama disebabkan oleh adanya pengecekan yang dilakukan oleh Readiness dan Liveness Probe. Untuk dapat membuat kluster lebih siap menghadapi suatu event, pengembang melakukan kon gurasi ulang pada HPA sebelum event dimulai. Namun, selain diperlukan pengkon gurasian pada HPA diperlukan juga pengkon gurasian terhadap Cluster Autoscaler (CA) dikarenakan pod yang dibuat HPA memiliki kemungkinan tidak aktif jika tidak terdapat node yang tersedia untuk digunakan. Karena pengkon gurasian dilakukan dengan campur tangan manusia, maka peluang human error seperti lupa atau salah hitung dapat terjadi. Maka dari itu, dalam penelitian ini akan dikembangkan sebuah aplikasi KubeEP yang dapat melakukan penjadwalan kon gurasi HPA dan pengkalkulasian banyaknya node yang dibutuhkan oleh suatu kluster pada saat event terjadi. Dampak dari aplikasi KubeEP akan diuji dengan memberikan pengujian beban kepada kluster yang telah dijadwalkan kon gurasi HPA-nya dan telah dikalkulasikan banyak node yang dibutuhkan. Pengujian dilakukan dengan membuat lonjakan HTTP Request pada saat event mulai. Hasil pengujian didapati bahwa kluster yang dilakukan penjadwalan kon gurasi serta pengkalkulasian jumlah maksimum node memiliki performa yang lebih baik. Sementara itu, kluster yang dilakukan penjadwalan kon gurasi namun jumlah maksimum nodenya hanya 2 kali lipat dari sebelumnya memiliki performa yang lebih rendah namun masih bisa memproses HTTP Request. Pada kluster yang dilakukan penjadwalan namun jumlah maksimum node nya tidak disesuaikan lagi memiliki performa yang cukup buruk, banyak sekali HTTP Request yang gagal dan memiliki response time yang tinggi. Performa yang lebih buruk didapati pada saat kluster tidak dilakukan penjadwalan dan pengkalku-lasian jumlah maksimum node yang dibutuhkan. Berdasarkan pengujian tersebut dilakukan analisis dan didapati bahwa dampak dari penjadwalan dan pengkalkulasian yang dilakukan oleh aplikasi KubeEP memberikan efek yang signi kan pada performa dan ketersediaan kluster.

.....Currently, a company creates an event to attract many users. However, it causes HTTP Request spikes to cluster pods. HTTP Request spikes cause thundering herd and the result is an increase in response time. In general, there is a Horizontal Pod Autoscaler (HPA) used for managing pod count according to the needs but it takes quite a long time. The long time is due to a check carried out by Readiness and Liveness Probe. To make kluster more ready to handle the event, developer recon gures the HPA before the event starts. However, besides that con guration on HPA is also required con guration of Cluster Autoscaler (CA) because the pod that HPA creates might had a chance to not active if there are no nodes available to be used. Because the con guration is done by human intervention, the possibility of human error such as forgetting or miscalculation can occur. Therefore, in this research, a KubeEP application will be developed that can

perform HPA con guration scheduling and calculating the number of nodes required by a cluster when the event occurs. The impact of KubeEP application will be tested by providing load testing to a cluster that had scheduled HPA con guration and calculated the required number of nodes. Testing is done by making HTTP Request spikes when event starts. The test results found that the cluster which had scheduled con guration and calculated the required maximum number of nodes had better performance. Meanwhile, cluster which had scheduled con guration but the maximum number of nodes is doubled the amount from before had a lower performance but it still can process the HTTP Request. In cluster which had scheduled con guration but the maximum number of nodes is not changed had a bad performance, many HTTP Requests had failed and they had high response time. Worst performance found in kluster which had no scheduling and calculation of the required amount of maximum nodes. Based on the tests, an analysis was carried out and it was found that the impact of scheduling and calculations performed by the KubeEP application had a signi cant effect on the performance and availability of the cluster