ABSTRACT

Kubernetes is a container orchestration that can provision services on servers and scaling containerized applications across clusters. Kubernetes services can be accessed or exposed by clients with Cluster IP, Node Port, and Load Balancer. But on the cluster IP it can only be accessed from within the cluster, on the Node Port and Load Balancer the client must know all pod IPs and Load Balancer IPs that are exposed to the public cloud. Therefore, Ingress can be used by clients to access services from outside the Kubernetes cluster using only the domain or name ingress. Ingress is a set of rules for incoming connections for a particular service. For the ingress controller to work, it must be running in the cluster so that clients can access the services contained in containers in the Kubernetes cluster. To have an ingress controller, the user must use an existing controller. Each ingress controller has a different configuration that affects service performance. This configuration causes each ingress controller to have a different performance advantage for each variation in the number of client requests. In this study, we will compare the performance of the ingress controller using Kong and Istio. This study uses test scenarios for increasing the number of clients, namely 50, 150, 250, 350, 450, 550, 650, and 750 clients, as well as a test scenario with Istio ingress controller and Kong ingress controller. Parameters compared are delay, throughput, CPU usage, and memory usage. Based on the research results, it was found that the Istio ingress controller. The Istio ingress controller excels in throughput, CPU, and memory parameters with each result being 15.206,631 Kbps, 10,13%, and 714,733 MB. Meanwhile, the Kong ingress controller only excels in the delay parameter with a result of 0,481 ms.

Keywords: *Ingress Controller, Istio, Kong, Kontainer, Kubernetes.*