

## ABSTRACT

Databases, also known as data storage media, is an essential part of cloud services. As the amount and type of data increases, database services have the possibility of experiencing downtime. To solve this problem, we build an infrastructure that can replicate itself, so the server can be able to share the load to the replication pod to avoid the server run into downtime. A container orchestration tool called micro Kubernetes cluster which has autoscaler and high availability features, so it can replicate themselves and guarantee service availability, to avoid downtime can be applied to build the infrastructure.

This final project build a NoSQL database service, namely MongoDB. In the service will be applied a horizontal pod autoscaler feature that is able to replicate pods, and ensure the autoscaling feature runs to share the load on pods in the cluster, thereby increasing the high availability of services to avoid downtime. The autoscaling process will be tested by sending load request on the service. The experiment is done by using the Apache JMeter application to send load requests to the service.

This study focuses on the comparison of the MongoDB service that was built monolithically with the service that build with micro Kubernetes cluster, and the service with HPA features and without HPA features. Comparisons are made to find the best service. Based on Response Time, Response Code per Seconds, and CPU Usage, the results obtained are that the service built on the micro Kubernetes cluster with HPA features is the best, with a constant value of response time, below 100 ms, Response Code per Seconds reaches 500 threads per second, and CPU Usage in the range of 30 – 55%.

**Keywords :** MongoDB, Microcluster Kubernetes, HPA.