KUBERNETES CLUSTER MANAGEMENT FOR CLOUD COMPUTING PLATFORM: A SYSTEMATIC LITERATURE REVIEW

Aris Nurul Huda[1], Sri Suning Kusumawardani[2]
[1, 2] Department of Electrical and Information Engineering
Gadjah Mada University, Yogyakarta, Indonesia
e-mail: aris.n.h@mail.ugm.ac.id[1], suning@ugm.ac.id[2]

ABSTRACT

Kubernetes is designed to automate the deployment, scaling, and operation of containerized applications. With the scalability feature of Kubernetes technology, container automation processes can be implemented according to the number of concurrent users accessing them. Therefore, this research focuses on how Kubernetes as cluster management is implemented on several cloud computing platforms. Standard literature review method employing a manual search for several journals and conference proceedings. From 15 relevant studies, 5 addressed Kubernetes performance and scalability. Seven literature review addressed Kubernetes deployments. Two articles addressed Kubernetes comparison and the rest is addressed Kubernetes in IoT. Regarding the cloud computing cluster management challenges that must be overcome using Kubernetes, it is necessary to ensure that all configuration and management required for Docker containers are successfully set up on on-premises systems before deploying to the cloud or on-premises. Data from Kubernetes deployments can be leveraged to support capacity planning and design Kubernetes-based elastic applications.

***Keywords**: Kubernetes, Cluster Management, Cloud Computing, Scalability.*

MANAJEMEN KLASTER KUBERNETES PADA PLATFORM KOMPUTASI AWAN: TINJAUAN LITERATUR SISTEMATIK

Aris Nurul Huda[1], Sri Suning Kusumawardani[2]
[1, 2] Departemen Teknik Elektro dan Teknologi Informasi
Universitas Gadjah Mada, Yogyakarta, Indonesia
e-mail: aris.n.h@mail.ugm.ac.id

ABSTRAK

Kubernetes dirancang untuk mengotomatiskan penerapan, penskalaan, dan pengoperasian aplikasi dalam container. Dengan fitur skalabilitas dari teknologi Kubernetes, proses otomatisasi container dapat diimplementasikan sesuai dengan jumlah pengguna secara bersamaan yang mengaksesnya. Oleh karena itu, penelitian ini berfokus pada bagaimana Kubernetes sebagai manajemen cluster diimplementasikan pada beberapa platform cloud computing. Metode tinjauan pustaka standar menggunakan pencarian manual untuk beberapa jurnal dan prosiding konferensi. Dari 15 studi yang relevan, 5 membahas kinerja dan skalabilitas Kubernetes. Tujuh tinjauan literatur membahas penerapan Kubernetes. Dua artikel membahas perbandingan Kubernetes dan sisanya membahas Kubernetes di IoT. Mengenai tantangan manajemen klaster komputasi awan yang harus diatasi menggunakan Kubernetes, perlu untuk memastikan bahwa semua konfigurasi dan manajemen yang diperlukan untuk container Docker telah berhasil disiapkan di sistem lokal sebelum diterapkan ke cloud atau di lokasi. Data dari penerapan Kubernetes dapat dimanfaatkan untuk mendukung perencanaan kapasitas dan merancang aplikasi elastis berbasis Kubernetes.

***Kata Kunci**: Kubernetes, Manajemen Klaster, Komputasi Awan, Skalabilitas.*

I. INTRODUCTION

REDUNDANT request on the server services is a reason for using a physical server with high specifications. These specifications at least have sufficient capabilities to carry out various types of processes simultaneously. These needs require complex configurations of data, applications and networks. To simplify server resources, virtualization technology is required.

Virtualization allows a single server to function as multiple "Virtual Machines (VM)", with each virtual machine being able to operate in a different environment. Virtualization can increase the flexibility and scalability of information technology while creating significant energy and cost savings [1]. This can improve performance and availability of resources, so that the administration of information technology is easier to manage and operate [2].

Server clustering is used to reduce workloads and downtime, by allowing other servers to take over the handling of an outage. When a server experiences a service outage, the workload is distributed to the other servers before the downtime is experienced by the client. Clustering servers are generally used for applications with frequently updated data on file servers and databases.

The main reason for server clusters is protection against outages and downtime. These interruptions can be in the form of application/service failures, system/hardware failures, and site failures. Failure protection results in reduced vulnerability to risk across the network.

By sharing these resources, the workload can be reduced significantly [3]. Docker is an open-source, lightweight containerization platform that automates the application deployment process [4]. Docker containers wrap a software module in a complete file system that contains everything it needs to run: code, runtime, system tools, and system libraries. This ensures that it will always work the same, regardless of the environment in which it is run. Containers have the same resource isolation and allocation benefits as VMs, but different architectural approaches allow them to be much more portable and efficient.

Application workloads running on a container can be monitored using a system called Kubernetes. Kubernetes is an open-source platform for managing containerized applications, including managing workloads and services [5]. Kubernetes is designed to automate the deployment, scaling, and operation of containerized applications. With the scalability feature of Kubernetes technology, container automation processes can be implemented according to the number of concurrent users accessing them. This container scalability process can be applied to Kubernetes with several supporting parameters.

Kubernetes, is a system developed at Google for managing applications in containers across cluster nodes [6]. There are three main reasons Kubernetes is used for server clustering, namely availability, scalability, and reliability. Therefore, this research focuses on how Kubernetes as cluster management is implemented on several cloud computing platforms.

II. BACKGROUND STUDIES

A. Kubernetes

Kubernetes is an open source platform introduced by Google to develop scalable, reliable systems, manage workloads and containerized services through application automation-oriented APIs. The more services that are sent over the network using the API, require the available system must be reliable [7]. Service scalability can certainly increase capacity to keep up with increasing usage without redesigning the distributed system that implements the service.

Kubernetes is an infrastructure for distributed systems, which is suitable for container-based development using RESTful APIs [8, 9]. Kubernetes consists of various types of resources such as containers, nodes, and services. Kubernetes services, support, and tools have a large and rapidly growing ecosystem.

Using Kubernetes will automatically create cluster (Kubernetes Cluster). A Kubernetes Cluster consists of a set of Worker Machines (Worker Nodes), which will run applications in containers. Each Cluster has at least one Worker Node. Worker Nodes host a Pods, which are components of the application workload. The Control Plane manages the Worker Nodes and Pods in the Cluster. In a production environment, a Control Plane typically runs across multiple computers and a Cluster typically runs multiple Worker Nodes. Here is a Kubernetes cluster diagram with all its components (Fig.1).

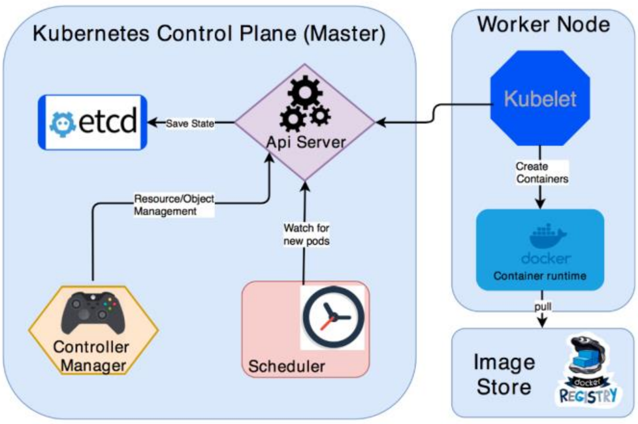

Fig.1 Kubernetes Cluster Diagram.

Control Plane has components such as: etcd (provides a reliable way to store data across machine groups), API Server (handles object validation before saving information to etcd), Scheduler (allocates what nodes the pods need to create), and Controller Manager (ensures the actual state of the system converges to the desired state, as specified in the resource specification). The components in this Control Plane are responsible for maintaining the cluster state.

The components on the Worker Node are responsible for deploying and running the application container. Each Worker Node consists of a Kubelet (monitors API Server for Pods scheduled to the Worker Node, and then starts the Pods container by instructing docker) and Docker (the runtime container used by the kubelet to run the container).

B. Cloud Computing

There have been many definitions of cloud computing. The National Institute of Standards and Technology defines it as the collection of computing resources to serve multiple users using a multitenancy model, with different physical and virtual resources dynamically assigned and moved according to user requests [10].

Cloud computing is the next stage in the evolution of the Internet. All facilities in cloud computing, whether computing, infrastructure, applications, to business processes are delivered as services that can be accessed wherever and whenever needed. These services can be tailored to the needs of users, where the cost of using the service is usually charged according to the number of resources that have been used [11, 12]. The computing paradigm in cloud computing has the advantage that the resources provided are easy to develop in real-time. Resources such as files, data, programs, hardware, and third party services can be accessed by users via a web browser.

Service or commonly called service, is a task that has been packaged, so that it can be automated and delivered to customers consistently and repeatedly. The cloud computing service model is divided into three types, namely: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Infrastructure as a Service offers the storage and compute resources that developers and organizations use. Platform as a Service offers a development environment that organizations can use to create business applications. Software as a Service offers business application services that are tailor-made for organizations. All service models require management and administration (including security), as depicted in the outer box in Fig.2.

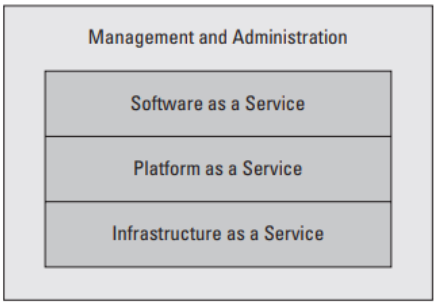

Fig.2 Service Model in Cloud Computing.

Infrastructure as a Service (IaaS) is the provision of computer hardware (servers, network technology, storage, and data centers) as a service. This includes operating systems and virtualization technologies for managing resources. Scaling services are dynamic, so customers can easily customize the infrastructure according to the resources needed.

Platform as a Service (PaaS) provides an integrated set of software developers need to build applications. PaaS is an evolution of web hosting, which provides software for developing websites. The main benefit of having PaaS is the ability to develop and deploy software that is completely cloud-based. Every aspect of software development, from the design phase, to source code management, testing, to deployment is all in the cloud.

The implementation of the cloud service is Software as a Service (SaaS), which is a business application hosted by the provider and delivered as a service. Before this type of service emerged, companies spent a lot of resources implementing and customizing applications to meet internal business needs. Not only are they difficult to implement, they are also difficult to learn and use. Customer Relationship Management (CRM) is one of the most common categories of software as a service.

The management and administration section of cloud computing services explains how to integrate services into an organization's information technology. The cloud must be designed to manage customer demand provisioning, workload management, security, monitoring, and billing services. For cloud services to be safe and effective, performance measurement and monitoring must be carried out.

The development model (deployment) in cloud computing is divided into 4 models, namely: private cloud, community cloud, public cloud, and hybrid cloud. A private cloud provides the infrastructure for use in a single organization of multiple consumers (for example, business units). The community cloud provides the infrastructure for deployment for a specific consumer community of organizations that share a common purpose. A public cloud is provided for open use by the general public, while a hybrid cloud is a composition of two or more different cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standard or proprietary technologies that enable data access and applications together.

III. RESEARCH METHODOLOGY

The study was carried out through a literature review as suggested by Kitchenham [13], include: research questions, literature search process, inclusion and exclusion criteria, data collection, data analysis.

A. Research Questions

To identify the Kubernetes cluster management for cloud computing platforms, Table I shows list of research questions and the motivations to rise such questions.

TABLE I. RESEARCH QUESTIONS AND MOTIVATIONS

No.	Questions	Motivations
1	What research topics are being addressed?	This study helps researchers understand the use of Kubernetes in various fields
2	What are the challenges in Kubernetes cluster management	Kubernetes Cluster management helps to managing server in allocating client requests
3	What are the research opportunities in cluster management	This article helps researchers to know current state and future research improvement

B. Data Sources

The literature used in this article are proceedings and published journal papers since 2016 in several sources. The selected journal publishers are shown in Table II. These publishers were selected because they have been used as sources for a literature review related to Kubernetes cluster management.

TABLE II. SELECTED PUBLISHERS

Publishers	URL
IEEE	https://ieeexplore.ieee.org
Springer	https://link.springer.com
Science Direct	https://sciencedirect.com

C. Search Strategy

Kubernetes came into light from 2014, as this technology closely related to cloud computing. Significant research was not done for cluster management to improve server scalability, so this research considered for review from past five years, from 2016. Based on the topic and the proposed research questions, then terms are made to maximize searching phase.

Terms “Kubernetes cluster management”, “cloud computing”, and “scalability” considered as primary keyword. Logical operators “OR” and “AND” used for associating the significant results. After a few tests, these terms give the adequate result. Table III shows list of strings and keywords used for searching process and framed the search string.

TABLE III. LIST OF STRINGS AND KEYWORDS

String	Batch1 (B1)	Batch2 (B2)
String1 (S1)	Kubernetes	Cluster Management
String2 (S2)	Cloud Computing	Scalability

Search String: (([S1, B1] AND [S1, B2]) OR ([S2, B1] AND [S1, B2]) OR ([S1, B1] AND [S2, B2]) AND ([S2, B1] AND [S2, B2])

D. *Article Selection Process*

The process of selecting articles begins with compiling research questions. After that, the search and selection of articles are carried out by framing the strings. The articles selected are those published in English. After finding the appropriate article, a study was conducted on Kubernetes cluster management on the cloud computing platform in each study. Fig.3 is an example of the article filtering process from the ScienceDirect datasource.

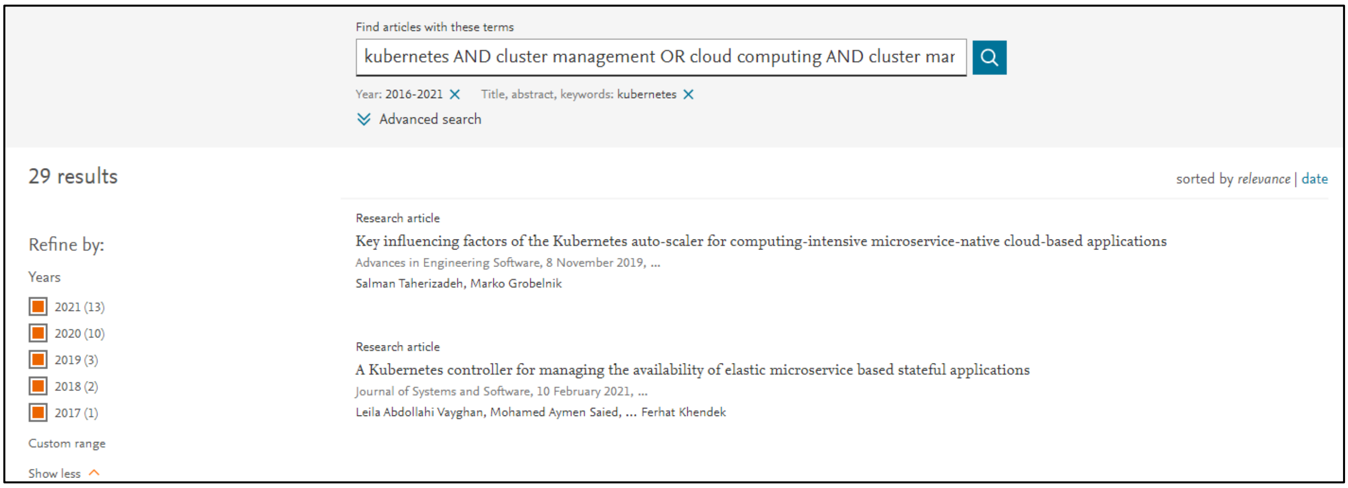

Fig.3 Articles Selection Process in ScienceDirect

The search process was ended by categorizing the cloud computing platform as a whole to ensure the completeness of this research. Most articles were screened because their titles did not match the determination criteria, or otherwise, abstracts were not identified to be considered in this search.

IV. Results

The flow diagram for scoping review as shown in Fig.4. The articles were identified using framed keywords, resulting 155 articles from the publishers mentioned in Table II. The articles were published between 2016 and 2021.

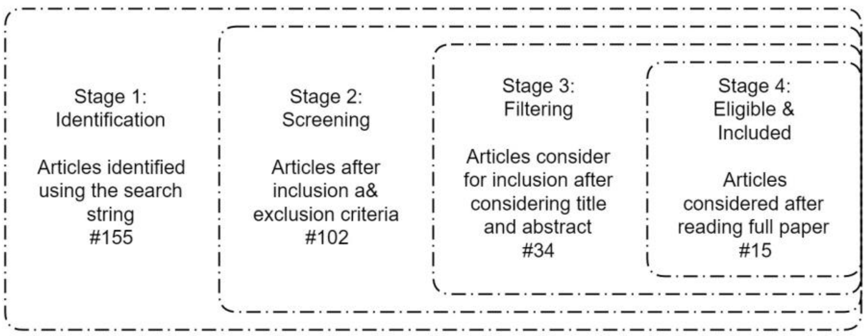

Fig.4 Articles Selection Process for Review.

To select significant related studies, inclusion and exclusion criteria were included in the search. Details of the inclusion and exclusion criteria are shown in Table IV. The results of the search for articles showed 102 articles.

TABLE IV. INCLUSION AND EXCLUSION CRITERIA

Inclusion Criteria	Exclusion Criteria
The study focuses on Kubernetes	The study that focuses on other issues in Kubernetes
The articles written only in English	The articles not written in English are not considered
The articles which are published by the above-mentioned publishers and journals	The articles which are not published and not peer-reviewed are not considered
The article's type are conference paper or journal article	The articles are not in conference paper type or journal article
The articles published in well-reputed indexed journals and conferences	The articles from keynote speeches and editorial

Based on the titles and abstracts of the articles selected in the inclusion and exclusion selection process, the number of articles selected was summarized to 34. So that 34 articles were fully examined and studied based on content that matched the categorization of Kubernetes cluster management on the cloud computing platform. Kubernetes cluster management generated 15 articles based on their final content (Table V).

TABLE V. LITERATURE REVIEW STUDY

Author	Year	Research Field	Article Type
Marathe, et all. [14]	2019	Kubernetes Comparison	Conference Paper
Muddinagiri, et all. [15]	2019	Kubernetes Deployment	Conference Paper
Truyen, et all. [16]	2020	Kubernetes Comparison	Journal Article
Dewi, et all. [17]	2019	Kubernetes Deployment	Conference Paper
Yang, et all. [18]	2020	Kubernetes Deployment	Journal Article
Truyen, et all. [19]	2018	Kubernetes Deployment	Conference Paper
Fayos-Jordan, et all. [20]	2020	Kubernetes IoT	Journal Article
Toka, et all. [21]	2020	Kubernetes Performance and Scalability	Conference Paper
Oh, et all. [22]	2018	Kubernetes Deployment	Conference Paper
Mohamed, et all. [23]	2019	Kubernetes Deployment	Journal Article
Diouf, et all. [24]	2020	Kubernetes Performance and Scalability	Journal Article
Medel, et all. [25]	2018	Kubernetes Performance and Scalability	Journal Article
Lei, et all. [26]	2019	Kubernetes Performance and Scalability	Conference Paper
Taherizadeh, et all. [27]	2020	Kubernetes Performance and Scalability	Journal Article
Rossi, et all. [28]	2020	Kubernetes Deployment	Journal Article

Based on defined criteria, essential research articles are selected after comprehensive matching of titles, abstracts, and published research to ensure that the research results truely relate to the current research (Fig. 2.).

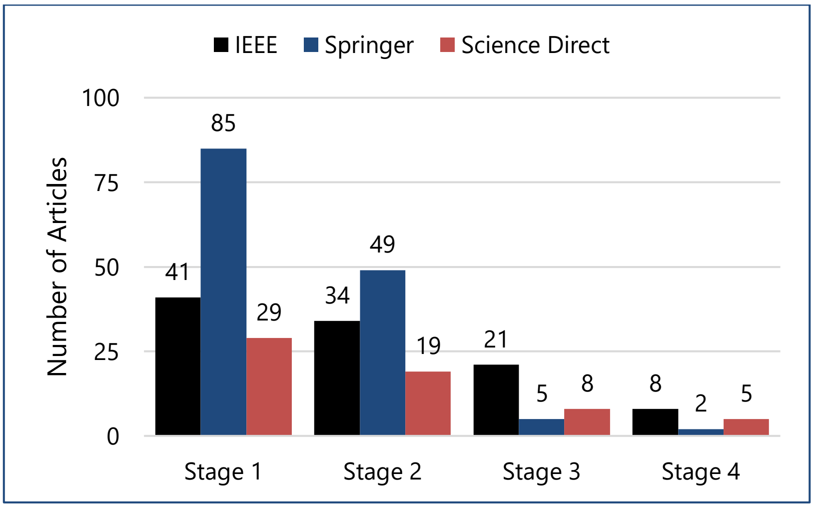

Fig.5 Articles Selected for Review.

V. DISCUSSION

In this section, the literature review discusses the following facts and answer the research question.

A. What research topics are being addressed?

Total of 15 selected papers were grouped according to their respective research fields. The results of the categorization of the research field are as follows:

1) Five papers are related to Kubernetes performance and scalability. Discussions on performance are mostly focused on evaluating performance, down to the Kubernetes architecture itself. Scalability has a lot to say about the methods used for Kubernetes auto-scaling.

2) Kubernetes deployment is discussed in 7 selected papers. Deployment of Kubernetes mostly aims to address application workload and computing resources issues. To achieve these objectives configurations and management is required so the Docker container is successfully set.

3) A paper is comparing Kubernetes with another Docker container and the other paper compares Kubernetes vendor to detect desired feature in any Kubernetes vendor. This study also reconfiguring the studied vendors with missing features in a uniform manner.

4) Research field in IoT Kubernetes found in a research paper to run cluster of commercial Small Board Computers (SBC) devices.

B. What are the challenges in Kubernetes cluster management

The following are the challenges of cloud computing cluster management which are to be addressed using Kubernetes.

1) Ensure all the configurations and management needed for a Docker container is set successfully on the local system before it is deployed onto the cloud or on the premise. [15]

2) Using fog computing with low-cost SBC devices in a context of IoT. [20]

3) Checkpoint-based approach of stateful container migration for orchestrated container clusters that are running in the containers with CRIU (Checkpoint/Restore in user space) feature. [22]

4) Characterized using data from a Kubernetes deployment, that can be exploited for supporting capacity planning and designing Kubernetes based elastic applications. [25]

C. What are the research opportunities in cluster management

Kubernetes has been developed a lot recently, with its implementation in various fields, such as banking, insurance, manufacturing, health care, government, transportation, education, customer service, and so on. Based on the results of previous research, the following are some opportunities that can be developed in the future by using Kubernetes to solve certain problems:

1) As mentioned in [15], the docker container using Kubernetes is very useful in finance and health care fields. Customer data or patient data, are private and carries a high security risk to be stored in to cloud. However, by using MiniKube, servers can implement applications in a container which can then be easily scaled to run for the entire organization. In addition, local testing methods before deploying to the cloud can be very useful for organizations. Thus, other fields have ample opportunities to be developed as well.

2) The prototype implementation focused on verifying the PoC level in [22] is still susceptible to hardware downtime and lost network connections. There are various issues that increase the problem and agility that can be adopted to overcome this limitation. For example, adapting an iterative migration for container migration, with additional variations adopting active and standby containers.

3) There is a set of key factors which should be considered in the development of auto-scaling methods under different workload conditions [27]. Kubernetes auto scaler can be extended to also offer vertical auto scaling for the allocation of disk and bandwidth resources to storage intensive and network intensive services, respectively. the combination of horizontal and vertical auto scaling mechanisms can be exploited to the same service.

This review only considers a limited number of databases, journals, and conferences. The keywords (strings) that have been used to search the literature are also been defined. Articles published before 2016 were not subject to this study. This review focuses only on cluster management issues and not on other related issues in Kubernetes.

VI. Conclusion

This research aims to know how Kubernetes as cluster management is implemented on several cloud computing platforms. Discussions on performance are mostly focused on evaluating performance, down to the Kubernetes architecture itself. Deployment of Kubernetes mostly aims to address application workload and computing resources issues. To achieve these objectives configurations and management is required so the Docker container is successfully set. The challenges of cloud computing cluster management which using Kubernetes are that the configurations and management of Docker container needs for successfully set before it is deployed onto the cloud. Some opportunities that can be developed in the future by using Kubernetes are servers can implement applications in a container which can then be easily scaled to run for the entire organization. There is a set of key factors which should be considered in the development of auto-scaling methods under different workload conditions. Kubernetes auto scaler can be extended to also offer vertical auto scaling for the allocation of disk and bandwidth resources to storage intensive and network intensive services, respectively.

REFERENCES

[1] Amirullah, R. M. Ijtihadie and H. Studiawan, "Optimasi Daya Data Center Cloud Computing Pada Workload High Performance Computing (HPC) Dengan Scheduling Prediktif Secara Realtime," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 15, no. 1, pp. 1-10, 2017.

[2] S. Dutta and I. Mia, "The global information technology report 2009-2010," in *World Economic Forum and INSEAD*, SRO-Kundig Geneva, Switzerland, 2010.

[3] IBM Cloud Education, "Containerization," 15 May 2019. [Online]. Available: https://www.ibm.com/cloud/learn/containerization. [Accessed 2 April 2021].

[4] A. Avram, "Docker: Automated and Consistent Software Deployments," 27 March 2013. [Online]. Available: https://www.infoq.com/news/2013/03/Docker/. [Accessed 2 April 2021].

[5] Kubernetes, "What is Kubernetes?," 26 September 2020. [Online]. Available: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/. [Accessed 1 April 2021].

[6] D. Marinescu, "Computer Clouds," in *Complex Systems and Clouds*, Massachusetts, Morgan Kaufmann, 2017, pp. 113-145.

[7] B. Burns, B. Grant, D. Oppenheimer, E. Brewer and J. Wilkes, "Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade," *Queue*, vol. 14, no. 1, pp. 70-93, 2016.

[8] E. Truyen, N. Kratzke, D. V. Landuyt, B. Lagaisse and W. Joosen, "Managing Feature Compatibility in Kubernetes: Vendor Comparison and Analysis," *IEEE Access*, vol. 8, pp. 228420-439, 2020.

[9] S. Baset, S. Suneja, N. Bila, O. Tuncer and C. Isci, "Usable declarative configuration specification and validation for applications, systems, and cloud," in *Middleware Conference: Industrial Track*, Las Vegas, 2017.

[10] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Gaithersburg, 2011.

[11] J. Hurwitz, R. Bloor, M. Kaufman and F. Halper, Cloud Computing for Dummies, Indiana: Wiley Publishing, 2010.

[12] K. Xiong and H. Perros, "Service performance and analysis in cloud computing," in *2009 Congress on Services - I*, Los Angeles, 2009.

[13] B. Kitchenham, "Procedures for Undertaking Systematic Reviews," Computer Science Department, Keele University, Australia, 2004.

[14] N. Marathe, A. Gandhi and J. M. Shah, "Docker Swarm and Kubernetes in Cloud Computing Environment," in *International Conference on Trends in Electronics and Informatics*, India, 2019.

[15] R. Muddinagiri, S. Ambavane and S. Bayas, "Self-Hosted Kubernetes: Deploying Docker Containers Locally With Minikube," in *International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)*, India, 2019.

[16] E. Truyen, N. Kratzke, D. V. Landuyt, B. Lagaisse and W. Joosen, "Managing Feature Compatibility in Kubernetes: Vendor Comparison and Analysis," *IEEE Access*, vol. 8, pp. 228420-228439, 2020.

[17] L. P. Dewi, A. Noertjahyana, H. N. Palit and K. Yedutun, "Server Scalability Using Kubernetes," in *Technology Innovation Management and Engineering Science International Conference*, Bangkok, 2019.

[18] D. Yang, D. Wang, D. Yang, Q. Dong, YeWang, H. Zhou, Daocheng and Hong, "DevOps in Practice for Education Management Information System at ECNU," *Procedia Computer Science*, vol. 176, pp. 1382-1391, 2020.

[19] E. Truyen, M. Bruzek, D. V. Landuyt, B. Lagaisse and W. Joosen, "Evaluation of Container Orchestration Systems for Deploying and Managing NoSQL Database Clusters," in *Cloud Computing (CLOUD)*, Seattle, 2018.

[20] R. Fayos-Jordan, S. Felici-Castell, J. Segura-Garcia, J. Lopez-Ballester and M. Cobos, "Performance comparison of container orchestration platforms with low cost devices in the fog, assisting Internet of Things applications," *Journal of Network and Computer Applications*, vol. 169, pp. 1-13, 2020.

[21] L. Toka, G. Dobreff, B. Fodor and B. Sonkoly, "Adaptive AI-based auto-scaling for Kubernetes," in *Cluster, Cloud and Internet Computing (CCGRID)*, Melbourne, 2020.

[22] S. Oh and J. Kim, "Stateful Container Migration employing Checkpoint-based Restoration for Orchestrated Container Clusters," in *International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, 2018.

[23] M. Mohamed, R. Engel, A. Warke, S. Berman and H. Ludwig, "Extensible persistence as a service for containers," *Future Generation Computer Systems*, vol. 97, pp. 10-20, 2019.

[24] G. M. Diouf, H. Elbiaze and W. Jaafar, "On Byzantine fault tolerance in multi-master Kubernetes clusters," *Future Generation Computer Systems*, vol. 109, pp. 407-419, 2020.

[25] V. Medel, R. Tolosana-Calasanz, J. Á. Bañares, U. Arronategui and O. F. Rana, "Characterising resource management performance in Kubernetes," *Computers and Electrical Engineering*, vol. 68, pp. 286-297, 2018.

[26] Q. Lei, W. Liao, Y. Jiang, M. Yang and H. Li, "Performance and Scalability Testing Strategy Based on Kubemark," in *Cloud Computing and Big Data Analytics (ICCCBDA)*, Suzhou, 2019.

[27] S. Taherizadeh and M. Grobelnik, "Key influencing factors of the Kubernetes auto-scaler for computing-intensive microservice-native cloud-based applications," *Advances in Engineering Software,* vol. 140, no. 1, pp. 1-11, 2020.

[28] F. Rossi, V. Cardellini, F. L. Presti and M. Nardelli, "Geo-distributed efficient deployment of containers with Kubernetes," *Computer Communications*, vol. 159, pp. 161-174, 2020.