VANJA KORAĆ
Mathematical Institute
Serbian Academy of Sciences and Arts
Belgrade, Serbia
E-mail: vanja@mi.sanu.ac.rs

Received: May 27th 2023
Accepted: November 30th 2023
Review article
UDC: 004.424.75:902/904
https://doi.org/10.18485/arhe_apn.2023.19.10

ZORAN DAVIDOVAC
K16 PRO, d.o.o.
Belgrade, Serbia

KUBERNETES AS AN APPLICATION MANAGEMENT PLATFORM

ABSTRACT

Kubernetes is a technology used to manage the deployment and scaling of applications in the cloud. It can enable automatic scaling of applications across multiple servers in order to ensure the applications are always running on the optimal number of resources. In addition, high availability is ensured by automatically redirecting traffic between servers. It could be useful in archaeology to manage the deployment of various applications for processing and analysing data from archaeological research. Kubernetes can improve application performance by making it easier to develop, test, and maintain containerised applications.

KEYWORDS: K8s, KUBERNETES, CLOUD, CONTAINER, APPLICATION MANAGEMENT PLATFORM.

INTRODUCTION

Kubernetes is one of the most important technologies today for managing applications and infrastructure. This open-source container orchestration system enables automatic deployment, scaling, resource management, and self-healing of applications, allowing focus on application development rather than infrastructure management. Kubernetes is often presented in the literature as “K8s” and actually represents a platform for managing and automating the deployment, scaling and management of applications. Originally designed by Google, it is now maintained by the Cloud Native Computing Foundation. It is commonly applied within an infrastructure as a service (IaaS), where it is used to automate processes associated with the deployment and management of containers, enabling easier development and maintenance of applications. Kubernetes can be deployed in many different situations, from small single applications to large systems with hundreds or even thousands of containers. Kubernetes is used worldwide in many different industries, including finance, technology, telecommunications, manufacturing, and more.

One of the most common ways of using Kubernetes is to automate procedures linked to the deployment and management of applications within one or more computer clusters. The ability to automatically scale applications across multiple servers also enables automatic traffic redirection between servers to ensure high availability. Kubernetes can improve application performance by enabling easier development, testing and maintenance, and scaling of containerised applications.

As an open-source system, it can be used on different computers in the cloud or on other physical infrastructures. Kubernetes can be applied in various scenarios, including the development and maintenance of large web applications, container management, testing and production. One example of using Kubernetes is to harness its capabilities to manage the deployment of containerised applications in large web services. This kind of implementation allows for an easier scaling of applications depending on changes in resource requirements, as well as a better utilisation of resources in case of load changes.

WHAT IS REQUIRED TO USE KUBERNETES?

In order to use Kubernetes, it is necessary to have at least one server running on the Linux operating system. Additionally, it is necessary to have experience in working with containers, as well as to have knowledge about networking, security and infrastructure. In practice, there are certain things necessary for the use of Kubernetes (Kubernetes 2023) (Burns, Beda and Hightower 2019):

- Infrastructure management system: Kubernetes can be used on a physical or a virtual system, so having an infrastructure management system that will support Kubernetes is essential. It can be a physical machine, a virtual machine, or a public cloud service, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure.
- Container management system: Kubernetes consists of a series of containers hosted on one or more servers. It is necessary to have a server to manage these containers and to ensure that they work together.
- Containers: Kubernetes uses containers to package applications and their dependencies. It is necessary to have ready-to-use containers in a Kubernetes environment. Containers can be system-based and application-based[1].

Nodes – computers on which the containers will run. Nodes must have some container manager (e.g., Docker) and Kubernetes agents installed (Kubeadm[2], Kubelet[3] and Kubectl[4]). A node can be a physical or a virtual machine and can be hosted on-site or in the cloud. A Kubernetes cluster can have a large number of nodes, and with the latest versions supporting up to 5,000 (Shtein 2023). A Kubernetes node is actually a single machine in a cluster that serves as an abstraction. Instead of managing specific physical or virtual machines, each node is treated as a pooled CPU and RAM resource on which to run containerised workloads. When an application is deployed to a cluster, Kubernetes distributes the work across the nodes. Workloads can be seamlessly moved between nodes in a cluster (Shtein 2023).

Kubectl is a command line interface used to control the cluster (Saito, Chloe Lee and Carol Hsu 2018: 10).

A kubelet is a master process on a Kubernetes node that communicates with the Kubernetes master node to handle the following operations (Saito, Chloe Lee and Carol Hsu 2018: 12):

- periodically accesses the API Controller for verification and reporting;
- performs container operations;
- runs an HTTP server to provide simple API.

[~]$kubectl get nodes -o=wide **(see Table 1)**

Kubernetes master node: Master node manages the Kubernetes cluster and is used to manage the data and configurations of the containers in the cluster.

Kubernetes worker nodes: Worker nodes are machines available to run containers in the cluster.

Kubernetes API: The Kubernetes API server

[1] System containers seek to emulate virtual machines and often initiate a full boot process. They often include a set of system services typically found in a VM, such as ssh, cron, and syslog. When Docker was new, these types of containers were much more common. Over time, they were considered bad practice, and application containers gained favour. Application containers differ from system containers in that they typically run a single program. While running one program per container may seem like an unnecessary limitation, it provides the perfect level of granularity for building scalable applications (Burns, Beda and Hightower 2019).

[2] Kubeadm is a command-line tool that simplifies the process of setting up and managing Kubernetes clusters.

[3] Kubelet is a technology that deploys, creates, updates and removes containers on a Kubernetes node.

[4] Kubernetes CLI tool (Kubectl) – this tool allows you to manage a Kubernetes cluster via a command line.

[~]$kubectl get nodes -o=wide

NAME	STATUS	ROLES	AGE	VER	INT-IP	EXT-IP	OS-IMG	KERNEL-VER	CONTAINER-RUNTIME
vmss000002	Ready	agent	21d	v1.22.11	10.163.4.62	<none>	Ubuntu 18.04.6 LTS	5.4.0-1090-azure	containerd://1.5.11 +azure-2
vmss000003	Ready	agent	11d	v1.22.11	10.163.4.4	<none>	Ubuntu 18.04.6 LTS	5.4.0-1090-azure	containerd://1.5.11 +azure-2

Table 1. Information about nodes

allows programs to communicate with the Kubernetes master node and to manage containers in the cluster. The API server actually represents the Kubernetes central station. All communication between all components must go through the API server, which means that all internal system components, as well as external user components, communicate through the same API. In it, the desired state of the application is defined through the configuration. This includes: choosing the container image to use, choosing open ports and the number of Pod replicas to run. All requests to the API server are subject to authentication and authorisation checks, but once these are done, the configuration YAML database becomes validated, included in the cluster repository, and deployed to the cluster (Poulton and Joglekar 2020: 2020: 14)

K8s - ADDITIONAL COMPONENTS AND SOME COMMANDS

Kubernetes then has a whole range of components, including PODs, services, volumes, Scheduler and other components that are necessary for the functioning of Kubernetes, as well as a Kubernetes dashboard tool that allows you to visually monitor and manage a Kubernetes cluster through a web interface. A Kubernetes POD is the smallest management unit in a Kubernetes cluster. A pod includes one or more containers, and operators can attach additional resources to the pod, such as storage volumes. A pod has its own IP, allowing pods to communicate with other pods on the same node or other nodes (Shtein 2023).

[~]$ kubectl get pods -o=wide **(see Table 2)**

[~]$kubectl get nodes -o=wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
agent	1/1	Running	0	5d19h	10.163.4.6	vmss000003
debug	1/1	Running	1	5d1h	10.163.4.10	vmss000003
eureka-0	1/1	Running	0	11d	10.163.4.73	vmss000002
kowl-ui	1/1	Running	0	4d21h	10.163.4.23	vmss000003
notifications	1/1	Running	0	4d20h	10.163.4.12	vmss000003

Table 2. Information about PODs

This command does not display all PODs located on the system. There are PODs that are part of the system itself. It often happens that ISP or cloud providers add their own pods, which are poorly or even not documented. In order to see Azure specific PODs, the following command is used: [~]$kubectl get pods -o=wide -A .

Kubernetes Scheduler runs on the master node and is responsible for searching for eligible worker nodes for each POD and placing it on those nodes. Each POD has a template that defines how many instances of the module should be run and on which types of nodes. If a node fails or does not have enough resources to run a module, the module is dropped and restarted on another node (Shtein 2023).

Services are also important components of Kubernetes. Care should be taken here regarding terminology, since there some terms that intersect, for example, when someone says service (this means Software as a service (or SaaS) or the port on which the application will work). In this sense, a service can mean a set of PODs that is necessary for some SaaS. Therefore, in Kubernetes terminology, it can be said that services represent services for the logical grouping of a set of PODs, which is necessary to provide network connectivity (Microsoft Learn 2023). The following service types are available (Microsoft Learn 2023): ClusterIP, NodePort, LoadBalancer and ExternalName. According to Google, there is also a fifth, the so-called Headless (Google Kubernetes Engine (GKE) 2023).

With the ClusterIP service, internal clients send requests to a stable internal IP address. With the NodPort service, clients send requests to the IP address of the node at one or more NodePort values specified by the service. The NodePort type is an extension of the ClusterIP type. Hence, a service of the NodePort type has a cluster IP address.

With the LoadBalancer service, clients send requests to the IP address of the network load balancer. The LoadBalancer type is an extension of the NodePort type. So, a service of the LoadBalancer type has a cluster IP address and one or more NodePort values. With the ExternalName service, internal clients use the service's DNS name as an alias for the external DNS name. When it comes to the Headless service (only with Google), this type of service can be used when you want to group PODs, but you do not need a stable IP address.

In Kubernetes, there are several different port configurations for Kubernetes services (Merron 2020):

- Port – exposes a Kubernetes service on the specified port within the cluster. Other PODs within the cluster can communicate with this server on the specified port.
- TargetPort is the port to which the service will send requests through which your module will listen. Your application in the container will also need to listen on this port.
- NodePort externally exposes the service to the cluster using the IP address of the target nodes and the NodePort. NodePort is the default setting if the port field is not specified.

BASIC STEPS FOR CONFIGURING KUBERNETES

Configuring Kubernetes involves several steps, including installing a Kubernetes cluster, configuring the network, setting up security, and installing additional components.

The first step is installing a Kubernetes cluster. This step involves installing the Kubernetes master and worker nodes. Kubernetes master controls the entire cluster, while worker nodes work with applications. Configuration files for various Kubernetes components (e.g., API server, Scheduler, etc.) are set there.

The second step is to configure the network. It is necessary to set up the network so that applications and containers can communicate with each other and with the outside world. This is also where the options for scaling and maintaining the cluster are set (e.g., resource management, container allocation, etc.).

Kubernetes can monitor PODs and scale them when CPU utilisation or some other metric exceeds a threshold. The autoscaling resource specifies the details (CPU percentage, how often to check) and the appropriate autoscaling controller adjusts the number of replicas, if necessary (Sayfan 2017: 215) (**Figure 1**).

The third step is to set the security. It is necessary to ensure the security of the Kubernetes cluster, by using authentication, authorisation and

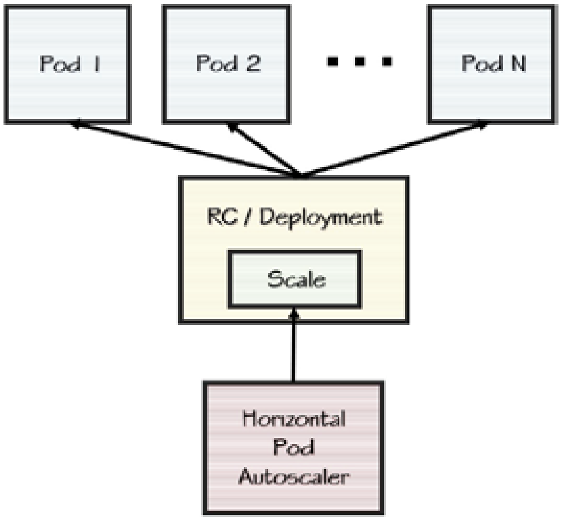

Figure 1. Horizontal POD autoscaling (figure made according to Sayfan 2017: 216).

encryption of data and other security measures. In order to increase security when using Kubernetes, there are several important actions that should be taken. Firstly, it is necessary to ensure that Kubernetes is installed with security settings according to the manufacturer's recommendations. This includes using security keys to authenticate and encrypt data, as well as ensuring that all network connections are secured. Secondly, ensure that all applications and services deployed through Kubernetes are regularly updated to prevent vulnerabilities. This can be achieved through automated updating and patching processes. Thirdly, it is necessary to ensure that users and resources within the Kubernetes environment are controlled according to the principle of "least privilege". This means that users are granted only those rights that are necessary for them to perform their tasks, and all other privileges are disabled. Additionally, it is necessary to implement controls for monitoring and auditing work within Kubernetes.

Implementing monitoring and auditing controls in a Kubernetes environment is important for several reasons. On the one hand, it allows monitoring of the operation of applications in real time and reacting immediately to possible problems or errors. On the other hand, it is possible to monitor the correct operation of the applications (in the sense of whether they behave in accordance with the set goals and expectations). This way, the auditing of the application operation is performed, which is important for achieving a high level of security and reliability in the operation of applications. Implementation of control for monitoring and auditing work in Kubernetes environment can be done in different ways, depending on specific requirements and needs. However, this is mostly done using different tools and technologies, such as Prometheus[5] (Wilson 2023), Fluentd (Fluentd 2019)[6], Elasticsearch (Jetha 2020)[7] and Kibana (Elastic 2023).

The fourth step is to install additional components. This step involves installing additional components, such as ingress controllers, volume managers, monitoring and others. These components allow the functionality of Kubernetes to be improved and adapted to the specified needs.

Configuring Kubernetes involves setting various options and parameters to enable Kubernetes to work properly and to be tailored to the needs of an organisation. After configuring the Kubernetes cluster, it is possible to use various tools and services to manage and develop applications on the cluster. This includes working with the Kubernetes API, creating and managing containers, and creating and maintaining Deployments and other objects in the cluster.

When it comes to technologies related to container management, Docker can also be found in the literature in addition to Kubernetes. Although both technologies are related to container management, there are some fundamental differences between them. Docker was originally a container platform that allowed development teams to package applications into containers and distribute them easily.

Kubernetes, on the other hand, is a platform for automating the deployment, scaling and management of containers on clusters of computers. Kubernetes allows development teams to deploy applications to different computers in a cluster, as well as to automatically adjust the number of application instances depending on the load. It can be said that Docker serves as a tool for packaging ap-

[5] Prometheus is a highly scalable open-source monitoring framework. It provides out-of-the-box monitoring capabilities for the Kubernetes container orchestration platform.
[6] Fluentd is a logging agent that takes care of collecting, parsing and distributing logs.
[7] Elasticsearch is typically used to index and search large amounts of log data, but it can also be used to search many different types of documents.

plications into containers, while Kubernetes serves as a tool for managing containers on computer clusters. Kubernetes is, in fact, integrated with Docker and can be used to manage containers that are created in Docker. Therefore, if you need to automate the management of applications in containers on multiple computers, Kubernetes could be a good choice, while Docker could be a good choice if you need to simply pack an application into a container.

USE OF KUBERNETES

Kubernetes offers efficiency in a number of ways. Firstly, it enables the easier management of large numbers of containers and automates many of the routine tasks required to maintain applications. This includes load balancing, resource allocation and scaling applications according to needs. Secondly, Kubernetes makes it easier to maintain applications in production. This means the ability to start and stop applications quickly, to maintain them easily, and to develop new versions more easily, without interruption. Together, this can increase efficiency and reduce the time required to maintain applications in production.

Kubernetes is designed to help developers and administrators build, deploy and manage distributed systems at scale, and provides a range of features and tools to help with this (Kubernetes 2023), including (Burns, Beda and Hightower 2019):

- Automatic deployment and scaling of applications, based on declarative configuration files.
- Self-healing capabilities, for automatic replacement of faulty or defective containers (Lukša 2020).
- Service discovery and load balancing to make it easier for containers to communicate with each other and access external resources.
- Storage orchestration, to automatically manage storage resources used by containers.
- Rollout and rollback, to facilitate application updates without downtime.
- Multi-tenancy and resource allocation, in order to run multiple workloads on the same cluster and allocate resources to them as needed.

The usefulness of Kubernetes can be seen in the case when there is an application that consists of multiple containers and a reliable and efficient way to manage those containers is needed. It is Kubernetes that enables the automation of many tasks related to container management, including the deployment and scaling of containers, maintaining the security and availability of the application, and maintaining the correctness of the application in case of errors (Lukša 2020).

Kubernetes is used in various industries and situations where a large number of computers and applications need to be managed. This includes managing servers in large organisations, as well as in the cloud and on-site systems. It is also used to develop and manage large and distributed data processing systems, as well as to manage applications in various environments, including those used in IoT devices. Kubernetes is also applied to manage and automate work with various resources in cloud environments, including containers, virtual machines, databases and other infrastructure solutions. It should be mentioned that Kubernetes is already used by all major cloud vendors like AWS, Azure, and Oracle Cloud Infrastructure. They all have their own services for managing Kubernetes clusters on their platforms. In AWS (AWS) it is called Amazon Elastic Kubernetes Service (EKS), in Azure (Azure) it is called Azure Kubernetes Service (AKS) and in Oracle Cloud Infrastructure (OCI), its name is Oracle Kubernetes Engine (OKE).

Kubernetes is additionally used in the development and maintenance of various types of applications, from simple web applications to complex micro services and server-less solutions. Each micro service is a separate application with its own development cycle. When a system consists of many micro services, automated management is crucial. Kubernetes provides exactly this automation. The features it offers ensure that the task of managing hundreds of micro services is almost trivial (Lukša 2020). Kubernetes will enable faster and more efficient application development, as well as easier maintenance and scaling of applications according to needs.

Kubernetes and archaeology

Kubernetes is a technology used to manage the deployment and scaling of applications in the cloud as well. It could be useful in archaeology to manage the deployment of various applications for processing and analysing data from archaeological surveys. Archaeological surveys often produce large amounts of data, including 3D models, geographic information, photographs, records and other relevant information. For example, applications processing photos or geolocation data can be deployed over Kubernetes to ensure that they are always available and capable of processing large amounts of data. Kubernetes can be used to manage all that data, as well as to facilitate the deployment and sharing of data with relevant researchers, archives and other organisations. However, data security should be taken into account in such a situation. One of the examples of the use of Kubernetes technology in the management of archaeological sites is the creation of an application (MyPompeii) (Pompeii Sites) by the Archaeological Park of Pompeii during the Covid-19 pandemic, which was used to help tourists adhere to social distancing protocols through a real-time map (Gutierrez 2021).

Another example of its application in archaeology could be the use of Kubernetes to manage archaeological simulations and models. Archaeological models are complex and require a large amount of computing power to run them. Kubernetes can be used to automatically scale the computing power required to run these models, as well as to manage the containers required to run the simulation.

Considering methods to store archaeological data using systems like Kubernetes can be tailored to the specific needs of archaeological research. Here are some of the methods used and details on how to implement this process:

Utilizing Microservices Architecture: Kubernetes enables the implementation of microservices, meaning different aspects of archaeological data can be stored in separate services. For example, you could have a microservice for storing information about sites, another for artifacts, and so on. This facilitates system scalability and maintenance.

Using Object Storage Resources: Instead of traditional databases, you could explore the use of Object Storage resources for storing large amounts of archaeological data, including images, scans, and other multimedia records. Object Storage can be efficient for storing unstructured data.

Implementing Blockchain Technology for Auditing: Consider the use of blockchain technology for storing an audit trail of changes and access to archaeological data. This can add a layer of transparency and integrity to the data, ensuring that unauthorised changes do not occur.

Automating Data Backup and Archiving: Configure Kubernetes to automatically perform backup and archiving of archaeological data. This ensures that data is regularly backed up, making it resilient to the loss of information due to unexpected events.

Adapting Resources to Project Needs: Kubernetes allows dynamic resource adaptation based on the current needs of the system. This is particularly useful for archaeological projects that may experience changes in data volume over time. For instance, resources can be automatically scaled during intensive data collection periods.

These ideas can serve as a starting point for contemplating innovative approaches to storing archaeological data using Kubernetes systems while adhering to security and reliability standards. It is important to tailor these ideas to the specific requirements of archaeological research.

CONCLUSION

Kubernetes is a useful tool for managing the deployment and maintenance of applications in the cloud and other environments, which helps optimise resources and enables the easier scaling of applications. It represents a unique and useful platform for managing various applications. It is used worldwide in many different industries, including finance, technology, telecommunications, manufacturing, and more. Its usefulness in the form of automated deployment and scaling of applications enables easier work with large systems. It is designed to automatically replace non-functional containers, increasing the stability and reliability of the system. It enables discovery and load balancing, which greatly simplifies communication between containers and access to external resources. When it comes to resource management, resource orchestration is provided, which means that it automatically manages the resources used by applications. Additionally, Kubernetes allows simple updating of applications without downtime, which enables the smooth operation and improvement of the system.

In archaeology, Kubernetes could be used to manage databases in a way that allows for an easier deployment and scaling of databases across multiple servers, which would facilitate data access and processing in research. It could also be used to automate the process of backing up and restoring databases in the event of a failure.

From all of the above, it can be seen that Kubernetes can be used in many different contexts, including large and small organisations, different industries and cloud platforms, and considering the fact that it enables the efficient design, deployment and management of scalable systems, it represents an extremely useful tool in the administration of complex systems.

BIBLIOGRAPHY

AWS
Products, Compute, Amazon Elastic Kubernetes Service, https://aws.amazon.com/eks/ (accessed on November 29th 2023).

Azure
Azure, Products, Azure Kubernetes Service (AKS), https://azure.microsoft.com/en-us/products/kubernetes-service/ (accessed on November 29th 2023).

Burns, B., Beda, J., Hightower K. 2019
Kubernetes: Up and running, Sebastopol: O'Reilly Media.

Elastic 2023
Elastic, Elastic Stack, Kibana, https://www.elastic.co/kibana/ (accessed on May 16th 2023).

Fluentd 2019
0.12, Articles, Kubernetes Fluendt, https://docs.fluentd.org/v/0.12/articles/kubernetes-fluentd (accessed on May 16th 2023).

Google Kubernetes Engine (GKE) 2023
Google Cloud, Guides, Services, https://cloud.google.com/kubernetes-engine/docs/concepts/service (accessed on May 16th 2023).

Gutierrez, C. 2021
Altoros, Blog, The Pompeii Museum develops a mobile app on Kubernetes in six weeks, https://www.altoros.com/blog/the-pompeii-museum-develops-a-mobile-app-on-kubernetes-in-six-weeks/ (accessed on November 29th 2023).

Jetha, H. 2020
Digital Ocean, Tutorial, How to set up an Elasticsearch, Fluentd and Kibana (EFK) Logging Stack on Kubernetes,
https://www.digitalocean.com/community/tutorials/how-to-set-up-an-elasticsearch-fluentd-and-kibana-efk-logging-stack-on-kubernetes (accessed on May 16th 2023).

Kubernetes 2023
Kubernetes documentation, Concepts, Overview, https://kubernetes.io/docs/concepts/overview/ (accessed on May 16th 2023).

Lukša, M. 2020
Kubernetes in action, second edition, Shelter Island: Manning Publications Co.

Merron, D. 2020
BMC, Blogs, Port configurations for Kubernetes services, https://www.bmc.com/blogs/kubernetes-port-targetport-nodeport/ (accessed on May 16th 2023).

Microsoft Learn 2023
Azure, Azure Kubernetes Service, Network concepts for applications in Azure Kubernetes Service (AKS), https://learn.microsoft.com/en-us/azure/aks/concepts-network (accessed on May 16th 2023).

OCI
Cloud, Cloud Native, Container Engine for Kubernetes, https://www.oracle.com/cloud/cloud-native/container-engine-kubernetes/ (accessed on November 29th 2023).

Pompeii Sites
Pompeii, Visiting Info, MyPompeii, The application which allows one to visit the archaeological site in safety, http://pompeiisites.org/en/visiting-info/mypompeii-the-application-which-allows-one-to-visit-the-archaeolog-

ical-site-in-safety/ (accessed on November 29th 2023).

Poulton, N. and Joglekar, P. 2020
The Kubernetes book, Victoria: Leanpub.

Saito, H., Chloe Lee, H. and Carol Hsu, K. 2018
Kubernetes cookbook, Birmingham: Packt Publishing.

Sayfan, G. 2017
Mastering kubernetes, Birmingham: Packt Publishing.

Shtein, N. 2023
Komodor, Learning centre, Kubernetes nodes – The complete guide, https://komodor.com/learn/kubernetes-nodes-complete-guide/ (accessed on May 16th 2023).

Wilson, B. 2023
Devopscube, How to guides, How to setup Prometheus monitoring on Kubernetes cluster, https://devopscube.com/setup-prometheus-monitoring-on-kubernetes/ (accessed on May 16th 2023).

REZIME

KUBERNETES KAO PLATFORMA ZA UPRAVLJANJE APLIKACIJAMA

KLJUČNE REČI: K8s, KUBERNETES, CLOUD, CONTAINER, APPLICATION MANAGEMENT PLATFORM.

Kubernetes je tehnologija koja se koristi za upravljanje raspoređivanjem i skaliranjem aplikacija u oblaku. Kao korisno oruđe za upravljanje raspodelom i održavanjem aplikacija u oblaku i drugim okruženjima pomaže u optimizaciji resursa i omogućava lakše skaliranje aplikacija. Skaliranja aplikacija mogu biti primenjena na više servera, kako bi se obezbedilo da se one uvek pokreću na optimalnom broju resursa. Dodatno, automatskim preusmeravanjem prometa između servera, obezbedjuje se visoka dostupnost. Tehnologija Kubernetes bi mogla biti korisna i u arheologiji, i to za upravljanje raspoređivanjem različitih aplikacija za obradu i analizu podataka iz arheoloških istraživanja.

Kubernetes može da unapredi rad aplikacija tako što omogućava lakši razvoj, testiranje i održavanje kontejnerizovanih aplikacija. Benefit njegove upotrebe ogleda se u tome što se može primeniti u mnogim različitim kontekstima, uključujući velike i male organizacije, različite industrije i cloud platforme, a s obzirom da omogućava efikasan dizajn, deploy-ovanje i upravljanje skalabilnim sistemima, predstavlja izuzetno koristan alat u administriranju složenih sistema.

* * *