Library Progress International
Vol.44 No.3, Jul-Dec 2024: P.3443-3450

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Automated Application Deployment using Kubernetes in Cloud

[1]Jayasimha S R*,[2]Rohitahaksha K,[3]Divakar H R ,[4]Bhavya B M, [5]Varadaraj R, [6] M J Anand.

[1]*Department of MCA, JSS Academy of Technical Education, Bengalore,* Karnataka, India, jayasimhasr@jssateb.ac.in
[2]*Department of MCA, JSS Academy of Technical Education, Bengalore,* Karnataka, India, rohithaksha.k@gmail.com.
[3]*Department of MCA, PES College of Engineering, Mandya,*Karnataka, India, divakarhr@gmail.com
[4]*Department of MCA,PES College of Engineering,Mandya,*Karnataka, India, bhavyabm@pesce.ac.in
5Department of MCA, Navkis College of Engineering, Hassan, Karnataka, India, varadaraj@navkisce.ac.in
[6]*Department of Electronics and Communication Engineering, PES College of Engineering, Mandya,* Karnataka, India, anamysore@gmail.com

How to cite this article: Jayasimha S R, Rohitahaksha K,Divakar H R ,Bhavya B M, Varadaraj R, M J Anand. (2024). Automated Application Deployment using Kubernetes in Cloud, *Library Progress International.* 44(3), 3443-3450.

ABSTRACT
In the cloud computing environment kubernetes is an open-source container-based orchestration platform. It automates the deployment, scaling, manage the applications. Kubernetes managed the deployment of thousands of applications within Google. It can be tracing back the applications from the cloud. Google open-sourced version of Borg is Kubernetes. Managing with large scale applications in the data center cloud is always a difficult task. To manage with large scale applications in different workload conditions and to handle critical data required containerized environment. In this paper discussed with the design and development of kubernetes in the cloud. Setting up master slave model and it responsible for allocating task and complete the deployment in cluster. The infrastructure using terraform is supported to scale down the application and track the status of nodes in kubernetes cluster. In the proposed model discussed low-cost deliver pay as you go model in the cloud computing.

KEYWORDS
Kubernetes, Terraform, Grafana, Ansible, Docker.

Introduction

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Originally kubernetes was designed by Google, but the Cloud Native Computing Foundation maintains cloud. Kubernetes, Docker, Container, and CRI-O Originally, it interfaced exclusively with the Docker runtime through a "Dockershim" but it was replaced with the container runtime interface[1]. Ansible playbooks, inventory, provisioning tools, and domain expertise are all combined in Kubespray to handle general OS/Kubernetes cluster configuration management duties. Kubespray offers a cluster, that is extremely available composable characteristics most widely used Linux distributions and is supported Ansible playbooks, inventory, provisioning tools, and domain expertise are all combined in Kubespray to handle general OS/Kubernetes cluster configuration management duties.

To isolate the performance of containers, Kubernetes is adequate. This differs from earlier studies, which have a stronger emphasis on effective resource management than on performance interference and to assess the performance interference between two sets of network and CPU-intensive containers, as well as between different sets of these containers. According to the findings of our evaluation, even with Kubernetes' resource management, co-located containers cause containers' performance to drop by 50%. Also demonstrates that CPU contention, not network bandwidth, is the primary driver of performance interference across numerous network-intensive containers. As a result, in order to reduce performance interference, Kubernetes must take into account the CPU utilization of network-related workloads in resource management.

Software engineers spend half of their working hours wading through the work of our forerunners, but veterans and great people of the community do emerge and just as the working individuals have learnt the lessons about

writing the code, the community has learned collective lessons to work with software development at scale. It is not easy to see these larger trends when they are busy doing small bug fixes. DevOps is one of these huge trends[2].

It is basically the mixture of two separate worlds of software development into one common cycle. It is not a recent act of invention. it is a result of years of iteration. The deployment and management of applications in large-scale clustered environments are always a difficult task. Docker, Kubernetes, Terraform are a few various container orchestration frameworks that exist in the market. There is still a lack of the best suited orchestration solution. The purpose of this work is to fill the gap and provide the better result for the same.

In the Kubernetes master and slave, Kubernetes consists of a master process that manages slave daemons running on each cluster node and frameworks that run tasks on these slaves. The Terraform framework which is built to provision the infrastructure initializing, planning and applying. [3]TheKubespray is used for building ready cluster configuration information and to track the status of nodes in the Kubernetes cluster. In the EC2 instance which is used to host the virtual machines which will contain the private and public address for every instances, it can be connected to the Kubernetes cluster in order to deploy the application.

1. Kubernetes: ServicesOffered

1.1 e-Commerce

Zopsmartgivesretailer'scustomersaworld-classshoppingexperienceduringIntuitivesearchtoquicklyfindproducts.Personalizedproductlistingforeaseofordering.Self-service rescheduling and returns functionality for frictionless order experience.

1.2 m-Commerce

Provide to the customer as superlative shopping experience on mobile devices - Responsive website that adapts to all screen sizes. Native Android and iOS mobile apps for world-class experience. Persistent cart across devices for convenience[4].

1.3 mm-Commerce

Provide to the customer as superlative shopping experience on mobile devices - Responsive website that adapts to all screen sizes. Native Android and iOS mobile apps for world-class experience. Persistent cart across devices for convenience[4].

1.4 DigitalMarketing

Createtargetedmarketing campaigns for highest ROI - Customer segmentation fortargetedcampaigns.Conditionalcouponstoavoidleakageandunintendeduse.Integrated communication engine to easily send emails, sms and app-notifications.

1.5 e-Wallet

Use integrated wallet to build customer loyalty - Integrated wallet to enable quick and hassle-free checkout. Refunds and cashback to wallet and encourage repeat purchase. Wallet-credit as gift-cards for acquiring new customers at low cost.

1.6 CustomerManagement

Manage your online as well as offline customers easily and efficiently - Single interfacetoviewallyourcustomerswiththeircontactdetails.Availabledetailsofeachcustomer's purchase history, orders and loyalty points. Option to contact the customerfrom the interface by phone-call, sms and email.

1.7 OrderManagement

Manage your orders easily and efficiently - Single interface to see all your orders withtheir details such as customer name, value and order-status. Predefined order statusesthatchangewithcompletionofeveryprocessstep.Optiontoaddoredittheitemsinthe order.

1.8 Monitoring andAnalytics

Runyourbusinesswithminimalsupervision - Real time tracking to monitor eachaspect of your business. Deviation alerts in the operations process to trigger timelyinterventions.Intelligentanalyticstoidentifytrendsforprocessimprovements.Detailed logs for each action to trace back any errors.

1.9 SmartPOS

Giveyourcustomers to atrulyomni-channelshoppingexperience-Single customeraccountacrosschannels(Walk-in,Online,KioskandTele).Enablespurchasefromachannel & returns from any channel. Common loyalty program across channels. Worksoffline too[5].

1.10 SmartSelf-checkout

Mobileapp-based queue-buster for your store - Adds products to cart by scanningproduct-barcodes. Adds products to cart using search-box. All modern online methodsavailable for payment. Enables quick security-check and exit from the store.

1.11 SmartProduct-locator

Mobile app-based product-locator for your store customers - Uses WiFi or infraredbeacons to locate customer's position. Provides an exact path to reach the product.Alertsstorestafftohelpcustomers.Alertsstorestaffifitemnotfound.Allowscustomers to proceed for self check-out.

2. Literaturereview

The infrastructure withload balancers and node pools and kubernetes cluster to provision the set of master and workernodes. To do this, we run a number of experimental evaluations of containers to track howsystemresourcessuchasCPU,memory,storage,andnetworkbehave.Forcomparison,aKubernetes cluster that was manually installed served as the baseline.

Thesystem'sspecification,component design, implementation, and evaluation ofchosen infrastructure to develop a monitoring engine and network-related statistics in the cloudplatform and also tells about the ip address of the instances and all the resources which areprovisioned. Discussed the advantage of employing the IaaC concept over manualactivities needed to restore the system in cases where complicated cloud systems are involved[6] .

The genericframeworkwhichisnamedasKube-Monitorforreal-timemonitoringofaheterogeneousKubernetes cluster's resources, which include traditional data center devices and raspberry pidevices,aswellas monitoringpodresourcerequirements. Kube-Monitorwillcollect15 differentcluster resource metrics and update their values every five seconds. The containerizing environment leads to an OS level virtualization and from that we can run multipleservices on a single host.This Kubernetes helps to scheduling and managing clusters and that act as anecosystem for managingclusters of containers.The Kubernetes environmenthave a manualload balancing for providing high availability. The solution combines our HyperFlow workflowmanagement system with de facto standards and widely used tools likeTerraform and Kubernetes.

The goal is to introduce generic framework which is named as Kube-Monitor for real-time monitoring of aheterogeneous Kubernetes cluster's resources [8], which include traditional data center devices andraspberry pi devices, as well as monitoring pod resource requirements but fail to work on thetestingor productionenvironment dueto theenvironment differences,if any.Containerization comesintothepicturetoaddresssuchchallenges.

The system'sspecification,componentdesign, implementation, and evaluation of chosen infrastructure to develop a monitoring enginefor exposing JVM and network-related statistics of the mentioned system using Docker Swarm,ApacheSpark,Graphite,andGrafana,Virtualmachinesolvesmanyproblemsbyoptimizingtheresources.The developers are concerned that the code is working fine on the development Environment [9,10].

Theaim to provide a comprehensive description of cloud orchestration approaches with containers [11], analyzing currentresearchefforts,existingsolutions andpresentingissues andchallengesfacing thistopicto easilyfind the configurations of the pods, running services and deployments in the cluster theytalkabouttheresultsshowthatthe overall performance of Kubernetes for the deployment purpose in the cloud and alsoworking it locally using the minikube cluster. [12]Effectiveness of the Kubernetes technology in a setting like this by testing and analysingthethreepreviouslystatedcharacteristics.Additionally,thetestingexperimentmight show a piece of data on the dashboard for the sake of visualization and analysis.

The Container orchestration systems[13], such as Docker Swarm, Kubernetesand Mesos, provide automated support for deployment and management of distributedapplications.By launching a web application on Microsoft Azure using only code, theyhave shown the value and significance of IaC using Azure Resource Manager (ARM)Templates. The deployment and management process were shown to be more agile whenARM templates were used. Using Terraform

and Pulumi, we will also demonstrate IaC.The relationship between IaC and the ideas of code repeatability and reusability is alsocovered in the study.

Themaingoalis to analyze current options of Dockercontainerorchestrationandtheiradvantagesanddisadvantageswhiledeployingtheapplications in the cloud and also retrieving the images from the docker hub registry orfrom the other registry and examines the TOSCA standard as well as the tools that havereceived the most citations in the literature, including Cloudify, Heat, CloudFormation,Terraform, and Cloud Assembly. An experiment in practice and a review of the literatureshowed that Terraform and Cloudify have a strong affinity for sky computing scenarios.Terraform performed better than Cloudify in the experiment in a number of ways.

Apache Mesos, acluster-wideresourcemanager, which is widely deployed in massive scale at severalCloudsandDataCenterstogivethebestserviceandalsotheperformance for theapplications which is deployed in the cloud.Thecoreofthiseffortismakingeffectiveuse of these modules when creating the code. Ansible is chosen above the other toolssince it lacks an agent and connects to Linux and Windows systems through the SSH andWinRMprotocols,respectively.UsingtheautomationofAnsibleandTerraformover cloud servers[14].

The technology based on Dockerapplication container cluster is gradually moving towards a production environment andalsoitsadvantagesin production deployment services and an intermediate templatebasedontheSpecification will be created using this metadata. It will then be processedbythedeployment and execution engine, which will further assemble the necessarytemplatepartsintotheappropriateinfrastructure-as-codefortoolslikeKubernetes,Terraform, and Ansible. In order to provide a user-friendly interface for those who areunfamiliar with Manufacturing-as-a-Service and Digital Twins, an implementation of thesolution is now being built for a European project looking into these topics[15].

ApacheMesosframeworkforcontainer orchestration[16] providing aREST API for starting, stopping, and scaling applications including non-containerizedinfrastructureslikeAmazonLambdaandnon-containerizedinfrastructureslikeKubernetes or Amazon ECS. The resulting technology enables the execution of hybridworkflows that make use of several computing infrastructures and considerably reducesthedifficultyassociatedwithmanagingrepetitiveinfrastructuresforcarryingoutscientific workflow research.[17] highlights about terraformas apowerful, feasible approach to build the infrastructure in the cloud according to the needof the application and mentioned the outlines of a novel approach to elastic containercloud scaling that uses a prediction model and a queuing network to estimate future webapplication request arrival rates and determine the ideal number of Kubernetes servicereplicasets.Finally,aKubernetesclusterisputupinatestingenvironment,andexperiments are planned to validate our approach's efficacy.Thesesurveypapersprovidedan overview of container orchestration tool frameworkslike Kubernetes, Docker swarm, and Terraform as well as an analysis of their currentissuesandupcomingchallengesthatmustbe resolved. In this way, the papers gavereaders a better understanding of the difficulties that these frameworks face. [18] It also explains about Marathon as a framework for the Terraform, Marathon provides thegraphical user interface for the connected masters and slaves of container orchestrationalsoprovides the REST Api for the coordination between the cluster.In these surveypapers got to know about the deployment of kubernetes clusters through terraform in thecloud. The creation of the VPC, RDS, instances and cluster services has got to know. Thestudy gives more concentration on deploying an application in the cloud through theautomation of terraform by specifying the required details[19].

The study highlights the technology based on Docker application container clusters isgraduallymovingtowardsaproductionenvironment.TheyalsoexplainedaboutKubernetes as aframework for container orchestration. It helped to analyze current options of Docker container orchestrationand their advantages and disadvantages. It includes discussion on Kubernetes, Terraform,which is widely deployed for massive scale Cloud applications and Data Centers[20].

3. ProposedSystem

Theproposedsystemintendstoprovidetheinfrastructuresolutionsthatprovidetheorganizationswithlowcostwaystodeliver theirwebsiteandwebapplicationalsoprovidingpay as you go, Database storage, compute power and other functionality to help businesses scaleand grow, enhancing security to protect data and workloads.

Thisplatform'screationtransformedhowbusinessesconstruct,moveapplicationsbetweendatacenters,andmaintainthemin real-time.Italsoservesasaplatformforapplicationdeployment.
Applicationdeployment,upkeep,scalability,andnetworkingareallautomatedby container orchestration. Container orchestration may be the only reliable option in the modern world, when businesses must deploy and manage several

hosts.

Technologiesused

- Kubernetes is a portable, extensible, open-source platform for managing containerizedworkloads and services, that facilitates both declarative configuration and automation. Ithasalarge,rapidlygrowingecosystem.Kubernetesservices,support,and tools arewidely available and the most well-liked platform for container orchestration allowsuserstobuildandexecuteseveralcontainersin cloud settings.Because performanceisolation is a crucial element in determining the quality of a service, Kubernetes providesresourcemanagementtoisolatethe resourceutilizationofcontainers onahostserver.

- Terraform is an infrastructure as code tool that lets you define both cloud and on-premresources in human-readable configuration files that you can version, reuse, and share.Youcanthenuseaconsistentworkflowtoprovisionandmanageallofyourinfrastructure throughout its lifecycle.Traditional forms of cloud service delivery includeInfrastructureasaService(IaaS),PlatformasaService(PaaS),andSoftwareas aService (SaaS).According to Gartner, by 2022, 90% of businesses that use IaaS serviceswill do so through a provider that also provides PaaS, and these businesses will employboth of that provider's services.

- AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platformprovidedbyAmazonthatincludesamixtureofinfrastructureasaservice(IaaS),platform as a service (PaaS) and packaged software as a service (SaaS) offerings andreview of security studies in the area of cloud security is provided. Following a securityanalysis,wehaveshownhowAWS(AmazonWebService)cloudcomputing.

Test-bed Environment

To setup a cloud environment required 32 GB Minimum and 64 GB Maximum disk space with processor 1.5 GHz 64 bit like NVIDIA, Hewlett-Packard, 16 GB Minimum and 32 GB Maximum RAM memory, Bare-metal server-Depending on the infrastructure requirement, Linux/MacOS Operating system, Firefox latest version 99.0- Browser, AWS CLI version 2 Platform, Minimum Version: 20.10.6 Maximum Version: 21.11.10, Docker and Minimum Version: 0.12, Maximum Version: 1.1.1 Terraform.

Deployment Specifications

In the kubernetes identify the optimal container orchestrationframeworkfrom a bunch of existing tools Kubernetes found to be the one which fulfill therequirement for the testbed. For the Terraform setup, Inminikube master and slave setups areconfigured. Terraform will provide the infrastructure for the resources by provisioning it in thecloud.

Further, it is to build a infrastructurefor the application toautomate in the terraform cloud using the github actions to create it in the amazon web serviceandtogivetheavailabilitytotheusersbyconfiguringtheyamlfiles to deploy it in thekubernetes cluster,

Oncetheapplicationisdeployedallthedeploymentdaemonsshouldberunninginthekubernetescluster. Containerizedandnon-containerizedapplications,makingiteasierforcorporateorganizationstochoosemesoswhenworkingonongoingprojects. Because it is weakly connected, using containerized Docker images will make it easierfor testers to avoid dependency problems.

User:Anyone who is interactingwith the software isa user.
Load balancer: to equally convey loadover a bunch of backend assets or servers.
VPNGateway: VPNPortalsendsencryptedactivitybetweenanA W S instanceandanon-premises area over the open Internet

Acronyms andAbbreviations

HCL: HashiCorp configuration Language
YAML : Yet another markup language

JSON: Javascript object notation
OS: operating system

First phase: Minikube Setup

a. Identifying the optimal container orchestration framework based on the existing infrastructure.
b. Kubernetes master and slave setup according to the requirement with the application.
c. Setting up the yaml files to configure the automation of the application performance in the deployment environment.
d. Creating the cluster with the configured files to run all the services and pod.

Second phase: Provisioning the Infrastructure Using Terraform

a) Terraform setup for initializing all the providers to download and the hashicorp registry api.
b) Terraform plan is for getting all the plan of the infrastructure that is going to be provisioned in the AWS platform
c) Terraform apply is to setup all the provisioned resources that will be provisioned in the AWS platform.

Deploying the Application

Purpose: To deploy the application with no downtime for the users Input: Docker File

Function: Docker file is validated and retrieved from the registry

Output: Kubernetes cluster will show the running deployment daemons, pods and services of the application in the cluster

Provisioning the Infrastructure

Purpose: To build the infrastructure in the AWS platform for the Kubernetes.

Input: Terraform files
Function: when terraform is applied it should provision the infrastructure in the cloud platform which will satisfy the application requirements.

Output: Provisioned resources in the cloud and also the backend state file.

Establishing Connection

Purpose:To connect the kubernetes cluster with the endpoint obtained in the terraform Input : IP address of the instance.

Function:By running the ssh commands to connect to the instance from the kubernetes cluster.

Output:The application will be accessible from any browser.

ExternalInterfacesRequirements

User Interface: The user interface for the software shall only be compatible with Browser.

Hardware Interface: MONITOR: Any LCD, LED, TFT displays can be used

SoftwareInterface:

Virtualboxisusedformachinesaswellasdifferenttechnologieslikescapy,netifacesand libraries have been used.
Amazon web services cloud platform is required to configure and deploythe website.

4. ArchitecturalDesign

Theinfrastructurefortheapplicationwhichis deployed, kubernetescluster to remove the downtime for the users and give the high availability in order to access theapplication in the cloud. The deployment of containers has changed the way organizations create,ship,and maintain applications in real-time. module enables an additional level of abstractionand it also ensures the smooth network flow traffic between the clients and the provided specificserversbyhidingtheactualserverdetails.Nginxreverseproxyactsasanintermediateserverthat intercepts client requests and forwards them to the appropriate upstream backend server andsubsequently forwarded a response from

the server back to the client. The reverse proxy providesvarious benefits as an abstract layer above upstream servers.

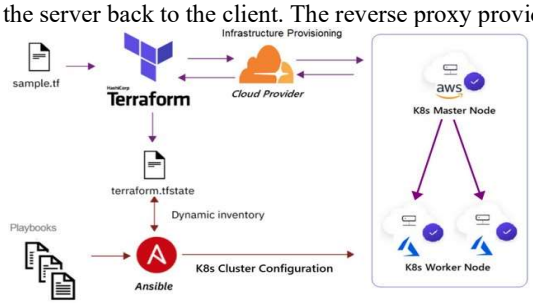

Figure 1: Block diagram of kubernetes

The block diagram of the task is basic and adaptable, here is the block diagram which comprisesthe relative multitude of significant modules present in the undertaking.

Figure 1 shows the design of the undertaking and the total progression of the interaction. Itshowsthegeneralengineeringoftheframeworkdeveloped. This includes provisioning theterraform infrastructure in the cloud and by running the ansible playbooks to make the readycluster in the kubernetes.

4.1 DeployingtheApplication

Deploying the application provides high availability in the kubernetes cluster to give the availability for the users to use the application which is already available in the internet and also allows users to specify the computing resources for their pods in the pod description file, which provides resource management. Therefore, while requesting the development of a pod, customers use the description file along with required computer resources. By hiding the actual server information, the module offers an additional degree of abstraction while simultaneously ensuring seamless network traffic between the clients and the specified particular servers. The Nginx reverse proxy functions as a middle server that receives client requests, routes them to the proper upstream backend server, and then routes the server's response back to the client. The reverse proxy offers a number of advantages as an auxiliary layer above upstream servers.

4.2 Provisioningtheinfrastructure

Whenthe terraform is initialized it willdownload all the required resources and it will beprovisioned in the cloud platformaccording to the requirement of the application and also toreduce the application downtime for the users to use the application. The computing resources arethen reserved using cgroups and tc in the chosen host server by Kubernetes, which chooses hostserverstooperate the pod. Coordinate infrastructure modifications among various public andprivatecloudserviceproviders. Itisa flexible infrastructure solution that may be used to manage software and services.

4.3 Deploying the application to cloud Module

The application should be available to end user through openly accessible URIs for that reasonthe application is deployed in cloud resources with load balancer load balancer helps to balancethe traffic on the virtual machines and also suppose one machine is down then another machineworks it increases availability and for deploying the project ansible is used ansible is automationtoolwhichcanautomatethewholedeploymentprocess. Acontainerizedapplicationrunssomewhere in the cluster, and Kubernetes tells its parts where to locate one another and keepsthemallrunningcontinuously. Regardlessofwhichnode is active, the programme does notcare. Asaresult,Kubernetescanmovetheapplication at any time by consuming more netresources than would be possible with manual configuration technique.

5. Conclusion

In this paper discussed with the development of generic platform to facilitate dynamic resource provisioning over kubernetes. paper insights with the applications of kubernetes and functionalities. The tools and technologies used and test-bed specifications to deploy a application in the kubernetes has discussed. Architectural design of kubernetes and the various stages of application deployment concept are highlighted. Finally the interaction of terraform, ansible and AWS while deploying an application has discussed. In future to retrieve the activities of a particular user and display at the starting page when the user logs in with google, so that each user activity is logged on the server can be implemented.

REFERENCES

[1]. Z. Cai and R. Buyya, "Inverse Queuing Model-Based Feedback Control for Elastic Container Provisioning of Web Systems in Kubernetes," in *IEEE Transactions on Computers*, vol. 71, no. 2, pp. 337-348, 1 Feb. 2022, doi: 10.1109/TC.2021.3049598.

[2]. Juan Marcelo Parra-Ullauri, Hari Madhukumar, Adrian-Cristian Nicolaescu, Xunzheng Zhang, Anderson Bravalheri, Rasheed Hussain, Xenofon Vasilakos, Reza Nejabati, Dimitra Simeonidou,kubeFlower: A privacy-preserving framework for Kubernetes-based federated learning in cloud–edge environments,Future Generation Computer Systems,Volume 157,2024,Pages 558-572,ISSN 0167-739X, https://doi.org/10.1016/j.future.2024.03.041.

[3]. Chenggang Shan, Yuanqing Xia, Yufeng Zhan, JinhuiZhang,KubeAdaptor: A docking framework for workflow containerization on Kubernetes, Future Generation Computer Systems,Volume 148,2023,Pages 584-599,ISSN 0167-739X,https://doi.org/10.1016/j.future.2023.06.022.

[4]. Sören Henning, Wilhelm Hasselbring,Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud,Journal of Systems and Software,Volume 208,2024,111879,ISSN 0164-1212, https://doi.org/10.1016/j.jss.2023.111879.

[5]. Engineer Bainomugisha, Alex Mwotil, Crane Cloud: A resilient multi-cloud service abstraction layer for resource-constrained settings, Development Engineering, Volume 7, 2022, 100102, ISSN 2352-7285,

[6]. A. Malhotra, A. Elsayed, R. Torres and S. Venkatraman, "Evaluate Canary Deployment Techniques Using Kubernetes, Istio, and Liquibase for Cloud Native Enterprise Applications to Achieve Zero Downtime for Continuous Deployments," in IEEE Access, vol. 12, pp. 87883-87899, 2024, doi: 10.1109/ACCESS.2024.3416087.

[7]. J. Han, Y. Hong and J. Kim, "Refining Microservices Placement Employing Workload Profiling Over Multiple Kubernetes Clusters," in IEEE Access, vol. 8, pp. 192543-192556, 2020, doi: 10.1109/ACCESS.2020.3033019.

[8]. X. Zhang, L. Li, Y. Wang, E. Chen and L. Shou, "Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters," in IEEE Access, vol. 9, pp. 105192-105204, 2021, doi: 10.1109/ACCESS.2021.3100082.

[9]. F. Lumpp, F. Fummi, H. D. Patel and N. Bombieri, "Enabling Kubernetes Orchestration of Mixed-Criticality Software for Autonomous Mobile Robots," in IEEE Transactions on Robotics, vol. 40, pp. 540-553, 2024, doi: 10.1109/TRO.2023.3334642.

[10]. X. Zhang, L. Li, Y. Wang, E. Chen and L. Shou, "Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters," in IEEE Access, vol. 9, pp. 105192-105204, 2021, doi: 10.1109/ACCESS.2021.3100082.

[11]. R. Gao, X. Xie and Q. Guo, "K-TAHP: A Kubernetes Load Balancing Strategy Base on TOPSIS+AHP," in *IEEE Access*, vol. 11, pp. 102132-102139, 2023, doi: 10.1109/ACCESS.2023.3313643.

[12]. Wang, H., Liu, L., Yue, C. *et al.* Cloud-native systems resilience assessments based on kubernetes architecture graph. *SOCA* (2024). https://doi.org/10.1007/s11761-024-00406-x.

[13]. Senjab, K., Abbas, S., Ahmed, N. *et al.* A survey of Kubernetes scheduling algorithms. *J Cloud Comp* 12, 87 (2023). https://doi.org/10.1186/s13677-023-00471-1.

[14]. Wang, YT., Ma, SP., Lai, YJ. *et al.* Qualitative and quantitative comparison of Spring Cloud and Kubernetes in migrating from a monolithic to a microservice architecture. *SOCA* 17, 149–159 (2023). https://doi.org/10.1007/s11761-023-00364-w.

[15]. Liu, J., Ding, Y. & Liu, Y. A balanced leader election algorithm based on replica distribution in Kubernetes cluster. *Cluster Comput* (2024). https://doi.org/10.1007/s10586-024-04333-6.

[16]. Kim, E., Lee, K. & Yoo, C. Network SLO-aware container scheduling in Kubernetes. *J Supercomput* 79, 11478–11494 (2023). https://doi.org/10.1007/s11227-023-05122-5.

[17]. Piontek, T., Haghshenas, K. & Aiello, M. Carbon emission-aware job scheduling for Kubernetes deployments. *J Supercomput* 80, 549–569 (2024). https://doi.org/10.1007/s11227-023-05506-7.

[18]. S. Burroughs *et al*., "Towards Autoscaling with Guarantees on Kubernetes Clusters," *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, DC, USA, 2021, pp. 295-296, doi: 10.1109/ACSOS-C52956.2021.00073.

[19]. L. Toka, G. Dobreff, B. Fodor and B. Sonkoly, "Machine Learning-Based Scaling Management for Kubernetes Edge Clusters," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 958-972, March 2021, doi: 10.1109/TNSM.2021.3052837.

[20]. Z. Li, H. Wei, Z. Lyu and C. Lian, "Kubernetes-Container-Cluster-Based Architecture for an Energy Management System,"