IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15357

Kubernetes on aws with eks

Rijo[1], Dr. S. Thavamani[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous),

Coimbatore, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Applications,

Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore, Tamil Nadu, India[2]

Abstract: The rapid growth of cloud computing and microservice-based application development has increased the demand for efficient container orchestration platforms. Kubernetes has emerged as the leading solution for managing containerized workloads, offering automation, scalability, and high availability. At the same time, cloud providers such as Amazon Web Services (AWS) provide managed services that simplify Kubernetes cluster management.

This paper presents a practical implementation of deploying a containerized e-commerce web application using Kubernetes on Amazon Elastic Kubernetes Service (EKS). The system demonstrates how containerization with Docker, combined with Kubernetes orchestration and AWS cloud infrastructure, can create a scalable and reliable application deployment environment.

The architecture integrates multiple AWS services including Amazon EC2, Amazon VPC, Elastic Load Balancing, and Amazon CloudWatch to manage compute resources, networking, traffic distribution, and monitoring. The proposed solution improves application reliability, simplifies infrastructure management, and enables automatic scaling based on demand. Experimental deployment results show that Kubernetes on AWS EKS provides a robust platform for modern cloud-native applications.

I. INTRODUCTION

Modern software systems increasingly rely on distributed architectures and microservices to deliver scalable and resilient applications. Traditional application deployment methods often rely on physical servers or virtual machines, which require significant manual configuration and maintenance. These approaches frequently lead to resource inefficiencies, deployment inconsistencies, and difficulties in scaling applications during periods of high demand.

Containerization has emerged as a powerful solution to these challenges. Containers package applications together with their dependencies, ensuring that they run consistently across different environments. Among containerization technologies, Docker has become widely adopted due to its simplicity and portability.

However, as applications grow in complexity and scale, managing large numbers of containers becomes challenging. Kubernetes addresses this problem by providing a container orchestration platform that automates deployment, scaling, networking, and management of containerized applications.

Cloud computing platforms further enhance this model by providing on-demand infrastructure resources. Amazon Web Services (AWS) offers Amazon Elastic Kubernetes

Service (EKS), a fully managed Kubernetes platform that removes the complexity of managing Kubernetes control plane components.

This paper explores the deployment of a containerized e-commerce web application using Kubernetes on AWS through EKS. The study focuses on designing the system architecture, implementing containerized deployment, and evaluating the benefits of using managed Kubernetes services in cloud environments.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15357

II. RELATED WORKS

The use of containerization and orchestration technologies has been widely explored in cloud computing research. Docker-based container environments have become a common method for packaging applications and their dependencies, providing portability and consistency across development and production environments.

Burns et al. introduced Kubernetes as a scalable container orchestration platform capable of managing distributed container clusters. Their work demonstrated how Kubernetes automates deployment, scaling, and resource allocation for containerized applications.

Turnbull emphasized the importance of Docker in modern application deployment, describing how containers simplify infrastructure management and enable continuous delivery practices.

Several studies have also explored Kubernetes deployment in cloud environments. Research on managed Kubernetes services shows that platforms such as Amazon EKS significantly reduce operational overhead by automating cluster management tasks such as upgrades, patching, and control plane maintenance.

Other researchers have highlighted the advantages of combining Kubernetes with cloud services including load balancing, auto-scaling, and monitoring. These integrations enable organizations to build highly available and scalable cloud-native systems.

This project builds upon these studies by implementing a real-world containerized application deployment using Kubernetes on AWS EKS, demonstrating practical integration of cloud infrastructure and container orchestration technologies.

III. OBJECTIVES AND CHALLENGES

1.2 Objectives

The primary objective of this project is to design and implement a scalable cloud-native deployment system using Kubernetes on AWS EKS. The system aims to demonstrate how containerized applications can be efficiently deployed and managed in a cloud environment.

The specific objectives of the project include:

- Deploying a Kubernetes cluster using Amazon Elastic Kubernetes Service.
- Containerizing a web-based e-commerce application using Docker.
- Managing application deployment using Kubernetes pods and services.
- Implementing load balancing and scalable infrastructure.
- Integrating monitoring and logging tools to observe system performance.

By achieving these objectives, the project provides a practical demonstration of modern DevOps and cloud-native application deployment practices.

2.2 Challenges

Despite the advantages of Kubernetes and cloud computing, several challenges arise during implementation.

One major challenge is the complexity involved in configuring Kubernetes clusters and networking environments. Proper configuration of virtual networks, security groups, and access policies is required to ensure secure communication between components.

Another challenge is managing container images and ensuring that applications are correctly packaged and deployed. Errors in container configuration or deployment files can lead to application failures.

Resource management and cost optimization also present challenges when deploying cloud infrastructure. Efficient scaling strategies are necessary to balance performance and operational cost.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15357

Finally, monitoring and debugging distributed systems require effective logging and performance tracking tools.

IV. SYSTEM ARCHITECTURE

The system architecture is designed to integrate container orchestration with cloud infrastructure services. The architecture consists of several key components that work together to manage application deployment and operation.

At the core of the system is the Kubernetes cluster managed by Amazon Elastic Kubernetes Service (EKS). The EKS control plane is managed by AWS and is responsible for coordinating cluster operations such as scheduling, resource management, and communication between nodes.

Worker nodes are created using Amazon EC2 instances. These nodes run containerized applications inside Kubernetes pods. Each pod contains one or more containers that execute application services.

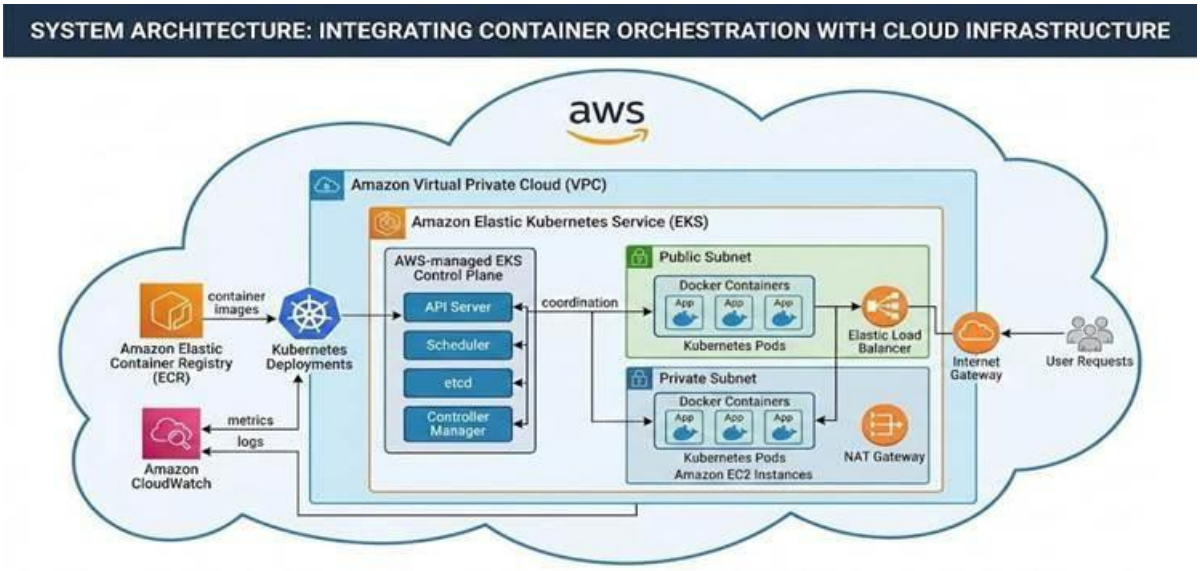
Networking is implemented using Amazon Virtual Private Cloud (VPC), which provides isolated and secure communication between cluster components. Public and private subnets are used to control network accessibility.

Elastic Load Balancing distributes incoming traffic across application pods, ensuring that the system can handle multiple user requests efficiently. This improves both system reliability and availability.

Container images are built using Docker and stored in container registries such as Amazon Elastic Container Registry (ECR). Kubernetes deployments use these images to create and manage application containers.

Monitoring and logging are implemented using Amazon CloudWatch, which tracks system performance metrics including CPU usage, memory utilization, and network activity.

The overall architecture ensures high availability, scalability, and efficient resource management for containerized applications.



V. IMPLEMENTATION

The implementation process involves several stages, including containerization, cluster setup, application deployment, and system configuration.

The first stage involves containerizing the application using Docker. A Dockerfile is created to define the environment required for running the application. The Docker image includes application code, dependencies, and runtime configurations.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15357

Once the image is built, it is stored in a container registry such as Amazon Elastic Container Registry (ECR). This allows the Kubernetes cluster to access the image during deployment.

Next, the Kubernetes cluster is created using Amazon Elastic Kubernetes Service. The cluster setup includes configuring worker nodes using Amazon EC2 instances and defining networking parameters within the Amazon VPC environment.

Application deployment is managed using Kubernetes YAML configuration files. These files define Kubernetes resources such as deployments, pods, and services. The deployment configuration specifies the number of application replicas, container images, and networking settings.

Kubernetes services expose the application to external users through load-balanced endpoints. This enables users to access the web application through a web browser.

Additional configurations include setting up security policies using AWS Identity and Access Management (IAM) and enabling monitoring through Amazon CloudWatch.

VI. EVALUATION RESULTS AND DISCUSSION

The deployed system demonstrates several advantages of using Kubernetes on AWS EKS for application management.

First, the system provides high availability by running multiple instances of application containers across worker nodes. If one container fails, Kubernetes automatically replaces it, ensuring continuous service availability.

Second, the system supports horizontal scalability. As user demand increases, Kubernetes can automatically increase the number of running container instances. This dynamic scaling ensures that the application maintains performance during periods of high traffic.

Third, the integration of load balancing improves system reliability by distributing traffic across multiple containers. This prevents individual containers from becoming overloaded.

Monitoring tools such as Amazon CloudWatch provide valuable insights into system performance. Administrators can track metrics, detect anomalies, and respond to potential issues before they impact users.

Overall, the evaluation results show that Kubernetes on AWS EKS provides an efficient and reliable platform for deploying cloud-native applications.

VII. CONCLUSION

This paper presented the design and implementation of a cloud-based container orchestration system using Kubernetes on AWS EKS. The project demonstrates how modern containerization technologies combined with managed cloud services can simplify application deployment and infrastructure management.

By using Docker for containerization and Kubernetes for orchestration, the system ensures consistent application environments and automated management of containerized workloads. The use of Amazon EKS further simplifies cluster management by eliminating the need to maintain Kubernetes control plane components.

The integration of AWS services such as EC2, VPC, Elastic Load Balancing, and CloudWatch enhances system performance, security, and monitoring capabilities.

The implementation results confirm that Kubernetes on AWS EKS provides a scalable, reliable, and efficient platform for deploying modern cloud-native applications.

VIII. FUTURE ENHANCEMENTS

Although the current system demonstrates the core capabilities of Kubernetes deployment on AWS EKS, several enhancements can further improve the platform.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15357

One possible enhancement is integrating Continuous Integration and Continuous Deployment (CI/CD) pipelines using tools such as Jenkins or GitHub Actions. This would automate application build, testing, and deployment processes.

Advanced monitoring tools such as Prometheus and Grafana can also be integrated to provide detailed performance dashboards and real-time alerts.

Security can be improved by implementing container image scanning tools that detect vulnerabilities before deployment.

Another enhancement involves implementing multi-region deployment to improve disaster recovery and system availability.

The system can also adopt serverless container technologies such as AWS Fargate to reduce infrastructure management overhead.

These improvements would enable the platform to support enterprise-level cloud applications with enhanced reliability and automation.

REFERENCES

[1]. Kubernetes Documentation. Available: https://kubernetes.io/docs
[2]. Amazon Web Services Documentation. Available: https://docs.aws.amazon.com
[3]. Amazon Elastic Kubernetes Service User Guide.
[4]. Burns, B., Beda, J., & Hightower, K. *Kubernetes: Up and Running*. O'Reilly Media.
[5]. Turnbull, J. *The Docker Book: Containerization is the New Virtualization*.
[6]. Morris, J. *Kubernetes in Action*. Manning Publications.
[7]. Docker Documentation. Available: https://docs.docker.com

