Kubernetes: เครื่องมือที่มีประสิทธิภาพสำหรับการจัดการคอนเทนเนอร์ในองค์กร
Kubernetes: An Efficient Tool for Managing Organizational Containers

สมชัย บุรีรักษ์[1*] และ ไพศิลป์ชัย ไชยพรเดชเจริญ[1]
Somchai Burirak[1*] and Paisinchai Chaiyaprondetcharoen[1]

บทคัดย่อ

ปัจจุบันที่การดูแลสุขภาพมีความซับซ้อนและมีความต้องการที่หลากหลาย องค์กรด้านสาธารณสุขต้องเผชิญกับความท้าทายในการจัดการระบบเทคโนโลยีสารสนเทศที่มีการเปลี่ยนแปลงอย่างรวดเร็ว Kubernetes ซึ่งเป็นระบบการจัดการคอนเทนเนอร์ที่มีประสิทธิภาพและความยืดหยุ่น ช่วยปรับปรุงการจัดการแอปพลิเคชันและทรัพยากรภายในองค์กรได้อย่างมีประสิทธิภาพ สามารถปรับขนาดและจัดการทรัพยากรได้อัตโนมัติ เป็นการตอบสนองต่อความท้าทายดังกล่าวได้อย่างมีประสิทธิภาพ

Kubernetes ช่วยลดเวลาในการพัฒนาและส่งมอบแอปพลิเคชัน เพิ่มความน่าเชื่อถือและเสถียรภาพของระบบผ่านการจัดการกู้คืนของแอปพลิเคชันที่เกิดจากข้อผิดพลาด ความสามารถในการปรับขนาดและการจัดการทรัพยากรที่มีประสิทธิภาพ ทำให้การดำเนินงานในองค์กรมีความราบรื่น และมีประสิทธิภาพมากยิ่งขึ้น

ดังนั้น Kubernetes จึงเป็นเครื่องมือที่มีประสิทธิภาพสำหรับการจัดการคอนเทนเนอร์ในองค์กร สามารถช่วยเพิ่มประสิทธิภาพและความยืดหยุ่นในการดำเนินงาน และเป็นทางเลือกที่ดีสำหรับองค์กรที่ต้องการพัฒนาระบบเทคโนโลยีสารสนเทศให้มีความทันสมัยและมีประสิทธิภาพสูง การนำ Kubernetes มาใช้ในองค์กรด้านสาธารณสุขเพื่อเพิ่มประสิทธิภาพ และรับมือกับความท้าทายที่เกิดขึ้นในยุคปัจจุบันได้อย่างมีประสิทธิภาพ

คำสำคัญ: เทคโนโลยีบริหารจัดการคอนเทนเนอร์; โครงสร้างพื้นฐานทางเทคโนโลยี; คอนเทนเนอร์

---

[1] คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล, ประเทศไทย, 10700
[1] Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok Thailand, 10700
[*] Corresponding author: e-mail: somchai.bua@mahidol.ac.th

วารสาร Mahidol R2R e-Journal ปีที่ 13 ฉบับที่ 1 ประจำเดือนมกราคม-เมษายน 2569
Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.

Abstract

Currently, healthcare management is characterized by complexity and diverse needs, and public health organizations confront challenges in managing rapidly evolving information technology systems. Kubernetes is an efficient and flexible container orchestration system, facilitates the enhancement of application and resource management within organizations effectively. It enables automatic scaling and resource management, effectively addressing the aforementioned challenges.

Kubernetes helps reduce the time required for the development and delivery of applications, while enhancing the reliability and stability of systems through the management of application recovery from system operational errors. Its scalability and efficient resource management capabilities ensure smooth and more efficient operations within an organization.

Therefore, Kubernetes emerges as an efficient tool for container management within organizations, augmenting operational efficiency and flexibility. It provided a viable option for organizations seeking to modernize and optimize their information technology systems. Implementing Kubernetes in healthcare organizations can improve efficiency and effectively address the challenges.

Keywords: Kubernetes; Infrastructure management; Container

บทนำ

เทคโนโลยีสารสนเทศ (Information Technology) มีบทบาทสำคัญในการดำเนินงานด้านการดูแลสุขภาพที่มีความซับซ้อนและหลากหลายมากขึ้น การจัดการและการใช้ทรัพยากรด้านเทคโนโลยีอย่างมีประสิทธิภาพจึงกลายเป็นความท้าทายสำคัญสำหรับงานด้านการแพทย์และสาธารณสุข

การนำเทคโนโลยี Kubernetes มาช่วยเสริมความแข็งแกร่งของการให้บริการด้านเทคโนโลยีสารสนเทศยิ่งทำให้มีความพร้อมในการให้บริการ ความรวดเร็ว ถูกต้องยิ่งขึ้น เนื่องจากคุณสมบัติของ Kubernetes ที่ทำงานลักษณะของคลัสเตอร์มีความสามารถในการเพิ่มลดคอนเทนเนอร์ ให้เหมาะสมกับการใช้งานรวมไปถึงการเฝ้าตรวจสอบให้แอปพลิเคชันมีความพร้อมใช้งานอยู่ตลอดเวลาในรูปแบบอัตโนมัติ ส่งผลให้ระบบเทคโนโลยีสามารถรองรับการให้บริการด้านสุขภาพ อย่างมีประสิทธิภาพมากยิ่งขึ้น

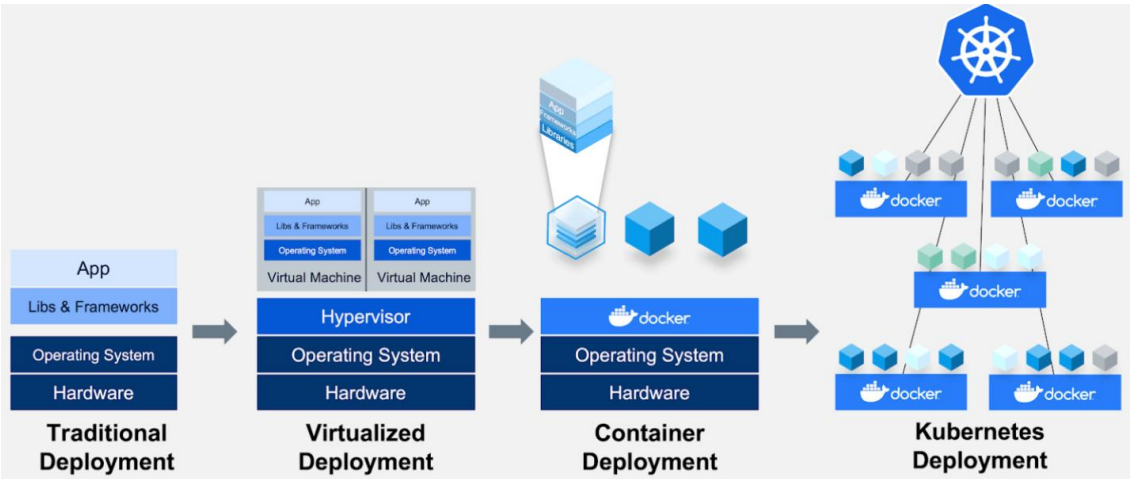
ซึ่งปัจจุบันการพัฒนาแอปพลิเคชันลักษณะไมโครเซอร์วิสและทำงานอยู่ในรูปแบบคอนเทนเนอร์มีแนวโน้มการใช้งานที่มากขึ้น เช่น ระบบที่สนับสนุนในการรักษา ระบบเก็บข้อมูลผู้รับบริการ ต้องมีความรวดเร็ว ความถูกต้องและพร้อมให้บริการตลอดเวลา เทคโนโลยี Kubernetes จึงเป็นทางเลือกที่ดีสำหรับองค์กรที่ต้องการพัฒนาและปรับปรุงระบบเทคโนโลยีสารสนเทศของตนให้ทันสมัยและมีประสิทธิภาพยิ่งขึ้น และการศึกษาทำความเข้าใจเทคโนโลยีที่นำมาใช้ เป็นอีกหนึ่งปัจจัยที่สำคัญซึ่งได้ระบุไว้ในบทความฉบับนี้

ความเป็นมาและความหมายของ Kubernetes

Kubernetes เป็นซอฟต์แวร์โอเพนซอร์สที่ได้รับแรงบันดาลใจจากซอฟต์แวร์ที่ชื่อว่า Borg Cluster Manager หรือเรียกสั้น ๆ ว่า Borg เป็นระบบจัดการคลัสเตอร์ที่พัฒนาโดย Google ร่วมกันพัฒนา Kubernetes ซึ่งเริ่มแรกได้ตั้งชื่อรหัสเรียก Kubernetes ว่า Project 7 และได้เปลี่ยนการเรียก Project 7 เป็น Kubernetes พร้อมออกแบบสัญลักษณ์เป็นพวงมาลัยเรือเจ็ดก้าน Kubernetes พัฒนาด้วยภาษา Go เป็นคำที่มีรากฐานมาจากภาษากรีกโบราณที่ หมายถึง ผู้ควบคุม หรือ นักบังคับเรือ และมีคำเรียกย่อว่า K8s ที่เกิดจากการนับตัวอักษร 8 ตัวอักษร ระหว่าง K กับ s Kubernetes ได้เปิดตัวอย่างเป็นทางการเมื่อปี 2014 Version 1.0 ได้รับการเผยแพร่ในวันที่ 21 กรกฎาคม 2558 (The Kubernetes Authors, 2025)

วิวัฒนาการของวิธีการติดตั้งแอปพลิเคชันในระบบคอมพิวเตอร์

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 1 แสดงวิวัฒนาการ Deploy Application (Kruparaju, 2024)

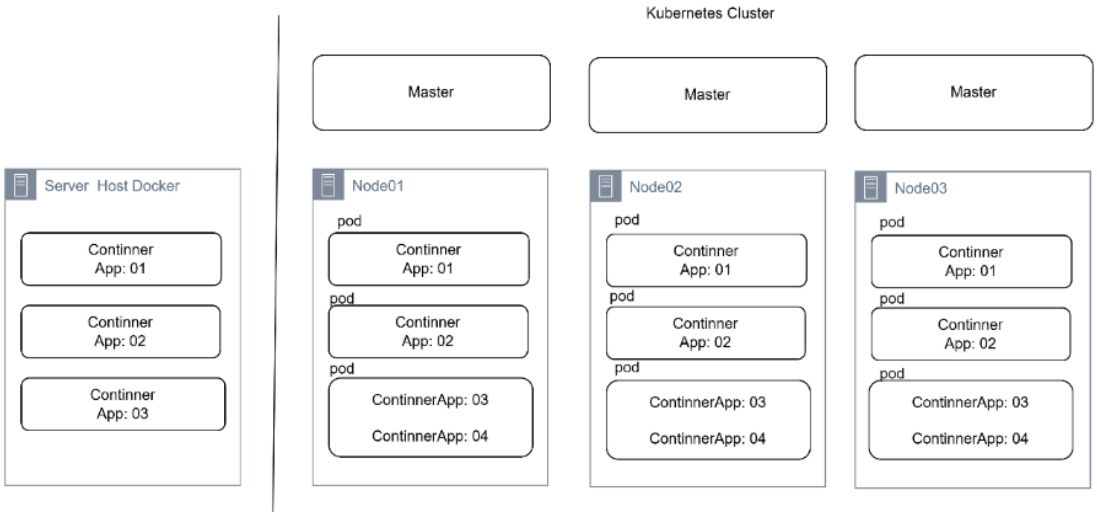
การประยุกต์ใช้ระบบการติดตั้งแบบดั้งเดิม (Traditional Deployment) เป็นการติดตั้งแอป-พลิเคชัน บนเครื่องแม่ข่ายทางกายภาพ (physical server) โดยตรง ซึ่งมีข้อจำกัดในการกำหนดขอบเขตของทรัพยากร เช่น หน่วยประมวลผล หน่วยความจำสำรอง และฮาร์ดดิสก์ การติดตั้งแบบนี้ทำให้องค์กรต้องจัดเตรียมทรัพยากรของเครื่องแม่ข่ายเพิ่มเติมอย่างไม่จำกัดทำให้ใช้งบประมาณปริมาณมาก

การประยุกต์ใช้การติดตั้งแบบเสมือนจริง (Virtualized Deployment) เป็นแนวทางที่มีประสิทธิภาพสูงกว่ายุคการติดตั้งแบบดั้งเดิม ต้องพึ่งพาเครื่องแม่ข่ายทางกายภาพเพียงอย่างเดียว ในยุคการติดตั้งแบบเสมือนจริงสามารถสร้างเครื่องแม่ข่ายได้หลาย ๆ เครื่องบนฮาร์ดแวร์เดียวกันและจัดแบ่งทรัพยากรกันอย่างมีประสิทธิภาพยิ่งขึ้น

การประยุกต์ใช้การติดตั้งแบบคอนเทนเนอร์ (Container Deployment) มีลักษณะที่คล้ายคลึงกับเทคโนโลยีการจำลองเครื่องแม่ข่าย Virtualization Technology ยุคก่อนหน้า แต่มีความแตกต่างที่คอนเทนเนอร์จะใช้ระบบปฏิบัติการร่วมกัน ซึ่งช่วยลดทรัพยากรที่ต้องใช้ในการติดตั้งระบบปฏิบัติการลง คอนเทนเนอร์ไม่จำเป็นต้องติดตั้งระบบปฏิบัติการภายในตัวเอง คอนเทนเนอร์จะมีไฟล์ระบบ filesystem CPU memory process และการกำหนดค่าที่จำเป็นสำหรับการรันแอปพลิเคชัน ทำให้คอนเทนเนอร์มีความเป็นอิสระจากฮาร์ดแวร์และระบบปฏิบัติการ

กาประยุกต์ใช้เทคโนโลยี Kubernetes บริหารจัดการคอนเทนเนอร์ ผลการสำรวจโดยบริษัท Docker พบผู้พัฒนากว่า 80% ใช้ Container ในการพัฒนาแอปพลิเคชัน ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล มีแนวโน้มการใช้งานคอนเทนเนอร์ที่เพิ่มขึ้น ดังนั้นการใช้ Kubernetes ในการช่วยบริหารจัดการคอนเทนเนอร์ จึงเป็นทางเลือกที่ดีเนื่องจาก Kubernetes มีคุณสมบัติหลัก ได้แก่ การปรับขนาดในลักษณะแนวนอนอัตโนมัติ การรีสตาร์คอนเทนเนอร์ที่ล้มเหลว และมีกระบวนการตรวจสอบสถานะการทำงานของคอนเทนเนอร์ โดยอัตโนมัติเพื่อให้แน่ใจว่าคอนเทนเนอร์ยังคงทำงานได้ตามปกติ รูปที่ 2 เปรียบเทียบเมื่อใช้ Kubernetes cluster บริหารจัดการคอนเทนเนอร์ในลักษณะของคลัสเตอร์ จะสามารถกำหนดให้คอนเทนเนอร์ที่รองรับการใช้งานมีมากกว่า 1 คอนเทนเนอร์ และ Kubernetes จะจัดการกระจายคอนเทนเนอร์ไปทำงานคนละ node ดังนั้นเมื่อแอปพลิเคชันบนคอนเทนเนอร์จาก node ใด node หนึ่งเสียหายก็ยังมีคอนเทนเนอร์ใน nodes อื่น ๆ รองรับการใช้งานและ Kubernetes ก็จะสร้างคอนเทนเนอร์ที่หยุดการทำงานขึ้นมาใหม่เท่ากับค่าที่กำหนดไว้และสามารถตั้งเงื่อนไขการ เพิ่ม ลด ปริมาณของคอนเทนเนอร์ ตามปริมาณการใช้งานได้อย่างมีประสิทธิภาพจะแตกต่างกับคอนเทนเนอร์ที่ทำงานบน Host Docker เมื่อ Host Docker เสียหายทำให้คอนเทนเนอร์เสียหายทั้งหมด (Flora et al., 2022; Floyd et al., 2024)

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 2 เปรียบเทียบคอนเทนเนอร์ ระหว่าง Host Docker กับ Kubernetes

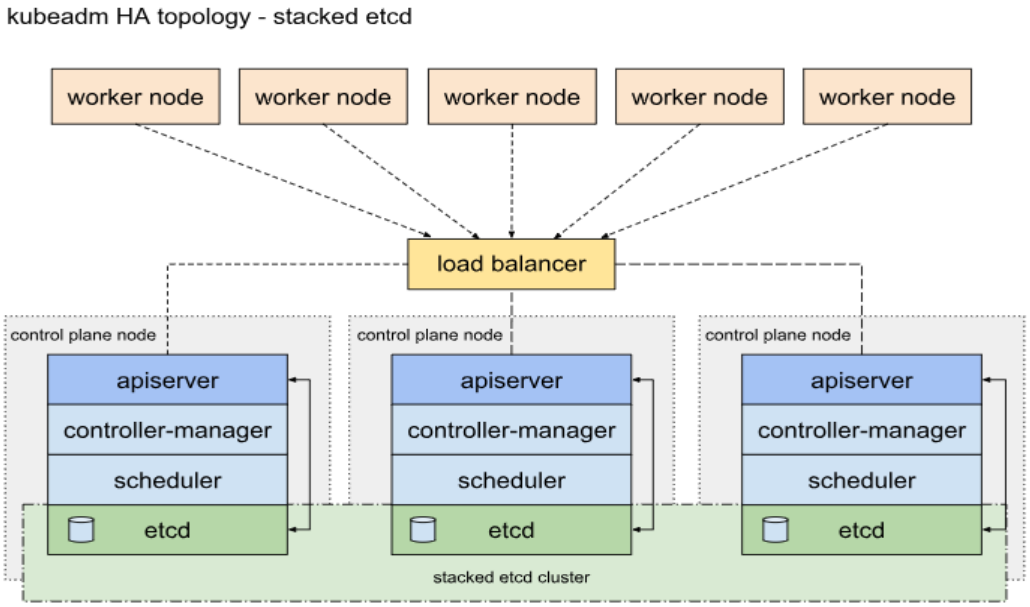
องค์ประกอบสำหรับการทำงานของ Kubernetes Cluster

การนำเทคโนโลยี Kubernetes มาจัดการคอนเทนเนอร์ ในองค์กรการศึกษาและทำความเข้าใจองค์ประกอบของ Kubernetes จึงมีความสำคัญเป็นอย่างมาก ซึ่งเป็นพื้นฐานที่จำเป็นในการเรียนรู้และการใช้งานอย่างมีประสิทธิภาพ ความเข้าใจสถาปัตยกรรมของระบบเป็นสิ่งสำคัญ ดังนั้นในบทความนี้จะนำเสนอองค์ประกอบพื้นฐานของ Kubernetes cluster จะแบ่งออกเป็น Control plane และ Worker node ดังนี้

1. Control plane หรือ Master node ทำหน้าที่บริหารจัดการ Kubernetes cluster และควรมีการกำหนดเป็นแบบ Multiple control plane ซึ่งประกอบด้วยส่วนประกอบสำคัญ

1.1 Etcd เป็นพื้นที่จัดเก็บข้อมูลชนิด Key-value ที่มีความสำคัญสำหรับการบริหารจัดการ Kubernetes cluster โดยเฉพาะการเก็บข้อมูล Configuration การกำหนด etcd ให้มีมากกว่า 1 ด้วย Stacked etcd Topology เป็นวิธีการที่ใช้ในการกระจายพื้นที่เก็บข้อมูลไปยังแต่ละ Control plane โดย etcd จะถูกกระจายไปยังทุก Control plane โดยมีการสื่อสารกันระหว่างพื้นที่เก็บข้อมูลในแต่ละ Control plane เพื่อให้ข้อมูลมีค่าเท่ากันทุกที่อัลกอริทึมที่ใช้ในการรักษาความถูกต้องและการเป็นไปตามกฎกติกาของข้อมูลใน etcd คือ Raft Consensus Algorithm โดยที่มีตัวกระจายโหลด ทำหน้าที่เป็นตัวกลางในการส่งคำขอไปยังแต่ละ Control Plane (Kubernetes, 2025)

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 3 แสดง HA topology -Stacked etcd (The Kubernetes Authors, 2025)

1.2 Kube-Scheduler เป็นองค์ประกอบในระบบ Kubernetes cluster ทำหน้าที่ในการเลือก Worker node ที่เหมาะสมสำหรับการรันคอนเทนเนอร์ โดยกระบวนการเลือกจะพิจารณาจากหลายปัจจัย เช่น สถานะของ Worker node Workload การตั้งค่า Affinity Anti-affinity Node Selectors เป็นต้น

1.3 Kube–Api-Server ทำหน้าที่เป็น ตัวกลางสำหรับการสื่อสารภายใน Kubernetes cluster ผ่านตัวกลางแลกเปลี่ยนข้อมูล (RESTful API) จะตรวจสอบความถูกต้องของคำขอต่าง ๆ ที่ผู้ใช้ร้องขอ ก่อนที่จะส่งคำขอเหล่านั้นไปยังส่วนอื่น ๆ ในระบบ เช่น การเพิ่ม ลบ แก้ไข Deployment Service Volume Network เป็นต้น

1.4 Cloud-Controller Manager (CCM) เป็นองค์ประกอบใน Kubernetes ที่จะใช้งานเมื่อระบบ Kubernetes ถูกปรับใช้อยู่บน Cloud Provider ที่ได้รับการสนับสนุนมีการทำงานที่คล้ายคลึงกับ Kube-Controller Manager แต่จะเน้นการจัดการกับ Cloud Provider API ของแต่ละผู้ให้บริการ

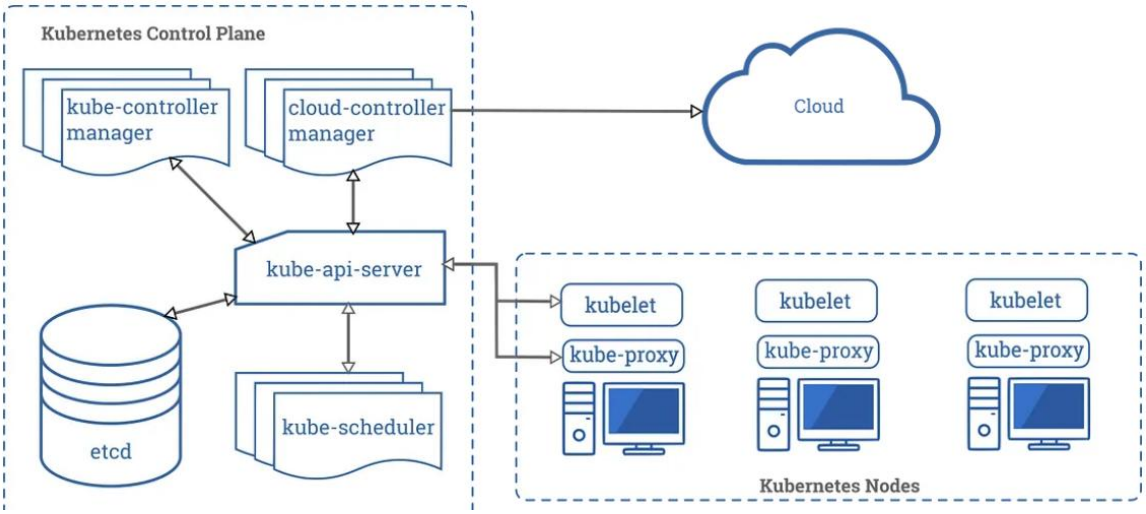
1.5 Kube-Controller Manager (KCM)) เป็นตัวจัดการควบคุม Kubernetes cluster ที่อยู่บน On-Premises ซึ่งจะประกอบด้วย Node controller Replication controller Endpoints controller Service Account & token controllers

2. Worker nodes เป็นส่วนที่รองรับการทำงานของแอปพลิเคชันการกำหนด Worker nodes บน Kubernetes cluster ควรมีมากกว่า 2 worker nodes เพื่อรองรับกรณี Worker node ตัวใดตัวหนึ่งหยุดให้บริการ จะมีส่วนประกอบที่สำคัญ 2 ส่วนคือ

2.1 Kubelet ทำหน้าที่เชื่อมต่อไปยังคอนเทนเนอร์รันไทม์ (Container Runtime) และ Kubelet จะทำหน้าที่คอยตรวจสอบการทำงานของ pods เช่น Start Delete Update เป็นต้น

2.2 Kube-Proxy มีหน้าที่จัดการการสื่อสารทั้งจากภายนอกและภายใน Worker node และกำหนดกฎการสื่อสาร โดย Kube-Proxy มีโหมดการทำงาน 3 โหมด ซึ่งคือ Iptables IPVS และ KernelSpace (Chinnam, 2024; Kumar, 2024)

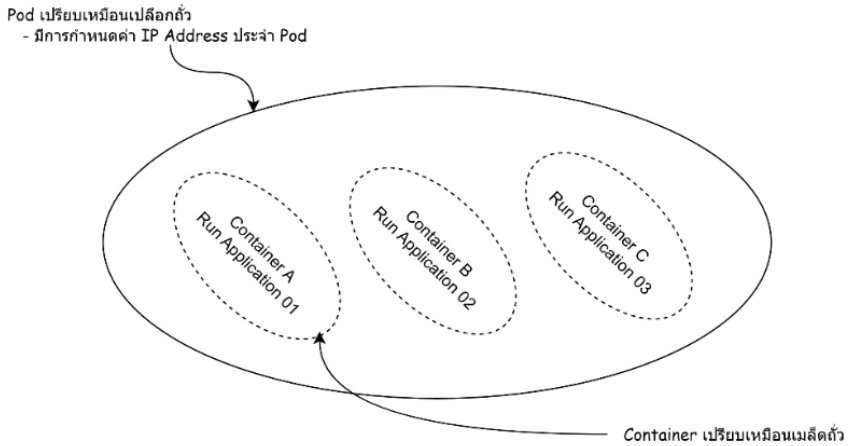
Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 4 แสดงองค์ประกอบของ Kubernetes Cluster (Pob, 2021)

องค์ประกอบพื้นฐานการทำงานของแอปพลิเคชันภายในของ Kubernetes Cluster

1. Pod คือส่วนที่เล็กที่สุดของระบบ Kubernetes เปรียบดังถั่วลิสง เปลือกถั่วคือ pod และเมล็ดถั่ว คือ container ที่อยู่ภายใน pod สามารถมีได้มากกว่า 1 container ซึ่งส่วนใหญ่จะใช้ 1 pod ต่อ 1 คอนเทนเนอร์ เพราะการบริหารจัดการที่ง่าย pod จะมีเลขประจำ pod สำหรับการสื่อสารเรียกว่า IP Address

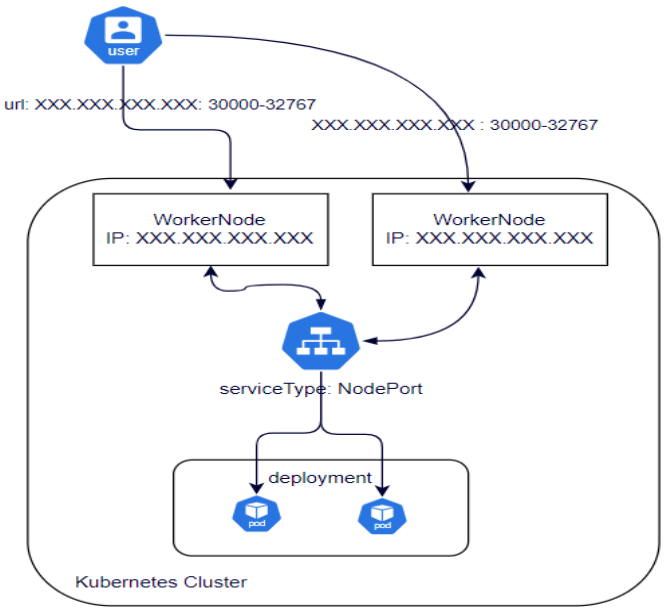


รูปที่ 5 แสดงเปรียบเทียบ Pod ใน Kubernetes

2. Service ตัวกลางการสื่อสารภายใน Kubernetes cluster จะเกิดขึ้นผ่านการใช้งาน Service ซึ่งเป็นส่วนสำคัญในการเชื่อมต่อและสื่อสารระหว่างกัน จะทำหน้าที่เป็นตัวกลางในการแจกจ่ายการทำงานไปยัง pods ต่าง ๆ และสร้างช่องทางสื่อสารระหว่างแอปพลิเคชันกับโลกภายนอก Kubernetes cluster โดย Service แบ่งออกเป็น 4 ประเภท ดังนี้

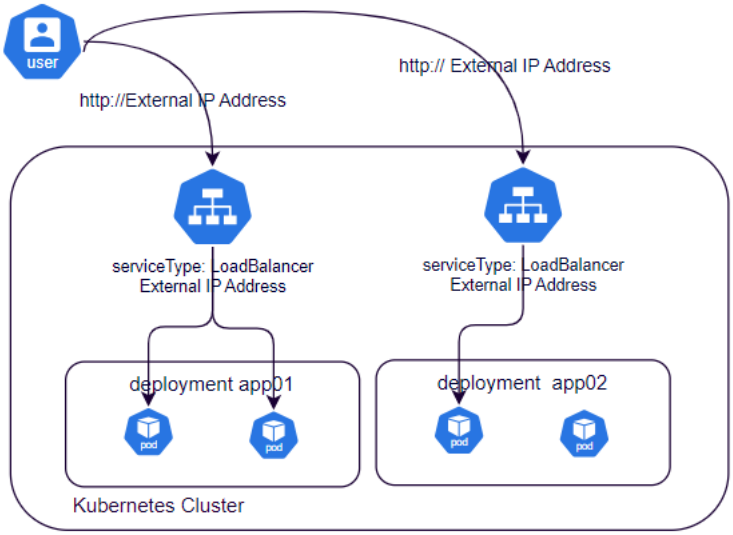
2.1 NodePort เป็น Service ที่สามารถเรียกใช้งานจากภายนอกโดยอาศัย Public IP ของ Worker node จะใช้ TCP port แบบสุ่มระหว่าง 30000-32767 จะมีข้อพึงระวังหากหมายเลขไอพีของ Worker node เปลี่ยนจะทำให้ผู้ใช้ไม่สามารถเรียกแอปพลิเคชันได้

วารสาร Mahidol R2R e-Journal ปีที่ 13 ฉบับที่ 1 ประจำเดือนมกราคม-เมษายน 2569
Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 6 แสดงการสื่อสารของ ServiceType NodePort

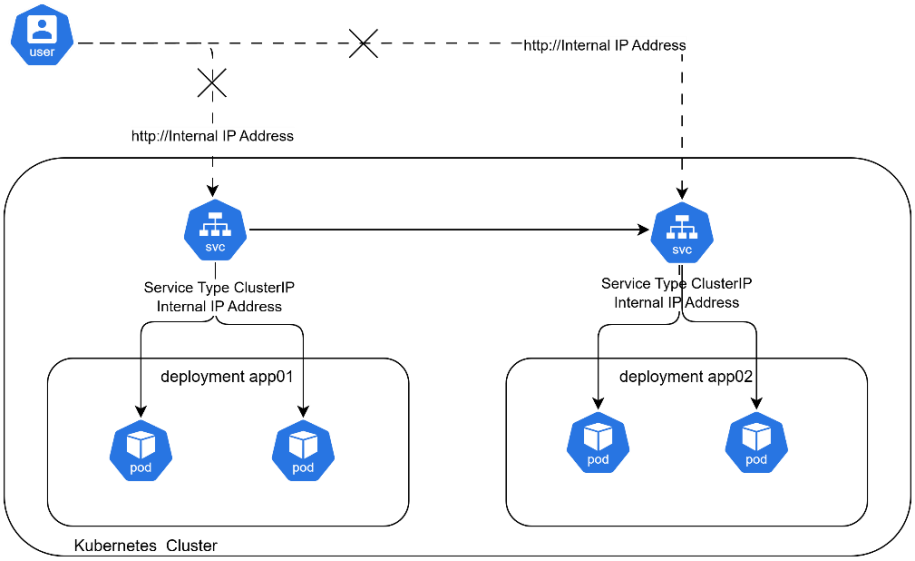
2.2 LoadBalancer อำนวยความสะดวกในการสื่อสารจากภายนอกไปยังแอปพลิเคชันภายใน Kubernetes cluster ซึ่ง Service ประเภทนี้จะกำหนดแบบ Fixed public IP Address ทำให้ Client สามารถเข้าถึงแอปพลิเคชันได้อย่างปลอดภัยและมีประสิทธิภาพผ่านช่องทางการสื่อสารประเภทนี้



รูปที่ 7 แสดงการสื่อสาร ServiceType LoadBalancer

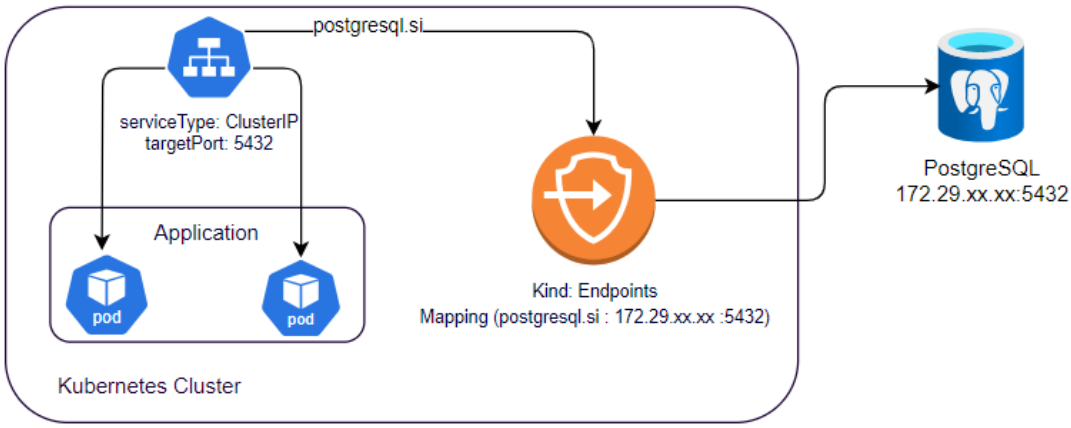
2.3 ClusterIP สำหรับการสื่อสารภายใน Kubernetes cluster รูปที่ 8 แสดงให้เห็นว่าผู้ใช้ไม่สามารถเข้าถึงแอปพลิเคชันผ่านช่องทางนี้ได้โดยตรง ซึ่ง Service ประเภทนี้จะมีการสื่อสารภายในเท่านั้น ไม่สามารถเรียกใช้งานจากภายนอกได้ แต่จะใช้วิธีสร้าง Service ประเภท Ingress เป็นตัวกลางระหว่างผู้ใช้งานกับ แอปพลิเคชันภายในนอกจากนี้ Ingress ยังมีคุณสมบัติกำหนดเส้นทางการสื่อสารจากผู้ใช้งานไปยังแอปพลิเคชันที่อยู่ในคอนเทนเนอร์และทำให้การสื่อสารมีความปลอดภัยยิ่งขึ้น

วารสาร Mahidol R2R e-Journal ปีที่ 13 ฉบับที่ 1 ประจำเดือนมกราคม-เมษายน 2569
Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 8 แสดงการสื่อสาร ServiceType ClusterIP

2.4 **External Service** เป็น Service ที่ทำการแปลงจาก IP Address ให้เป็นชื่อที่กำหนดเพื่อไม่ให้ส่งผลกระทบต่อแอปพลิเคชันเมื่อต้นทางทำการเปลี่ยนสภาพแวดล้อมใหม่จากรูปที่ 9 แสดงให้เห็นถึงการเรียกใช้งานฐานข้อมูล PostgreSQL จากภายใน Kubernetes cluster ผ่าน Kind: Endpoints ชื่อ postgresql.si การใช้ External Service จะช่วยให้การเชื่อมต่อมีความยืดหยุ่นและมั่นคง เนื่องจากไม่จำเป็นต้องระบุ IP Address ในการสื่อสารแต่จะใช้เป็นชื่อที่ถูกกำหนดไว้ (VMware, n.d.)



รูปที่ 9 แสดงการสื่อสารของ External Service

3. Ingress: ทำหน้าที่เป็นจุดเริ่มต้นสำหรับการรับส่งข้อมูลขาเข้า Kubernetes cluster โดยจะมีวิธีการจัดการและควบคุมการกำหนดเส้นทางคำขอภายนอกไปยังบริการหรือทรัพยากรภายในที่เหมาะสม Ingress มีหน้าที่รับผิดชอบในการวิเคราะห์คำขอที่เข้ามาตามกฎที่กำหนดไว้และส่งต่อไปยังแอปพลิเคชันปลายทางที่ถูกต้องภายใน จากรูปที่ 10 แสดงให้เห็นคุณสมบัติการกำหนดเข้ารหัสรับส่งข้อมูลที่เป็นคุณสมบัติของ Ingress และข้อมูลคำขอจะส่งไปยังคอนเทนเนอร์ตามกฎที่กำหนดไว้ สามารถแบ่งการสื่อสารใน Kubernetes cluster ออกเป็น 2 ส่วนคือ การสื่อสารภายในและภายนอก (Barnard, 2024)

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.

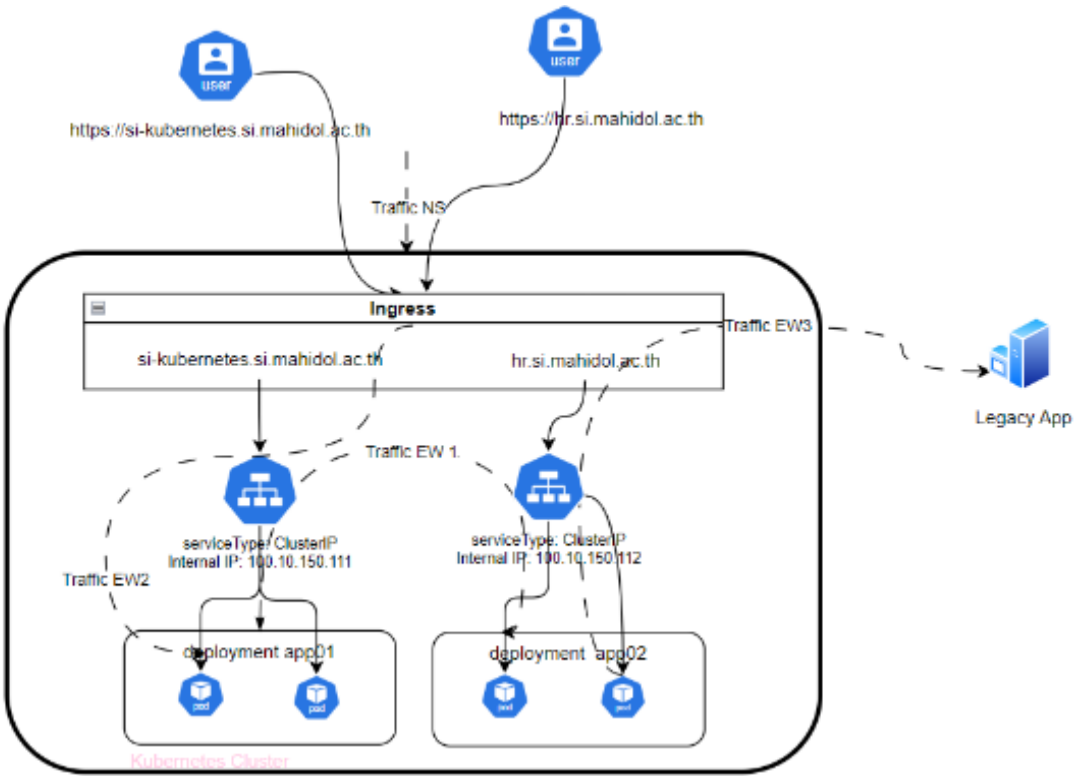
1. การสื่อสารจากภายนอกไปยัง Kubernetes cluster การเชื่อมต่อที่มาจากภายนอก Kubernetes เช่น ผู้ใช้เรียกใช้งานไปยังแอปพลิเคชันภายใน Kubernetes cluster ผ่าน Ingress

2. การสื่อสารภายใน Kubernetes cluster จะเป็นการสื่อสารภายในและการสื่อสารจากแอปพลิเคชันภายในไปยังแอปพลิเคชันภายนอก แบ่งออกเป็น 3 ลักษณะดังนี้

2.1 การสื่อสารระหว่าง pod หนึ่งไปยังอีก pod หนึ่งที่อยู่ภายใต้ Service ที่ต่างกัน

2.2 การสื่อสารจาก Ingress ผ่าน Service ไปยัง pod ตามกฎที่ได้กำหนดไว้

2.3 การสื่อสารจากแอปพลิเคชันภายใต้ pod ไปยังแอปพลิเคชันที่อยู่ภายนอก Kubernetes cluster



รูปที่ 10 แสดงการสื่อสาร Kubernetes cluster

ข้อดีข้อเสียการใช้งาน Kubernetes

ข้อดี Kubernetes เป็นเทคโนโลยีสำคัญที่ใช้สำหรับควบคุมดูแลคอนเทนเนอร์ให้สามารถทำงานได้อย่างมีประสิทธิภาพและมีความยืดหยุ่นสูง บทความฉบับนี้ได้สรุปข้อดีไว้ดังนี้

1. **ความสามารถในการขยายขนาด** สามารถจะปรับเพิ่ม ลดขนาดของทรัพยากรให้เหมาะสมกับปริมาณการใช้งานได้ตามความต้องการ ทำให้มีความสามารถในการรองรับการใช้งานที่มากขึ้น
2. **ติดตั้งในสภาพแวดล้อมที่หลากหลาย** ด้วยความสามารถในการแบ่งแยกส่วนประกอบของระบบซึ่งทำงานบนระดับแต่ละชั้น (layer) โดยแต่ละชั้นจะมีหน้าที่และความสำคัญที่แตกต่างกันของ Kubernetes จะทำให้สามารถทำงานได้ทั้ง On-Cloud หรือ On-Premises
3. **ความรวดเร็วในการส่งมอบ** เมื่อนำ Kubernetes มาใช้ในองค์กรเพื่อเพิ่มประสิทธิภาพกระบวนการทำงานจะมีความแตกต่างจากเดิม ซึ่งกระบวนการทำงานแบบ (agile) ที่มีความคล่องตัวจะเป็นส่วนสำคัญในการสร้างและการพัฒนาซอฟต์แวร์ส่งผลให้มีความคล่องตัวในการส่งมอบแอปพลิเคชันอย่างรวดเร็วปลอดภัย และมีความยืดหยุ่นมากยิ่งขึ้น

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.

4. **ชุมชนการแลกเปลี่ยน** Kubernetes Slack เป็นหนึ่งในช่องทางที่สำคัญในการสร้างและร่วมกันพัฒนาความรู้ของชุมชน Kubernetes โดยสามารถเข้าร่วมในช่องทางที่เกี่ยวข้องกับความสนใจหรือปัญหาที่เกิดขึ้น เช่น ช่องทางสำหรับการถาม ตอบ แบ่งปันข่าวสาร และการแลกเปลี่ยนประสบการณ์ในการใช้งาน Kubernetes
5. **เลือกสภาพแวดล้อมที่เหมาะสม** สำหรับการใช้งาน Kubernetes จะช่วยให้การบริหารจัดการคอนเทนเนอร์อย่างมีประสิทธิภาพ โดยมีคุณสมบัติตรวจสอบและกำหนด Worker node ให้กับ pod ซึ่งจะทำให้แอปพลิเคชันมีความพร้อมและทำงานได้ตลอดเวลา
6. **ระบบมีความพร้อมให้บริการ** Kubernetes มีคุณสมบัติตรวจสอบดูแลคอนเทนเนอร์ให้ทำงานอยู่ได้ตามเงื่อนไขที่กำหนดไว้ หากมีคอนเทนเนอร์ที่ทำงานผิดพลาดหรือหยุดการทำงานจะมีคำสั่งให้สร้างคอนเทนเนอร์ใหม่ขึ้นมาทดแทนคอนเทนเนอร์ที่เสียหายโดยอัตโนมัติ
7. **ประหยัดต้นทุน** ข้อดีที่หลาย ๆ องค์กรคาดหวัง คือ การลดค่าใช้จ่าย Kubernetes เป็นเครื่องมือบริหารจัดการคอนเทนเนอร์ที่ทำให้แอปพลิเคชันพร้อมให้บริการ มีความรวดเร็วของการขึ้นใช้งาน และมีความยืดหยุ่นต่อการใช้งาน ดังนั้นการลดต้นทุนเมื่อใช้งาน Kubernetes จะต้องกำหนดการปรับขนาดของ pod ในแนวระนาบ หรือการปรับขนาดอัตโนมัติของ Kubernetes cluster รวมถึงการการสร้าง Kubernetes cluster ที่มีขนาดใหญ่และนำระบบที่คล้ายกันมาอยู่ด้วยกัน เพื่อลดการใช้ทรัพยากรและการกำหนดขีดจำกัดของทรัพยากร CPU Memory ของคอนเทนเนอร์ให้เพียงพอ เพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพ (Chinnam, 2024)

ข้อเสียของการใช้งาน Kubernetes บทความฉบับนี้ ยกตัวอย่างข้อเสียของการนำ Kubernetes มาใช้ภายในองค์กรดังนี้

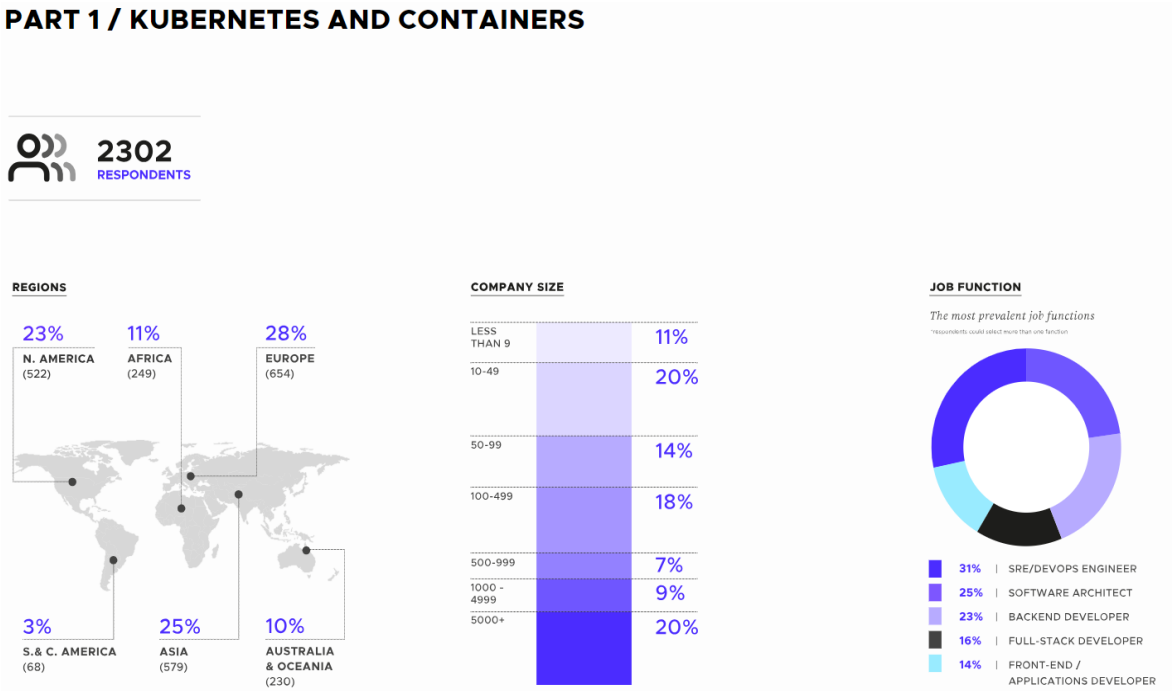
1. **ใช้เวลาในการเรียนรู้ที่ยาวนาน** เนื่องจากเทคโนโลยี Kubernetes เป็นเทคโนโลยีที่ใหม่สำหรับบางองค์กรและเป็นเทคโนโลยีที่มีความซับซ้อน ดังนั้นเมื่อนำมาใช้งานในองค์กรทำให้บุคลากรต้องศึกษาค้นคว้าการทำงานของ Kubernetes ใช้เวลาทุ่มเทกับการเรียนรู้และทดลองเป็นเวลานานเพื่อให้เข้าใจองค์ประกอบและการทำงานของ Kubernetes อย่างถ่องแท้
2. **การตรวจสอบและแก้ไขปัญหาต้องใช้ผู้เชี่ยวชาญ** ความซับซ้อนในการทำงานของ Kubernetes ทำให้การดีบั๊กและการแก้ไขปัญหาต้องอาศัยผู้เชี่ยวชาญและสามารถตรวจสอบวิเคราะห์ปัญหาใน Kubernetes ได้อย่างรวดเร็วและมีประสิทธิภาพ
3. **เกิดการเปลี่ยนแปลงรูปแบบการทำงาน** Kubernetes เป็นเทคโนโลยีที่ต้องอาศัยองค์ความรู้หลาย ๆ ด้าน เช่น การดูแลเครื่องแม่ข่าย ความรู้ด้านเครือข่าย ความรู้ด้านความปลอดภัยของข้อมูล รวมถึงความรู้ด้านพัฒนาซอฟต์แวร์ และการขึ้นใช้งานแอปพลิเคชันจะมีความเป็นอัตโนมัติมากขึ้น การเลือกหรือวางแนวทางสำหรับการใช้งานแอปพลิเคชันที่เกี่ยวข้องมาเสริมให้การทำงานมีประสิทธิภาพมากขึ้น ทำให้บุคลากรที่เกี่ยวข้องต้องปรับตัวในระยะเริ่มต้นพอสมควร
4. **ไม่เหมาะสำหรับองค์กรที่มีขนาดเล็ก**: การใช้งาน Kubernetes เหมาะสมกับองค์กรขนาดใหญ่หรือองค์กรที่มีการขยายขนาดขององค์กรเนื่องจากเป็นเทคโนโลยีที่ต้องอาศัยกลุ่มคนมาร่วมกันสร้างขึ้นมาและในระยะเริ่มต้นองค์กรต้องสนับสนุนค่าใช้จ่ายในการอบรมเพื่อให้การใช้งานเป็นไปอย่างถูกต้อง

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.

5. **ต้องมีการวางแผนล่วงหน้าอย่างรอบคอบ**: เมื่อองค์กรนำ Kubernetes มาใช้ควบคุมดูแลคอนเทนเนอร์จะต้องทำความเข้าใจและวางรูปแบบการทำงานของแอปพลิเคชันการสื่อสารรวมถึงการกำหนดความปลอดภัย ดังนั้นการออกแบบอย่างรอบคอบเป็นสิ่งที่สำคัญในการนำแอปพลิเคชันมาใช้ภายใต้ Kubernetes

แนวโน้มการใช้งาน Kubernetes

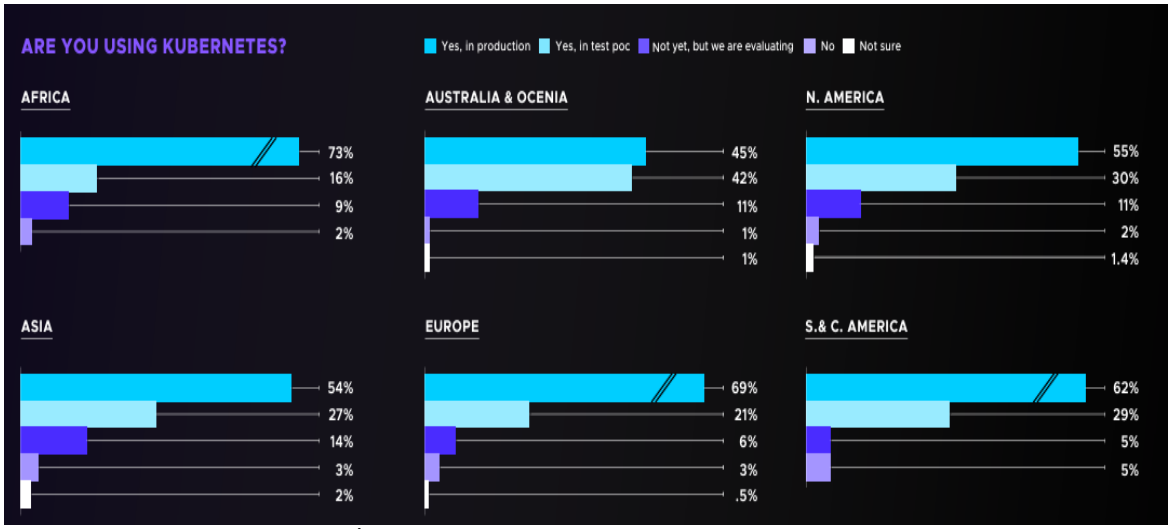
จากผลการสำรวจของ CNCF (2021) ทำการสำรวจการใช้งาน ของ Kubernetes และคอนเทนเนอร์ของแต่ละทวีปผลการสำรวจทวีปที่ใช้ Kubernetes มากสุดคือ N.America 23% น้อยสุดคือ Australia & oceania ส่วน Asia จะอยู่ที่ 25% ตามรูปที่ 11



รูปที่ 11 แสดงสถิติการใช้งาน Kubernetes และ คอนเทนเนอร์ (CNCF, 2021)

กล่าวได้ว่า Kubernetes ได้ก้าวผ่านการยอมรับและกลายเป็นเทคโนโลยีกระแสหลักของโลกไปแล้วจากข้อมูลของผู้ตอบแบบสอบถามของ CNCF พบว่า 96% องค์กรกำลังประเมินและใช้งาน Kubernetes และจากผลการสำรวจพบว่าแต่ละทวีปมีการนำ Kubernetes มาใช้งานเป็น Production เป็นส่วนใหญ่ เช่น AFRICA 73%, AUSTRALIA & OCENIA 45%, N.AMERICA 55%, ASIA 54%, EUROPE 69%, S&C.AMERICA 62% ตามรูปที่ 12

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 12 แสดงการใช้งาน Kubernetes (CNCF, 2021)

ความท้าทายของการใช้งาน Kubernetes

Kubernetes เป็นเครื่องมือที่มีความซับซ้อนและใช้ความพยายามในการศึกษาเรียนรู้ เนื่องจากมีโครงสร้างที่ซับซ้อนต้องมีความเข้าใจหลาย ๆ ด้าน เช่น Network Server Application Security

1. ความซับซ้อนและโค้ดเบสที่ใหญ่ Kubernetes ถูกมองว่าเป็นเครื่องมือที่มีโค้ดเบสที่ใหญ่และซับซ้อน ซึ่งทำให้ยากที่จะเรียนรู้และเข้าใจได้เร็ว เอกสารประกอบการศึกษามีจำกัด ทำให้การนำทางสำหรับการใช้งานและพัฒนาทำได้ยากขึ้น

2. ใช้เวลาในการเรียนรู้ที่นาน การเรียนรู้ยาวนานเปรียบดังภูเขาที่สูงชันกว่าเทคโนโลยีอื่น ๆ โดยเฉพาะในการทำความเข้าใจส่วนประกอบต่าง ๆ เช่น pods services deployments volumes RBAC Authorization เป็นต้น

3. ข้อผิดพลาดที่อาจเกิดจากการกำหนดค่าอาจเกิดขึ้นได้ง่ายมากหากผู้ดูแลระบบไม่เข้าใจแนวคิดพื้นฐาน หรือโครงสร้างพื้นฐานของ Kubernetes

4. ความจำเป็นในการฝึกอบรม ทีมที่เริ่มใช้งาน Kubernetes มักประสบปัญหาหากไม่มีการฝึกอบรมที่เพียงพอการลงทุนเวลาในการเรียนรู้พื้นฐาน Kubernetes เป็นสิ่งสำคัญ

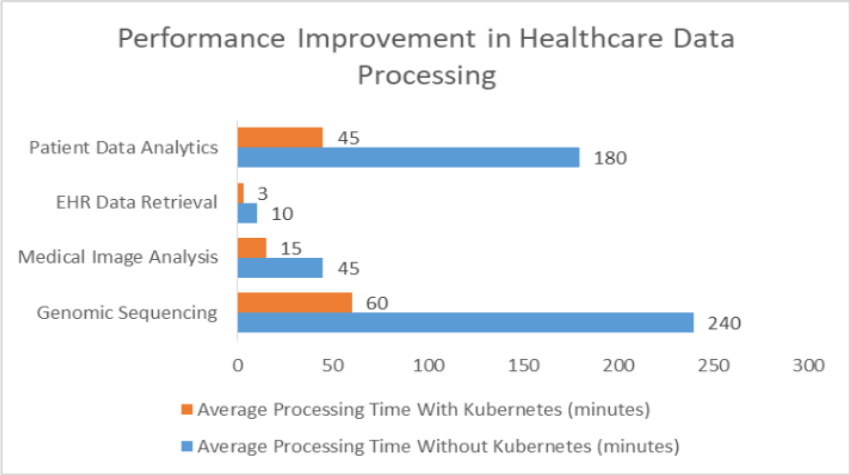
5. ปัญหาด้านความปลอดภัยและการดีบัก ความซับซ้อนของ Kubernetes สามารถนำไปสู่ช่องโหว่ด้านความปลอดภัยและปัญหาในการดีบักระบบยากยิ่งขึ้น เนื่องจาก Kubernetes เป็นระบบที่เปิดกว้างให้กำหนดค่าได้ (Sehgal, 2025)

ความจำเป็นของ Kubernetes ในงานด้านสาธารณสุข

โครงสร้างพื้นฐานด้านไอทีของการดูแลสุขภาพเป็นสิ่งจำเป็นอย่างยิ่ง โดยเฉพาะงานด้านสาธารณสุข ซึ่งต้องการความรวดเร็ว ความถูกต้องและความปลอดภัย Kubernetes จึงเป็นโซลูชันที่ดีเนื่องจากมีคุณสมบัติดังนี้

1. มีกลไกการรักษาตัวเอง Kubernetes จะมีกลไกในการโยกย้ายหรือสร้างใหม่ เมื่อมีคอนเทนเนอร์ หรือ Node เกิดความเสียหาย ช่วยลดการหยุดชะงักของการบริการด้านการดูแลสุขภาพ
2. การจัดการข้อมูลผู้ป่วยที่ละเอียดอ่อน Kubernetes มีระบบควบคุมการเข้าถึงตามบทบาท (RBAC) ช่วยให้องค์กรด้านการดูแลสุขภาพสามารถกำหนดนโยบายตามกฎระเบียบ เช่น HIPAA
3. Kubernetes โดดเด่นในการจัดการงานประมวลผลข้อมูลขนาดใหญ่ ซึ่งมีความสำคัญต่อแอปพลิเคชันด้านการดูแลสุขภาพที่ต้องการจัดการข้อมูลผู้ป่วยจำนวนมาก เช่น ภาพถ่ายทางการแพทย์
4. ความสามารถในการจัดสรรทรัพยากร โดยสามารถปรับแต่งการจัดสรรทรัพยากรได้อย่างละเอียดตามลำดับความสำคัญของแอปพลิเคชัน ซึ่งช่วยให้มั่นใจได้ว่าแอปพลิเคชันที่สำคัญจะได้รับทรัพยากรที่เพียงพอ (Chinnam, 2024)

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.



รูปที่ 13 ประสิทธิภาพในการประมวลผลข้อมูลด้านการดูแลสุขภาพด้วย Kubernetes

สรุป

ความน่าเชื่อถือ ความถูกต้อง ความรวดเร็ว และความพร้อมใช้งานของระบบถือเป็นแกนหลักในการดำเนินงานขององค์กร ซึ่งการเลือกเทคโนโลยีมาใช้งานเป็นสิ่งที่สำคัญอย่างยิ่ง เพื่อรองรับการพัฒนาแอปพลิเคชันในรูปแบบที่ติดตั้งบนคอนเทนเนอร์ที่มีแนวโน้มที่มากขึ้น

เทคโนโลยี Kubernetes เป็นเทคโนโลยีที่มีบทบาทในการบริหารจัดการคอนเทนเนอร์ให้มีความพร้อมใช้งาน ความปลอดภัย ความสามารถรองรับการใช้งานปริมาณมาก มีความยืดหยุ่นสามารถกำหนดเงื่อนไขการขยาย การลดปริมาณของคอนเทนเนอร์ให้เหมาะสมกับการใช้งานแบบอัตโนมัติและการใช้งาน Kubernetes จำเป็นอย่างยิ่งต้องทำความรู้จักความเป็นมาของ Kubernetes องค์ประกอบสำหรับการทำงานของ Kubernetes cluster ที่ได้กล่าวถึง Master node และ Worker node ซึ่งทำหน้ารองรับการทำงานของแอปพลิเคชัน นอกจากนี้ต้องมีความเข้าใจการทำงานของแอปพลิเคชัน รวมถึงการสื่อสารภายใน Kubernetes ที่อธิบายไว้ในบทความฉบับนี้ ซึ่งเป็นส่วนประกอบพื้นฐานที่สำคัญสำหรับการต่อยอดศึกษาและนำไปใช้งาน

อย่างไรก็ตามการนำเทคโนโลยี Kubernetes มาใช้งานเพื่อบริหารจัดการคอนเทนเนอร์ให้อยู่ในรูปแบบของคลัสเตอร์ จะมีความท้าทายที่หลากหลาย เช่น การศึกษาเรียนรู้ของผู้ที่เกี่ยวข้องเนื่องจากเป็นเทคโนโลยีที่มีความซับซ้อน จะต้องใช้เวลาในการเรียนรู้ที่ยาวนานเพื่อทำความเข้าใจ ส่วนที่สำคัญนโยบายขององค์กรก็อาจจะต้องปรับเพื่อให้สอดคล้องกับเทคโนโลยีที่นำมาใช้ และรูปแบบการพัฒนาแอปพลิเคชันที่จะเป็นไปในลักษณะอัตโนมัติมากยิ่งขึ้น

กิตติกรรมประกาศ

บทความทางวิชาการฉบับนี้สำเร็จได้ด้วยการสนับสนุนจากฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล ที่ได้สนับสนุนการเรียนรู้เทคโนโลยีสมัยใหม่ และทดสอบการใช้งาน จนกระทั่งสามารถเข้าใจกระบวนการทำงานของ Kubernetes ที่เป็นเครื่องมือบริหารจัดการ Infrastructure ที่ได้รับความนิยมในปัจจุบัน และผู้เขียนได้มีโอกาศเข้าร่วมเป็นส่วนหนึ่งของการพัฒนาระบบโดยใช้เครื่องมือ Kubernetes ทำให้เกิดองค์ความรู้ ความเข้าใจ ในการนำไปพัฒนาองค์กรต่อไป ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

เอกสารอ้างอิง

Barnard, P. (2024, March 21). *What is kubernetes Ingress?.* Kong. https://konghq.com/blog/learning-center/what-is-kubernetes-ingress

Chinnam, S. K. (2024). Enhancing patient care through Kubernetes-powered healthcare data management. *Journal for Research in Applied Science and Engineering Technology*, *12*(8), 861-864. https://doi.org/10.22214/ijraset.2024.63954

Received: June 21, 2024, Revised: April 9, 2025, Accepted: May 13, 2025
http://doi.org/10.14456/jmu.

CNCF. (2021). *Annual survey 2021*. https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR_FINAL-edits-15.2.21.pdf

Floyd, R., Wilson, J., & Diachkova, O. (2024, 1 March). *The docker state of application development report 2024*. Docker. https://www.docker.com/resources/2024-state-of-application-development-report/

Flora, J., Goncalves, P., Teixeira, M., & Antune, N. (2022). A study on the aging and fault tolerance of microservices in Kubernetes. *IEEE Access*, *10*, 132786-132799. https://doi.org/10.1109/ACCESS.2022.3231191

Kruparaju, K. (2024, 23 February). *Streamlining software deployment: A Journev from operating systems to kubernetes orchestration*. Medium. https://kallakruparaju.medium.com/streamlining-software-deployment-a-journey-from- operating-systems-to-kubernetes-orchestration-ae00b98e989b

Kumar, A. (2024, 8 October). *High availability in kubernetes| detailed guide*. K21Academy. https://k21academy.com/docker-kubernetes/high-availability-and-scalable-application-in- kubernetes

Pob Apiwat. (2564, 16 กรกฎาคม). *Kubernetes architecture 101*. Medium. https://medium.com/devspree/kubernetes-architecture-101-d7c68191e52d

The Kubernetes Authors. (2025). *Options for highly available topology.* Kubernetes Documentation. https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/ha-topology/

_____. (2025). *Overview.* Kubernetes Documentation. https://kubernetes.io/docs/concepts/overview/

Sehgal, J. (2025). Release engineering in the age of kubernetes: Managing deployments in containerized environments. *International Journal on Science and Technology (IJSAT)*, *16*(1), 1-7. https://doi.org/10.71097/IJSAT.v16.i1.2265

VMware. (n.d.). *Kubernetes services*. VMware. https://vmware.com/topics/kubernetes-services