

คืออะไร

* Kubernetes (u-b-e-r-n-e-t-e) (เรียกสั้นๆว่า K8S ) คือ [Container Orchestration](https://www.youtube.com/watch?v=kBF6Bvth0zw&t=66s) (ตัวจัดการ container) ที่มาจัดการ application หรือ service ที่อยู่ในรูปของ container ทั้งเรื่องการ deploy, auto scaling หรืออื่น ๆ เพื่อทำให้ deployment process ง่ายขึ้น ( Orchestration  คนถือไม้โบกๆ คุมจังหวะ)
* [Introduction to Kubernetes](https://www.youtube.com/watch?v=vW1Yq5ftWZ4) (Video นี้อธิบายถึงตั้งแต่การ Deploy software ยุคก่อน จนมาถึง [Infrastructure as Code (IaC))](https://www.techtalkthai.com/introduce-infrastructure-as-code/)
* [Kubernetes ใน 60 วินาที](https://blog.nextzy.me/kubernetes-in-60-seconds-36666e1e3ef8)

ที่มาของ Kubernetes

* Kubernetes เป็น Open-source ถูกพัฒนาโดย Google โดยมีแนวคิดมาจากระบบจัดการ container ที่ใช้เองภายในบริษัท และได้นำมาเปิดตัวต่อสาธารณะเมื่อปี 2014 โดย Kubernetes มาจากคำว่า helmsman ในภาษากรีกแปลว่า “กัปตันเรือ”

**ทำไมต้องมี ?**

* เอา Docker Compose File ไปวางแล้วสั่งรันด้วยมือเราเอง ?
* แล้วเราจัดการ Deployments ของ Containers ของเราอย่างไร (ตัวอย่าง ถ้ามีการ Updates / Releases เกิดขึ้น แล้วจะจัดการอย่างไร)
* เมื่อเรา Deploy Containerized App ไปแล้ว เราจัดการอย่างไร ให้ระบบรู้ว่า Application นั้น ๆ มันไม่ทำงาน หรือ Down ลงไป แล้วจะ Restart อย่างไร แล้วไหนจะต้องทำการสร้างตัวทดแทนขึ้นมา ให้ระบบรองรับผู้ใช้งานต่อไปได้
* ถ้า Containerized App ของเรา รันๆปกติ แต่จู่ ๆ เกิดปัญหาว่า Hardware Resource ไม่พอ เราจะแก้ไขอย่างไร
* แล้วการ Scale ให้ระบบของเรามันรองรับผุ้ใช้งานได้มากขึ้นละ แล้วถ้าผู้ใช้งานน้อยลง ก็ Scale down ลงมา จะได้ประหยัดค่าใช้จ่าย

**กระบวนการทำงาน แบบสั้นๆ**

* ถ้ายังจำได้ Docker ก็คือ Software ที่จำลองสภาพแวดล้อมขึ้นมาในเครื่อง Host ที่เรียกว่า Container แล้วยัด Software เข้าไป โดยการทำงานจะไม่ไปกระทบกับ Operating system ของเครื่อง Host หรือว่า Container อื่น ๆ ส่วน Kubernetes ก็มาครอบ Docker อีกชั้นหนึ่ง โดยรวมกลุ่ม Docker Server ให้เป็นกลุ่ม (Cluster) แล้วจึงควบคุมและบริหารจัดการทรัพยากรของ Cluster (Orchestrator) ทำให้ Application ถูกบริหารจัดการได้ง่าย สามารถทำงานและได้รับทรัพยากรที่ต้องการตามที่เราได้กำหนดไว้ว่าจะทำอะไร

**Concept of Kubernetes**

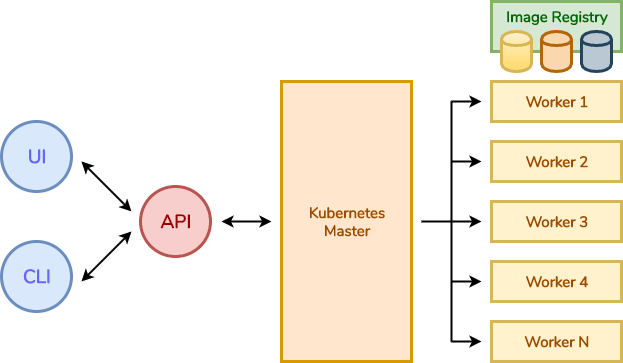
* Kubernetes เราจะแบ่ง application ออกเป็น application เล็ก ๆ ทำงานแค่หน้าที่เดียว (microservice application)
* ในการทำงานแบบ microservice application แน่นอน การเชื่อมต่อระหว่าง container เป็นสิ่งที่จำเป็น Kubernetes ทำให้มันเกิดขึ้นได้ด้วย [concept ของ service และ API calls](https://dev.to/peepeepopapapeepeepo/lfs258-4-15-kubernetes-apis-and-access-3780)
* service จะเป็นตัวเชื่อมโยง container เข้าด้วยกัน มองคล้ายๆ load balancer ที่คอยแจก traffic ไปยัง containers
* API calls คือการที่ container ไม่ต้องผูกติดกันแบบตายตัว แต่เรียกใช้งานกันผ่านการ call API ระหว่างกัน ทำให้การเพิ่มลดของ containers เป็นไปได้ไม่ยาก

**Architecture**

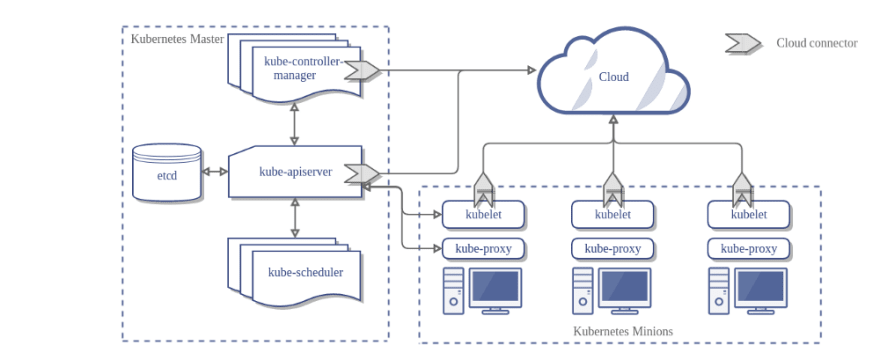
Kubernetes ประกอบด้วย Master และ Worker

Master: ทำหน้าที่เป็น control plane

Worker: เป็นแค่ workload



**high level architecture ของ kubernetes cluster**



[Kubernetes Architecture](https://dev.to/peepeepopapapeepeepo/lfs258-1-15-basics-of-kubernetes-2f8g)

[Terminology](https://dev.to/peepeepopapapeepeepo/lfs258-3-15-kubernetes-architecture-285i#services)

Master Nodes เป็นที่อยู่ของ module ที่ทำหน้าที่เป็น control plane ที่มี

* kube-apiserver  master process ของ cluster
* kube-scheduler รับ request จาก API server มาเพื่อหา node ที่เหมาะสมในการ run container
* etcd อาจ run ใน cluster หรือนอก cluster ก็ได้ เป็น Storage ในการเก็บ stage ของ containers, cluster setting และ network configuration
* kube-controller-manager เป็น core control loop daemon ที่คอยติดต่อกับ kube -apiserver เพื่อตรวจสอบ state ของ cluster

Worker Nodes เป็น server ที่ใช้ในการ run pods ที่เป็น workload ที่มี

* Kubelet คอยส่ง status ของทั้ง Pods และ resources , ดูแลเรื่องการสร้าง, เปลี่ยนแปลง และ ลบ resources
* kube-proxy เป็น process ที่จัดการเรื่อง connectivity ให้ containers
* Services เป็น components ที่ช่วยให้การ decoupling ทำได้ง่ายขึ้น โดยทำหน้าที่รวม resources เข้าด้วยกัน และ reconnect หากมี resource นั้นเพิ่มขึ้น, ลดลง หรือ restart

[Controllers](https://dev.to/peepeepopapapeepeepo/lfs258-6-15-managing-state-with-deployments-4abi) เป็นตัวที่คอยดูแล object ในความรับผิดชอบให้มี status เหมือนอย่างที่เราระบุไว้

* มี replicaSet เป็นตัวช่วยในการสร้าง pods และ scale pods ให้มีจำนวนเท่ากับที่เรากำหนด

Pods เป็นหน่วยที่เล็กที่สุด ที่ Kubernetes สามารถควบคุมได้ตรง ๆ โดย pods จะ watch loop ในการ monitor ให้ container run อยู่ตามที่กำหนด โดยปกติ จะมีแค่ 1 container ต่อ 1 pods แต่ใน 1 pods สามารถมีได้หลาย containersซึ่งมีแค่ 1 container ที่ทำ main logic ของ application ส่วน container อื่น ๆ จะทำหน้าที่เป็นส่วนเสริม

Containers เป็นส่วนที่อยู่นอกเหนือ การควบคุมของ kubernetes แต่ Kubernetes จะควบคุมมันได้อ้อมๆ ผ่านทางcontainer runtime และ resource limitation

Network  เป็นส่วนที่ให้การติดต่อสื่อสารระหว่าง pod ทั้งที่อยู่ใน node เดียวกันหรือ ต่าง node (pod-to-pod communication) เกิดขึ้นได้

Link สำหรับอ่านบทความ

[ศัพท์](https://kubernetes.io/docs/reference/glossary/?fundamental=true)

[understanding-kubernetes](https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams)

[12 Factors Application Principles](https://12factor.net/)

[เปรียบเทียบนิดหน่อยระหว่าง Openshift กับ Kubernetes](https://cloudowski.com/articles/10-differences-between-openshift-and-kubernetes/?fbclid=IwAR3k3295tVqpz8gFCy3R2OEqw187Uh0tPwMmRzpS7tLBhyyU6qAqCNKIop4)

[A Kubernetes story: Phippy goes to the zoo](https://www.youtube.com/watch?v=R9-SOzep73w)

[Kubernetes in 5 mins](https://www.youtube.com/watch?v=PH-2FfFD2PU)

[What is Kubernetes](https://www.youtube.com/watch?v=a_sQuL2mOAY)

[Container Orchestration Explained](https://www.youtube.com/watch?v=kBF6Bvth0zw&t=66s)

[Kubernetes Volumes explained](https://www.youtube.com/watch?v=0swOh5C3OVM)

[Kubernetes Explained](https://www.youtube.com/watch?v=aSrqRSk43lY)

[Kubernetes YAML File Explained](https://www.youtube.com/watch?v=qmDzcu5uY1I)

[Kubernetes Tutorial for Beginners](https://www.youtube.com/watch?v=X48VuDVv0do)

[Docker and Kubernetes Tutorial for Beginners](https://www.youtube.com/playlist?list=PLy7NrYWoggjwPggqtFsI_zMAwvG0SqYCb)

[[Kubernetes] ว่าด้วยเรื่อง Services แต่ละประเภท](https://thanwa.medium.com/kubernetes-%E0%B8%A7%E0%B9%88%E0%B8%B2%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2%E0%B9%80%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87-services-%E0%B9%81%E0%B8%95%E0%B9%88%E0%B8%A5%E0%B8%B0%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B9%80%E0%B8%A0%E0%B8%97-25bade6d4725)

[Series of K8s](https://dev.to/t/lfs258)