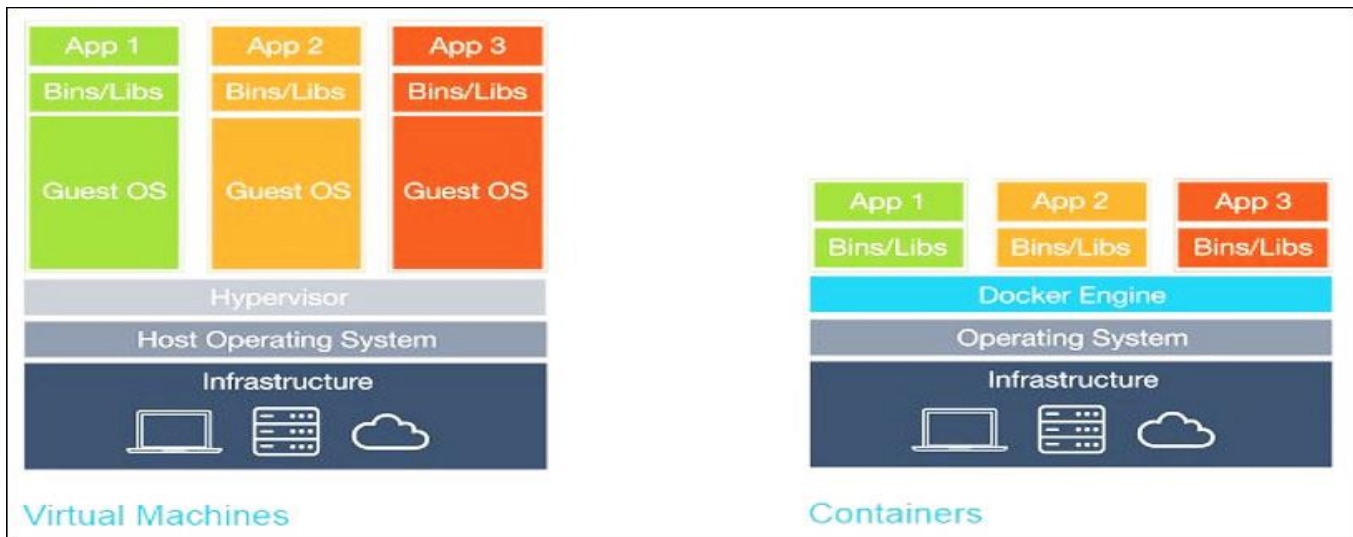


ระบบจัดการควบคุม VM กับ คอนเทนเนอร์

by TaskS



ที่มา: <http://www.its.au.edu/images/KM/Docker/p1-Different.jpg>

Virtual Machine คือ การจำลองเครื่องและสภาพแวดล้อมรวมถึงระบบปฏิบัติการ และมีการประมวลผล Service หลายๆ Service ใน VM เดียวกัน ทำให้แต่ละ VM ต้องใช้ทรัพยากรจำนวนมาก

Container คือ การจำลองและควบคุมสภาพแวดล้อมสำหรับการรันเฉพาะบาง Service เช่น Container ที่รัน nginx ใน ubuntu ก็จะสามารถ Environment เหล่านี้ไว้เป็น 1 Container และรัน Service เท่าที่จำเป็นต้องใช้เท่านั้น ทำให้ใช้ทรัพยากรน้อยกว่า Virtual Machine

แนะนำซอฟต์แวร์ Kubernetes, Docker, OpenShift, Apache Mesos, Amazon Elastic Container Service (ECS)

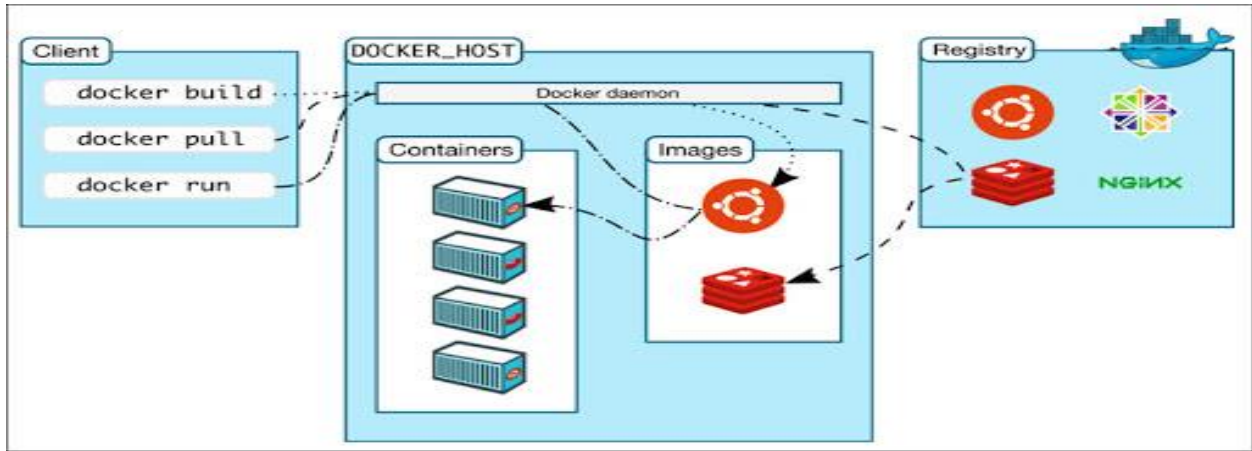
1) <https://www.portainer.io/>

2) <https://kitematic.com/>

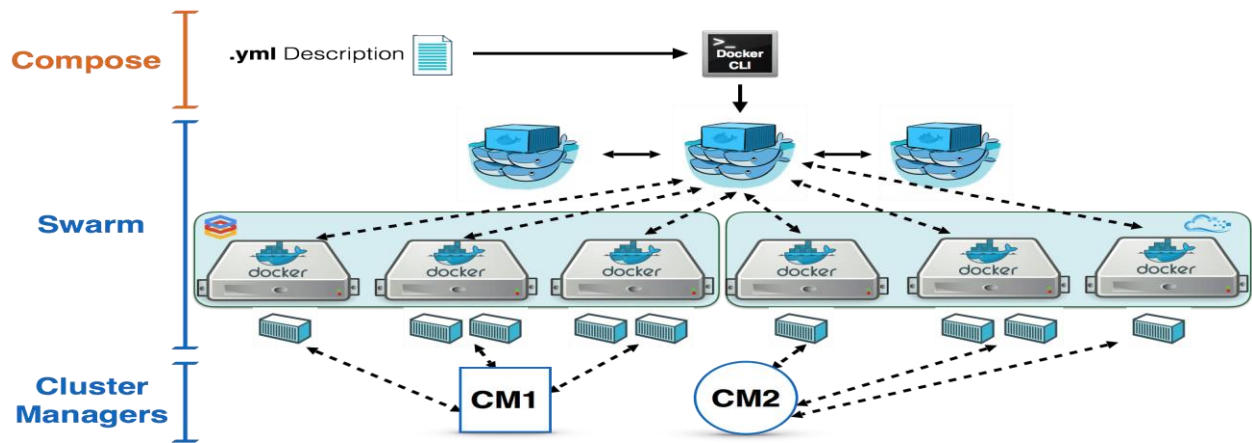
3) <https://rancher.com/>

แนวโน้มเครื่องมือในการใช้ Containers

<https://www.veritis.com/solutions/containerization-services/tools-resources-trends/>



ที่มา: <http://www.its.au.edu/images/KM/Docker/p2-docker.jpg>



ที่มา: <https://devopscube.com/wp-content/uploads/2016/09/docker-swarm.png>

Docker คือ ซอฟต์แวร์จัดการ Container ได้ง่าย Image มีขนาดเล็ก แยกเป็นชั้นๆ สร้างแนวคิด build, ship, run ในการ deploy application มีองค์ประกอบดังนี้

Docker Image คือ ต้นแบบของ Container ข้างในจะเป็น Linux ที่มีการติดตั้ง Application และมีการ Configuration เอาไว้แล้ว ซึ่งเกิดมาจากการ build ไฟล์ Docker file ขึ้นมาเป็น image

Docker Container คือ Container ที่ถูกสร้างมาจาก Docker Image ที่เป็นต้นแบบ เกิดเป็น Container จะได้ Service หรือ Application ที่สามารถเรียกใช้งานได้ทันที

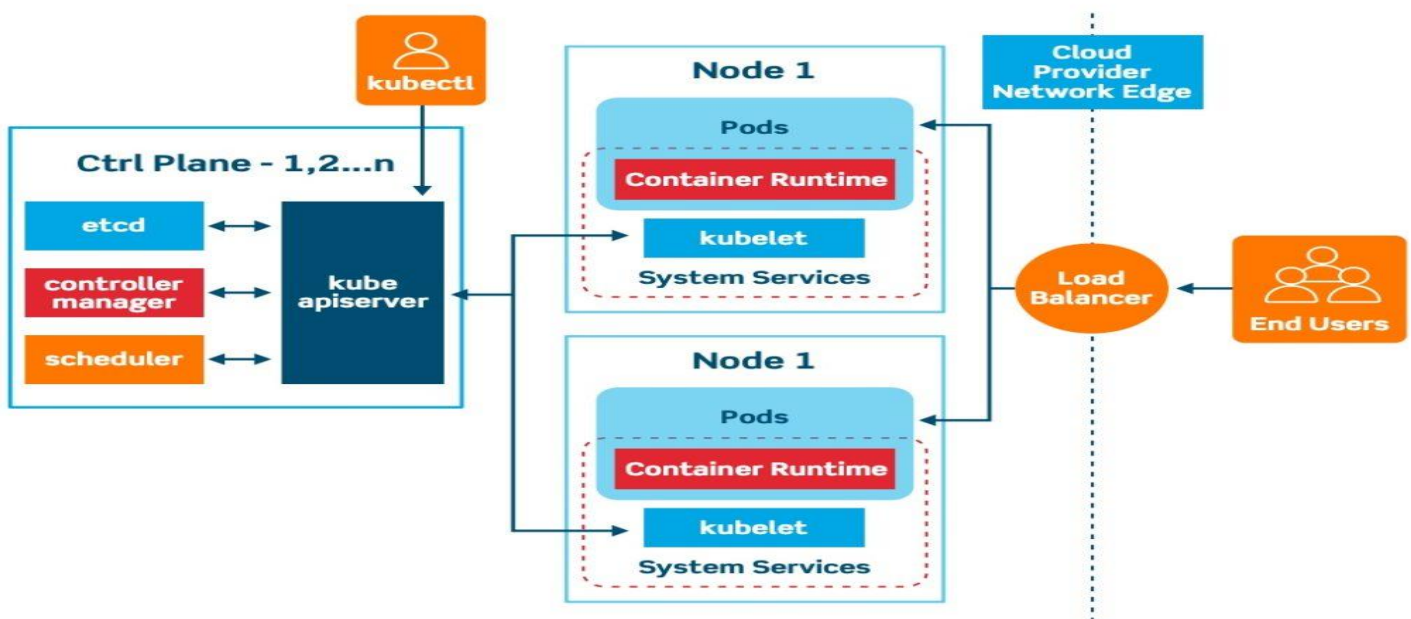
Docker Registry คือ Docker Image แล้วนำไปเก็บรวบรวมไว้บน server (ลักษณะเดียวกับการเก็บ Source Code ไว้บน Github) โดย Docker Registry ณ ปัจจุบันก็มีให้เลือกใช้งานได้หลากหลาย โดยมี Docker Hub เป็น Docker Registry หลัก ในการเรียกใช้(pull) Docker Image และนอกจากนี้ ยังมีผู้ให้บริการ Docker Registry เจ้าอื่นๆ ด้วย เช่น Gitlab, Quay.io, Google Cloud เป็นต้น

ที่มา: <http://www.its.au.edu/index.php/knowledge-management-km/its-km/178-docker.html>

Docker มีชุดซอฟต์แวร์ให้ใช้งานดังนี้

- **Docker Engine** เป็น Core หลักในการทำงาน • **Docker Machine** เป็นซอฟต์แวร์สำหรับสร้างหรือ เชื่อมต่อเครื่องให้พร้อมสำหรับ Container
- **Docker Swarm** เป็นซอฟต์แวร์ที่เอาไว้เชื่อมต่อ Docker Engine หลายๆ เครื่องมาเชื่อมต่อรวมกันให้เป็น cluster
- **Docker Compose** เป็นซอฟต์แวร์สำหรับอำนวยความสะดวกในการรัน Container โดยสามารถสั่งรันหลายๆ Container ได้พร้อมๆ กัน ตั้งค่าต่างๆ รวมไปถึงเชื่อมโยง Container ให้สามารถทำงานร่วมกันได้
- **Kitematic** เป็นเครื่องมือ GUI ที่รวบรวมคำสั่งของ Docker Command ต่าง ๆ ในรูปแบบ UI ให้สามารถใช้งานได้ง่าย ๆ ยิ่งขึ้น

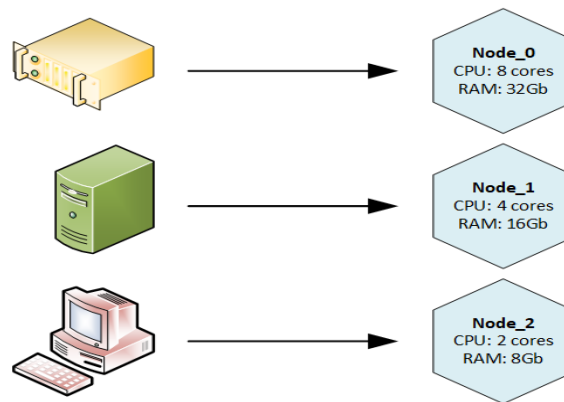
องค์ประกอบ Kubernetes



ที่มา: <https://platform9.com/wp-content/uploads/2019/05/kubernetes-constructs-concepts-architecture.jpg>

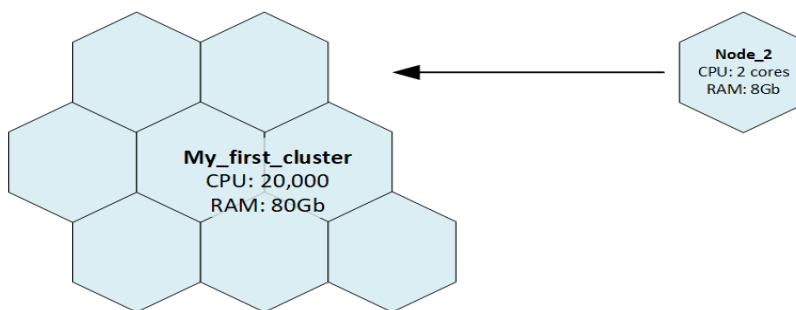
Kubernetes คือ ซอฟต์แวร์ที่เอาไว้บริหารจัดการคอนเทนเนอร์เปิดตัวปี 2558 สร้างขึ้นโดย Google ปล่อยให้ใช้ฟรีเป็นโอเพ่นซอร์สคอนเทนเนอร์ สำหรับจัดการและควบคุม โดยคอนเทนเนอร์ นำเทคโนโลยี "คอนเทนเนอร์" ในฝั่งแอปพลิเคชัน มาจับคู่กับ "คลัสเตอร์" ในฝั่งของโครงสร้างพื้นฐาน มีหน้าที่บริหารจัดการตรง คือคำว่า Container management หรือ Orchestration Container สามารถติดตั้ง Kubernetes ได้บนเครื่อง Server หรือติดตั้งบน Virtualization อย่าง VMware หรือ บนคลาวด์ เช่น AWS หรือ GCP เป็นต้น

Node(s) คือ เครื่อง Server ซึ่งแต่ละเครื่องไม่จำเป็นต้องมี Spec เหมือนกัน โดยพื้นฐานแต่ละเครื่อง จะรายงานจำนวน ซีพียู Core และขนาดของหน่วยความจำ (RAM) คุณสมบัติพิเศษบางอย่างเช่น การ์ด GPU สามารถถูกกำหนดเข้าไปอยู่ใน Spec ของ Node เพื่อใช้เลือกว่าจะรันงานเครื่องไหน



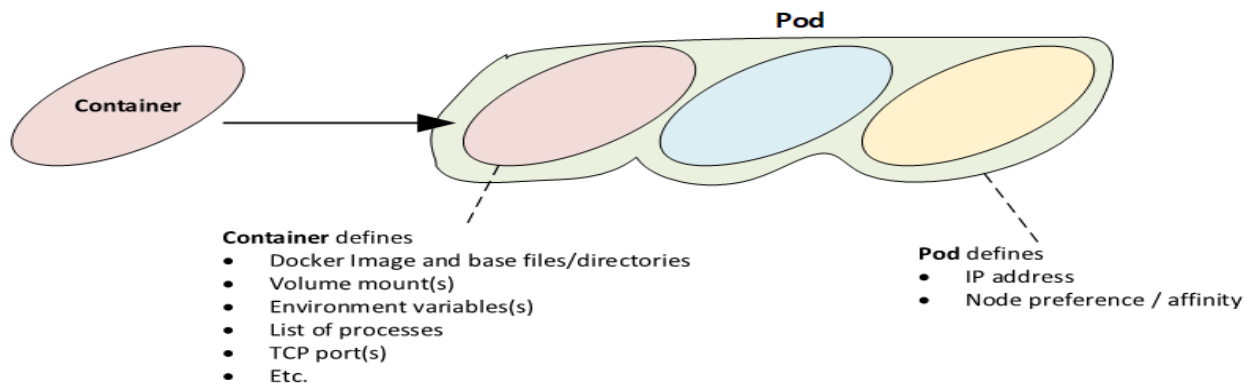
ที่มา: https://cdn-images-1.medium.com/max/900/1*pGUktdVA91ukrNL78gavXA.png

Cluster คือ เครื่อง Server ตั้งแต่หนึ่งเครื่องขึ้นไป เข้ามาทำงานร่วมกัน ตกลงกัน แสดงความเป็นเอกภาพในการทำงานร่วมกัน (Represent) รูปด้านล่างใช้หน่วย milicore เพื่อแสดงจำนวนซีพียู ซึ่งจำนวนซีพียูและหน่วยความจำของ Cluster มาจากผลรวมจากของทุกๆ เครื่องที่อยู่ใน Cluster



ที่มา: https://cdn-images-1.medium.com/max/900/1*Dzra8EPlu-QgYY4QkOMIVA.png

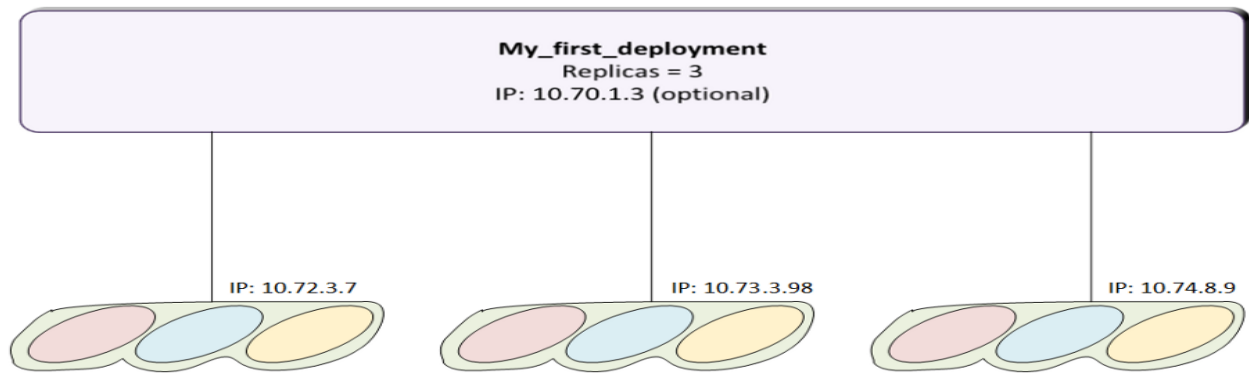
Pod คือ เป็นหน่วยการทำงานที่เล็กที่สุดของ Kubernetes ถ้าเราต้องการจะรัน Container เล็กๆ ซักอันหนึ่ง เราต้องเอา Container ของเรายัดเข้าไปใน Pod ก่อนถึงจะรันได้เราอาจจะมองว่า Container เป็นเมล็ดถั่วลิสง และ Pod เป็นเปลือกของถั่ว ในเปลือกถั่วลิสงหนึ่งเปลือกอาจจะมีเมล็ดได้หลายเมล็ด



ที่มา: https://cdn-images-1.medium.com/max/900/1*ZWSjggU4GI3CrzL1b2NwFg.png

Frontend + Business Logic + Datastore ไม่ควรอยู่ใน Pod เดียวกันเพราะว่า ในแต่ละส่วน (Layer) ควรจะขยายหรือลดจำนวนแบบตัวใครตัวมัน ถ้ามีคนเปิดหน้าเว็บเยอะๆ ก็ควรขยายแค่จำนวน Frontend อย่างเดียว ไม่จำเป็นต้องขยายส่วนอื่นให้เปลืองทรัพยากร หรือถ้าจะอัปเดตเวอร์ชัน Frontend ก็ไม่จำเป็นต้องติดตั้ง Business Logic ใหม่ (ซึ่งถ้าอยู่ใน Pod เดียวกันจะต้องติดตั้งไปพร้อมกันเสมอ)

Deployment คือ การใช้งานจริง เราจะไม่ประกาศและติดตั้ง Pod ตรงๆ (Naked Pod) เพราะถ้าเราติดตั้ง Pod ตรงๆ เมื่อ Pod นั้นมีปัญหาและหยุดการทำงาน Kubernetes จะไม่จัดการอะไรให้เรา เราจะต้องมาติดตั้ง Pod นั้นใหม่เองอีกครั้ง แต่ถ้าเราใช้ Deployment เมื่อ Pod ไหนหยุดการทำงาน Kubernetes จะติดตั้ง Pod ใหม่มาแทนที่ให้อัตโนมัติ จนกว่าจะครบตามจำนวนที่ต้องการ (Replicas)



ที่มา: https://cdn-images-1.medium.com/max/900/1*ZC4Cov-mw3c4nZC0xf7FkA.png

- Deployment เหมาะสำหรับงานที่แต่ละ Pod ทำงานเป็นอิสระจากกัน (Stateless)
- StatefulSet เหมาะกับงานที่แต่ละ Pod มีลำดับ (1 2 3) หรือบทบาทต่างกัน
- DaemonSet เหมาะกับงานที่ตั้งใจให้รันในทุกๆ Node เช่น การเก็บ log หรือ เก็บ Metrics
- ReplicationController คล้ายๆ กับ Deployment แต่ไม่แนะนำให้ใช้แล้ว
- ReplicaSet คล้ายๆ กับ Deployment ปกติแล้วจะใช้ Deployment และให้ Deployment สร้าง ReplicaSet อีกที
- Job เหมาะกับการรันงาน Batch สามารถ retry ถ้าการทำงาน fail สามารถรัน parallel ได้
- CronJob ใช้สำหรับ Schedule Job

Docker Swarm VS Kubernetes

Docker Swarm	Kubernetes
1 No Auto Scaling	1 Auto Scaling
2 Good community	2 Great active community
3 Easy to start a cluster	3 Difficult to start a cluster
4 Limited to the Docker API's capabilities	4 Can overcome constraints of Docker and Docker API
5 Does not have as much experience with production deployments at scale	5 Deployed at scale more often among organizations

ที่มา: 1) <https://phoenixnap.com/blog/wp-content/uploads/2019/10/kubernetes-vs-docker-swarm-explained-1024x544.png>

2) <https://phoenixnap.com/blog/kubernetes-vs-docker-swarm>

สรุป Docker จัดการทรัพยากรคอนเทนเนอร์ที่มีประสิทธิภาพมากกว่าเครื่องเสมือน (VM) ส่วน Kubernetes เป็นเทคโนโลยีจัดการคอนเทนเนอร์ของ Google