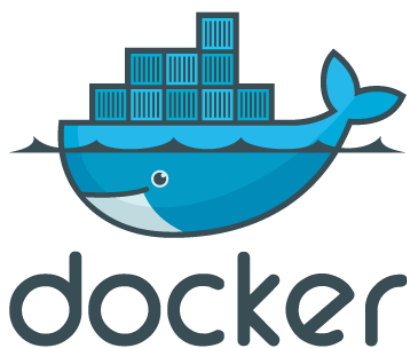


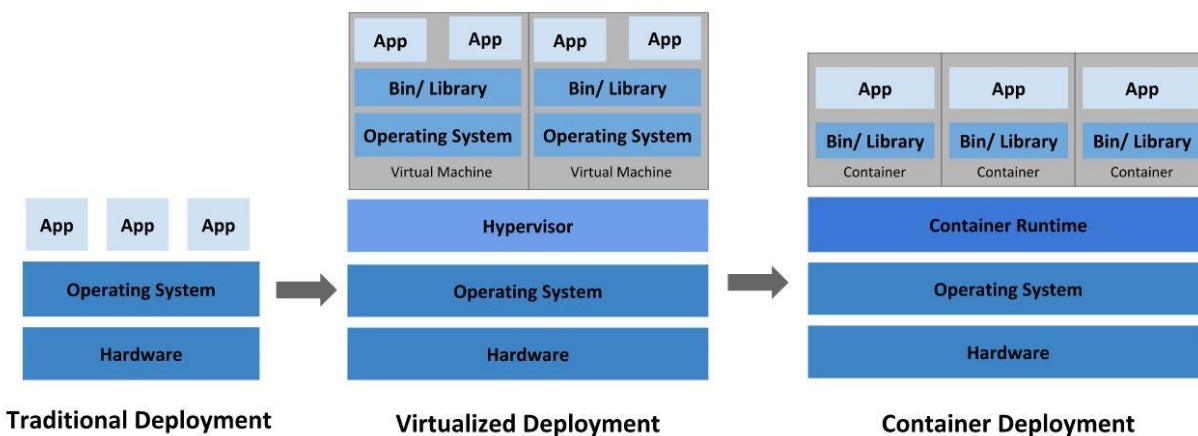
Docker คืออะไร?



Docker คือเครื่องมือแบบ open-source หรือแพลตฟอร์มซอฟต์แวร์ ที่ช่วยให้คุณสร้าง ทดสอบ และติดตั้งแอปพลิเคชันใช้จริงได้อย่างรวดเร็ว Docker จะบรรจุซอฟต์แวร์ลงไปเป็นหน่วยที่เป็นมาตรฐาน เรียกว่า คอนเทนเนอร์ (Container) ซึ่งจะมีทุกสิ่งที่ซอฟต์แวร์ต้องใช้ในการเรียกใช้งาน รวมทั้งไลบรารี เครื่องมือสำหรับระบบ โค้ด และรันไทม์ เมื่อใช้ Docker คุณสามารถติดตั้งใช้จริงและปรับขนาดแอปพลิเคชันให้เหมาะกับทุกสภาพแวดล้อม (Deploy) และทราบว่าโค้ด

ของคุณจะเรียกใช้ได้อย่างอย่างรวดเร็ว Docker จึงเป็นที่รู้จักในวงกว้างและเริ่มเข้ามามีบทบาทอย่างมากในโลกของการพัฒนา Software สามารถรองรับการติดตั้งใช้งานบนระบบปฏิบัติการที่หลากหลายไม่ว่าจะเป็น Linux, Windows, MAC

วิวัฒนาการของการ Deploy

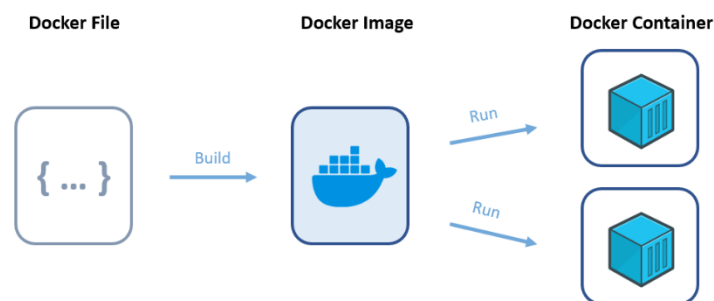


อ้างอิงข้อมูลจาก <https://blog.skooldio.com/what-is-docker/#Docker>

1. Traditional Deployment ในยุคที่เราใช้ Physical server 1 เครื่อง ในการ deploy และเพื่อความคุ้มค่า Physical server ที่เรามีจะถูกใช้ในการลง application หลายๆ อันพร้อมๆ กัน ซึ่งทำให้เกิดปัญหา element ของแต่ละ application ตีกัน เช่น application แต่ละตัวมีการลง JAVA ซึ่งเป็นในเครื่องเรามี JAVA หลายๆ เวอร์ชัน 1.5, 1.6, 1.8 ทำให้เวลา run มีปัญหาเกิดขึ้น ทั้งการ maintenance และปัญหาในการเลือกเวอร์ชัน

2. Virtualized Deployment เป็นยุคที่มีการเกิดขึ้นของ software hypervisor ซึ่ง concept คือการจำลองเครื่อง Physical server ขึ้นมา เรียกว่า Virtual Machine ทั้ง CPU, memory, hard disk, hardware ต่างๆ ขึ้นมาเสมือนคอมพิวเตอร์เลย ปรับสเปคปรับความเร็วต่างๆ ได้ตามงบประมาณที่เรา มี ซึ่งส่วนใหญ่ก็จะสร้าง VM ขึ้นมาหลายๆ เครื่อง ให้แต่ละเครื่องเพียงพอต่อการ run application แต่ละตัว แต่ปัญหาคือจะเกิดปัญหา Overhead ทั้งในเรื่องการจำลอง hardware ต่างๆ ทำให้ทำงานได้ช้าลง 2-5 เท่า รวมถึง Overhead ในกรณีที่เรามีการ run ใน environment ที่ใกล้เคียงกัน จะทำให้เปลือง resource ไปโดยใช่เหตุ
3. Container Deployment ซึ่งในยุคนี้จะพูดถึงการสร้าง Container ขึ้นมาเพื่อชิง resource สร้างกำแพงขึ้นมาแบ่ง resource ทำให้เราสามารถใช resource ได้อย่างมีประสิทธิภาพมากขึ้น เลือก container ไปใช้กับแต่ละ application ได้ดียิ่งขึ้น ทำให้ปัญหา overhead ลดลง และเครื่องมือในการสร้าง Container ที่นิยมก็คือ Docker

องค์ประกอบพื้นฐาน Docker



อ้างอิงข้อมูลจาก https://race.nstru.ac.th/home_ex/blog/user/data/image/4200/24092021-121556.png

- Docker File คือ เอกสารบอกโค้ดคำสั่ง สำหรับสร้าง Docker Image นั้นๆ
- Docker Image เป็นแม่แบบที่ใช้ในการสร้างเป็น Docker Container ซึ่งประกอบไปด้วยแอปพลิเคชันต่างๆ ที่จะทำงานเมื่อมีการเรียกใช้งานจาก Docker Container นั้นๆ รวมทั้งการตั้งค่าจำลองสภาพแวดล้อม (Environment) ที่จำเป็นสำหรับการทำงานของมันไว้ด้วย
- Docker Container เป็นที่บรรจุรวมของแอปพลิเคชัน สภาพแวดล้อมที่จำเป็นต่อการทำงาน และองค์ประกอบต่างๆ ที่จำเป็นต่อการทำงานของมัน ซึ่งสามารถสร้างจาก Docker Image ผ่านการกำหนดโครงสร้างของมันที่ Docker File

ความน่าสนใจของ docker

- Docker engine สามารถใช้งานได้บนหลาย platform ทั้งบน Linux, Mac และ Windows
- Docker มีขนาดเล็ก สามารถใช้งาน และติดตั้งได้อย่างรวดเร็ว และสะดวกในการ start / stop หรือ แม้แต่การย้ายไปใช้งานสำหรับเครื่อง server อื่นที่มีการ run docker engine ก็ยังสามารถทำได้โดยไม่ซับซ้อน
- ผู้ใช้งาน docker ไม่จำเป็นต้องติดตั้ง OS อีกครั้งเพื่อติดตั้ง container รวมทั้งไม่จำเป็นต้อง config เพิ่มเติมในส่วนที่ไม่จำเป็นอีกด้วย
- Docker มีความต้องการในการใช้ CPU, RAM และพื้นที่น้อยกว่า Virtual Machine ทั้งนี้ในทรัพยากรที่มีเท่ากัน docker สามารถใช้งาน container ได้มากกว่า Virtual Machine
- เนื่องจากผู้ใช้งาน สามารถสร้าง docker image ได้เอง จาก Docker file ดังนั้นการใช้งาน docker ยังช่วยลดปัญหาสภาพแวดล้อมที่ต่างกัน ที่มักพบเมื่อบาง application สามารถทำงานได้บน development server แต่ไม่สามารถใช้งานบน production server ได้
- Docker ยังมี docker registry ซึ่งผู้ใช้งานสามารถเลือก pull image ต่างๆ ที่มีการสร้างไว้ให้แล้วมาใช้งาน โดยมี Docker Hub เป็น registry หลักในการเรียกใช้ image

ข้อดีของ Docker เมื่อเทียบกับ Virtual Machine

- Docker ไม่ต้องเสียเวลาในสร้าง OS ใหม่ และการ config แต่ละ OS เลย
- Docker เบาและเร็วกว่ามาก ไม่ว่าจะเป็น start stop และ restart เพราะมันใช้ OS, CPU และ RAM ร่วมกันกับ Host OS
- Docker สามารถรัน container ได้มากกว่า VMs ในเครื่องที่มีทรัพยากรที่เท่ากัน
- Docker มีระบบ Registry ทำให้สามารถเคลื่อนย้าย หรือติดตั้ง Container ได้สะดวก และรวดเร็วกว่ามาก
- Containers มันรันอยู่บน Docker Engine ทำให้ไม่ต้องสนใจว่า Infrastructure หรือ Host OS ว่าจะเป็นอะไรยังไง ทำให้หมดปัญหาว່ว่าเครื่อง Dev รันได้ แต่เครื่อง Production มันรันไม่ได้บ้าง หรือเครื่อง Dev แต่ละคนติดตั้งเครื่องมือคนละเวอร์ชันกัน เราก็ build container เป็น image แล้วส่งในคนในทีมใช้ ก็หมดปัญหาแล้ว

Reference

1. <https://mthai.win/index.php/2021-08-04-09-30-18/89-virtual-machines>
2. <https://blog.skooldio.com/what-is-docker/#Docker>
3. <https://medium.com/@teamteam/@teamteam/ทำความเข้าใจกับ-docker-software-container-c6338629da11>
4. https://race.nstru.ac.th/home_ex/blog/topic/show/4200
5. <https://www.hostpacific.com/using-docker-on-centos7/>
6. <https://km.cc.swu.ac.th/archives/3620>
7. <https://medium.com/@rachatatongpagdee/docker-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3-7e77145967b6>