

MAKALAH
“TOOLS DEVOPS - DOCKER”



Disusun Oleh :

5200411332 - Dwi Muhammad Faza

5200411409 - Pramudea Yohano Firmansyah

5200411416 - Muhammad Ilham Triwibowo

5200411417 - Rusli Pramono

PROGRAM STUDI METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

YOGYAKARTA

2021

DAFTAR ISI

Bab I	Pendahuluan	1
1.1	Latar Belakang	1
1.2	Rumusan Masalah	2
1.3	Tujuan.....	2
Bab II	Pembahasan.....	3
2.1	Definisi Docker.	3
2.2	Cara kerja Docker.....	3
2.3	Fungsi Docker.	3
2.4	Arsitektur pada Docker	4
2.5	Keuntungan menggunakan Docker	5
Bab III	Penggunaan Tools DevOps	6
Bab IV	Penutup.....	8
Referensi	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan komputer dan internet saat ini aplikasi berbasis web berkembang sangat pesat. Selain itu, aplikasi berbasis web juga semakin banyak digunakan karena dapat diakses di berbagai platform komputer hanya dengan menjalankan web browser. Sehingga, kemudahan proses deployment (penyebaran) aplikasi web beserta software pendukung seperti web server, database server, dependency dan environment lain ke server sangat dibutuhkan.

Secara umum ada dua metode deployment aplikasi web ke dalam server. Pertama menginstall aplikasi web beserta environment yang dibutuhkan ke dalam server tunggal, kelebihanannya adalah melakukan konfigurasi terhadap server lebih mudah, sederhana dan cepat dalam proses deployment. Tetapi metode tersebut memiliki kekurangan yaitu setiap aplikasi tidak terisolasi, sehingga apabila mendeploy beberapa aplikasi yang masing-masing memiliki ketergantungan dengan paket versi tertentu maka dapat menimbulkan Dependencie Hell.

Metode yang kedua yaitu dengan memanfaatkan teknologi virtualisasi berbasis hypervisor, jadi setiap aplikasi dan dependency yang dibutuhkan di deploy ke dalam mesin virtual yang berbeda. Dengan metode ini dapat meningkatkan skalabilitas, karena setiap aplikasi berjalan pada sumber daya (CPU, memori, penyimpanan data) yang berbeda sehingga dapat dengan mudah ditambahkan sesuai kebutuhan.

Docker sangat ringan dan mempunyai mekanisme yang lebih maju jika dibandingkan dengan perangkat lunak virtualisasi berbasis hypervisor. Indikasinya adalah adanya efektivitas lebih pada Docker dalam hal penggunaan sumber daya mesin host. Karena dalam proses deployment, docker akan menjalankan sebuah container menggunakan base image dengan metode file system as a layer yang berarti Docker hanya akan menyalin lapisan perubahannya saja untuk dijalankan sebagai duplikasi container yang berbeda dengan base image yang sama.

1.2 Rumusan Masalah

Berdasarkan pemaparan di atas maka dapat disimpulkan beberapa rumusan masalah sebagai berikut:

1. Bagaimana langkah melakukan virtualisasi docker container untuk menjamin web menjadi stabil pada linux ubuntu 18.04 LTS.
2. Bagaimana cara membuat sebuah server menjadi lebih efektif dengan hardware yang minimum.

1.3 Tujuan

Tujuan yang ingin dicapai dari perancangan tugas ini adalah sebagai berikut:

1. Mengimplementasikan sistem virtualisasi server berbasis docker container.
2. Melakukan pengujian terhadap efektivitas docker dalam penggunaan sumber daya CPU dan memori
3. Meningkatkan efektivitas penggunaan sumber daya hardware mesin server. Penelitian ini dapat memberikan manfaat sebagai.

BAB II

PEMBAHASAN

2.1 Definisi Docker.

Docker adalah platform perangkat lunak yang memungkinkan Anda membuat, menguji, dan menerapkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam unit standar yang disebut kontainer yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi termasuk pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan Docker, Anda dapat dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode Anda akan berjalan.

Dalam pengembangan aplikasi, developer memerlukan virtualisasi di server agar aplikasi bisa berjalan di berbagai platform dengan konfigurasi hardware yang berbeda-beda. Sayangnya, ketika menggunakan virtualisasi, Development harus menyiapkan satu sistem operasi secara penuh. Jika membutuhkan beberapa virtualisasi, server perlu resource yang besar. Oleh karena itu, container bisa digunakan sebagai alternatif virtualisasi sehingga tidak perlu menyiapkan sistem operasi secara penuh. Dengan container, ukuran file menjadi lebih kecil dibandingkan virtualisasi yang biasa digunakan.

2.2 Cara kerja Docker.

Docker berfungsi dengan menyediakan cara standar untuk menjalankan kode Anda. Docker adalah sistem operasi untuk kontainer. Mirip dengan cara mesin virtual memvirtualisasi (menghilangkan kebutuhan untuk secara langsung mengelola) perangkat keras server, kontainer memvirtualisasi sistem operasi server. Docker diinstal di setiap server dan memberikan perintah sederhana yang dapat Anda gunakan untuk membuat, memulai, atau menghentikan kontainer.

2.3 Fungsi Docker.

1. Mempermudah Pengembangan Aplikasi.

Docker bisa mempermudah pekerjaan developer ketika mengembangkan aplikasi. Alasannya, Docker lebih hemat resource dan mampu menyediakan environment yang stabil untuk dijalankan di perangkat apapun, mulai dari cloud server hingga komputer pribadi.

2. Menyederhanakan Konfigurasi.

Docker tidak memiliki overhead sehingga developer bisa menjalankan aplikasi yang diuji tanpa konfigurasi tambahan.

3. Memudahkan Pengembangan Kode Pipeline.

Developer bisa memanfaatkan Docker container sebagai tempat pengujian kode Pipeline beserta tools yang diperlukan dengan lebih mudah.

4. Bisa Digunakan untuk Debugging.

Adanya fitur debug bisa membantu developer untuk mengatasi masalah pada aplikasi tanpa perlu bersusah payah meninggalkan environment di Docker.

5. Mendukung Multitenancy.

Docker cocok digunakan untuk membuat aplikasi berstruktur multitenancy seperti Software as a Service (SaaS). Anda bisa membuat lebih dari satu environment yang terisolasi dan menjalankan objek aplikasi untuk setiap tenant.

6. Meningkatkan Sumber Daya dengan Cepat.

Dengan Docker, peningkatan sumber daya perangkat dapat dilakukan dengan cepat sehingga durasi pengembangan software lebih singkat.

2.4 Arsitektur pada Docker

1. Docker Daemon adalah sebuah service yang dijalankan di dalam host dalam Operating System (OS). Fungsinya adalah membangun, mendistribusikan, dan menjalankan container docker. Pengguna tidak dapat langsung menggunakan docker daemon, akan tetapi untuk menggunakan docker daemon maka pengguna.
2. Docker Client adalah seperangkat perintah command line untuk mengoperasikan docker container, misalnya membuat container, start/stop container, menghapus (destroy), dan sebagainya yang ada di docker daemon.
3. Docker Images adalah sebuah komponen docker yang berupa template dasar yang bersifat read only. Template ini sebenarnya adalah sebuah OS atau OS yang telah diinstall berbagai aplikasi. Docker images berfungsi untuk membuat docker container, dengan hanya satu docker image dapat membuat banyak docker container.

4. Docker Container merupakan sebuah images yang dapat dikemas dan bersifat read-write, container berjalan diatas images. Docker container juga bisa dikatakan sebagai sebuah folder, dimana docker container ini dibuat dengan menggunakan docker container. Setiap docker container disimpan maka akan terbentuk layer baru tepat diatas docker images atau base images diatasnya.
5. Docker Registry merupakan repository distribusi kumpulan docker images yang bersifat private maupun public yang dapat di akses melalui docker hub.

2.5 Keuntungan menggunakan Docker

1. Kirim lebih banyak perangkat lunak dengan lebih cepat

Pengguna Docker rata-rata mengirimkan perangkat lunak 7x lebih sering daripada pengguna non-Docker. Docker memungkinkan Anda mengirim layanan terisolasi sesering yang diperlukan.

2. Menstandarkan operasi

Aplikasi dalam kemasan kecil memudahkan penerapannya, mengidentifikasi masalah, dan memutar kembali untuk remediasi.

3. Memindahkan dengan mulus

Aplikasi berbasis Docker dapat dipindahkan dari mesin pengembangan lokal ke penyebaran produksi di AWS.


4. Menghemat uang

Kontainer Docker memudahkan untuk menjalankan lebih banyak kode pada setiap server, meningkatkan pemanfaatan Anda dan menghemat uang Anda.

BAB III

PENGUNAAN TOOLS DEVOPS

Judul Artikel	IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER
Penulis	Saleh Dwiyatno, Edy Rakhmat, Oki Gustiawan
Nama Jurnal	PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer Vol. 7 No. 2 (2020)
Media Publikasi	PDF / e-jurnal.lppmunsera.org
Link Artikel	https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/2520
Tahun, halaman	Sep 1 2020, Halaman 165 - 175
Tujuan Penelitian	<p>Penelitian ini dilakukan bertujuan untuk:</p> <ol style="list-style-type: none"> 1.Mengimplementasikan sistem virtualisasi server berbasis docker container. 2.Melakukan pengujian terhadap efektifitas docker dalam penggunaan sumber daya CPU dan memori. 3.Meningkatkan efektifitas penggunaan sumber daya hardwaremesin server.
Metode Penelitian	Tipe penelitian yang dipilih yaitu penelitian terapan, yang bertujuan untuk memberikan solusi serta pengembangandalam infrastruktur server yang ada di ICT SMK Negeri 1 Rangkasbitung.

Alur Kerja	 <pre> graph TD A[Studi Literatur] --> B[Identifikasi Masalah] B --> C[Perancangan Sistem] C --> D[Implementasi] D --> E[Pengujian Sistem] E --> F[Analisis Hasil Pengujian] F --> G[Kesimpulan dan Saran] </pre>
Kesimpulan	<p>Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan, yaitu:</p> <ol style="list-style-type: none"> 1. Dengan adanya penerapan virtualisasi server berbasis docker container pada ubuntu 18.04 LTS, dapat menjamin web menjadi stabil. Karena pada docker container memastikan aplikasi dan sumber daya yang terisolasi serta terpisah sehingga para penggunanya dapat menyesuaikan kebutuhan di setiap aplikasi tanpa perlu mempengaruhi konfigurasi pada aplikasi yang lain. 2. Adanya pemanfaatan docker container pada perancangan server dapat memanfaatkan hardware yang ada untuk digunakan secara maksimal. Karena pada docker container dimana kernel yang digunakan adalah dari bagian sistem operasi host nya sendiri, sehingga tidak membebani kinerja dari server host

BAB IV

PENUTUP

Docker adalah platform perangkat lunak yang memungkinkan untuk membuat, menguji, dan menerapkan aplikasi dengan cepat. Dengan menggunakan Docker, mampu melakukan dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode akan berjalan.

Fungsi Docker antara lain :

1. Mempermudah pengembang aplikasi
2. Menyederhanakan konfigurasi
3. Memudahkan pengembangan kode pipeline
4. Bisa digunakan untuk debugging
5. Mendukung multitenancy
6. Meningkatkan sumber daya dengan cepat

Dengan menggunakan Docker tentunya memiliki keunggulan tersendiri yaitu mengirim lebih banyak perangkat lunak dengan cepat, menstandarkan operasi, memindahkan dengan mulus, dan menghemat uang.

Perbandingan dengan metode pengembangan perangkat lunak lain contohnya Waterfall lebih cocok digunakan untuk sistem yang bersifat generic, Prototype lebih cocok untuk sistem yang bersifat customize, RAD cocok untuk sistem yang bersifat customize, berskala besar, dan memerlukan waktu yang singkat. Metode Agile yang cocok untuk sistem jangka pendek dan fleksibel, serta metode DevOps yang lebih cocok untuk sistem microservice website.

Pada project-project yang sebelumnya sudah kami coba, kami lebih memilih menggunakan sistem RAD karena project RAD dapat berkembang sesuai keinginan pengguna yang dimana terbentuknya komunikasi yang bagus antara developer dan pengguna aplikasi/website tersebut. Kami tidak menggunakan DevOps dengan Docker karena skala dari project yang kami bangun sebelumnya termasuk ke dalam skala project kecil, sedangkan Docker sendiri digunakan untuk project yang memiliki skala yang sedang/besar seperti microservice. Bisa saja kami menggunakan

Docker pada project kami, namun itu dirasa tidak perlu karena penggunaan Docker tidak terlalu berefek banyak dan malah memperlambah waktu development. Docker sendiri dapat sangat berguna jika digunakan pada project skala sedang hingga besar.

REFERENSI

Dwiyatno, S., Rakhmat, E., & Gustiawan, O. (2020). IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer Vol. 7 No. 2*, 165 -175.