TEKNOLOGI CLOUD COMPUTING



DISUSUN OLEH :

Nama : Lusi Padmawati

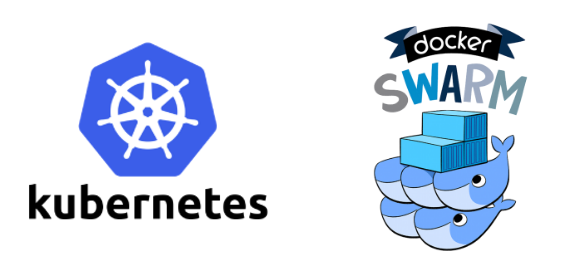
NIM : 175610045

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER

AKAKOM YOGYAKARTA

2019

**KOMPARASI KUBERNETES VS DOCKER SWARM**



**Kubernetes**

Kubernetes adalah platform open-source yang dibuat oleh Google untuk operasi penyebaran kontainer, naik turun, dan otomatisasi melintasi kelompok host. Platform siap produksi, tingkat perusahaan, penyembuhan sendiri (penskalaan otomatis, replikasi otomatis, restart otomatis, penempatan otomatis) ini bersifat modular, sehingga dapat digunakan untuk penyebaran arsitektur apa pun.

Kubernetes juga mendistribusikan beban di antara kontainer. Ini bertujuan untuk meringankan alat dan komponen dari masalah yang dihadapi karena menjalankan aplikasi dalam cloud pribadi dan publik dengan menempatkan wadah ke dalam kelompok dan menamakannya sebagai unit logis. Kekuatan mereka terletak pada penskalaan yang mudah, portabilitas agnostik lingkungan, dan pertumbuhan yang fleksibel.

**Docker Swarm**

Sebagai sebuah platform, Docker telah merevolusi cara perangkat lunak itu dikemas. Docker Swarm atau hanya Swarm adalah platform orkestrasi wadah sumber terbuka dan merupakan mesin pengelompokan asli untuk dan oleh Docker. Perangkat lunak, layanan, atau alat apa pun yang berjalan dengan wadah Docker berjalan dengan baik di Swarm. Juga, Swarm menggunakan baris perintah yang sama dari Docker.

Swarm mengubah kumpulan host Docker menjadi virtual, host tunggal. Swarm sangat berguna bagi orang-orang yang mencoba untuk merasa nyaman dengan lingkungan yang diatur atau yang perlu mematuhi teknik penyebaran sederhana tetapi juga memiliki lebih dari satu lingkungan cloud atau satu platform tertentu untuk menjalankan ini.

**Kubernetes Versus Docker Swarm**

Meskipun kedua platform orkestra open-source menyediakan banyak fungsi yang sama, ada beberapa perbedaan mendasar antara bagaimana keduanya beroperasi. Di bawah ini adalah beberapa poin penting. Bagian ini membandingkan fitur Docker Swarm dan Kubernetes dan kelemahan / kekuatan memilih satu platform di atas yang lain.

1. **Definisi aplikasi**

Kubernetes: Suatu aplikasi dapat digunakan di Kubernetes menggunakan kombinasi layanan (atau layanan microser), penyebaran, dan pod.

Docker Swarm: Aplikasi dapat digunakan sebagai layanan mikro atau layanan di cluster swarm di Docker Swarm. File YAML (YAML Ain Markt Language) dapat digunakan untuk mengidentifikasi multi-kontainer. Selain itu, penulisan Docker dapat menginstal aplikasi.

1. **Jaringan**

Kubernetes: Model jaringan adalah jaringan datar, memungkinkan semua pod untuk saling berinteraksi. Kebijakan jaringan menentukan bagaimana pod berinteraksi satu sama lain. Jaringan datar diimplementasikan biasanya sebagai overlay. Model ini membutuhkan dua CIDR: satu untuk layanan dan yang lainnya dari mana pod mendapatkan alamat IP.

Docker Swarm: Node yang bergabung dengan cluster swarm menghasilkan jaringan overlay untuk layanan yang menjangkau setiap host di docker swarm dan jaringan docker host-only untuk kontainer. Para pengguna memiliki pilihan untuk mengenkripsi lalu lintas data kontainer sambil membuat jaringan overlay sendiri di gerombolan buruh pelabuhan.

1. **Skalabilitas**

Kubernetes: Untuk sistem terdistribusi, Kubernetes lebih merupakan kerangka kerja semua-dalam-satu. Ini adalah sistem yang kompleks karena memberikan jaminan kuat tentang status kluster dan satu set API yang disatukan. Ini memperlambat penskalaan dan penyebaran kontainer.

Docker Swarm: Docker Swarm, bila dibandingkan dengan Kubernetes, dapat menggunakan kontainer lebih cepat dan ini memungkinkan waktu reaksi yang lebih cepat untuk mengurangi permintaan.

1. **Ketersediaan Tinggi**

Kubernetes: Semua pod di kubernet didistribusikan di antara node dan ini menawarkan ketersediaan tinggi dengan mentolerir kegagalan aplikasi. Memuat layanan penyeimbangan di kubernet mendeteksi polong yang tidak sehat dan membuangnya. Jadi, ini mendukung ketersediaan tinggi.

Docker Swarm: Karena layanan dapat direplikasi dalam node Swarm, Docker Swarm juga menawarkan ketersediaan tinggi. Node manajer Swarm di Docker Swarm bertanggung jawab atas seluruh cluster dan menangani sumber daya node pekerja.

1. **Pengaturan Kontainer**

Kubernet: Kubernet menggunakan definisi YAML, API, dan klien masing-masing dan masing-masing berbeda dari yang setara dengan standar buruh pelabuhan. Dengan kata lain, Anda tidak dapat menggunakan Docker Compose atau Docker CLI untuk mendefinisikan kontainer. Saat berpindah platform, definisi dan perintah YAML perlu ditulis ulang.

Docker Swarm: Docker Swarm API tidak sepenuhnya mencakup semua perintah Docker tetapi menawarkan banyak fungsi yang akrab dari Docker. Ini mendukung sebagian besar alat yang berjalan dengan Docker. Namun demikian, jika Docker API kekurangan operasi tertentu, tidak ada cara mudah untuk memanfaatkannya.

1. **Penyeimbang beban**

Kubernet: Pod dibuka melalui layanan, yang dapat digunakan sebagai penyeimbang beban di dalam kluster. Secara umum, ingress digunakan untuk menyeimbangkan beban.

Docker Swarm: Mode Swarm terdiri dari elemen DNS yang dapat digunakan untuk mendistribusikan permintaan yang masuk ke nama layanan. Layanan dapat ditetapkan secara otomatis atau dapat berjalan pada port yang ditentukan oleh pengguna.

1. **Penyimpanan**

Kubernetes Dua API penyimpanan: Yang pertama menyediakan abstraksi untuk backend penyimpanan individual (mis. NFS, AWS EBS, ceph, flocker). Yang kedua menyediakan abstraksi untuk permintaan sumber daya penyimpanan (mis. 8 Gb), yang dapat dipenuhi dengan berbagai backend penyimpanan. Memodifikasi sumber daya penyimpanan yang digunakan oleh daemon Docker pada node cluster memerlukan untuk sementara menghapus node dari cluster. Kubernetes menawarkan beberapa jenis volume persisten dengan dukungan blok atau file. Contohnya termasuk iSCSI, NFS, FC, Layanan Web Amazon, Google Cloud Platform, dan Microsoft Azure. Volume emptyDir tidak persisten dan dapat digunakan untuk membaca dan menulis file dengan sebuah wadah.

Docker Engine dan Swarm mendukung pemasangan volume ke wadah.Filesystem bersama, termasuk NFS, iSCSI, dan saluran serat, dapat dikonfigurasi node. Opsi plugin termasuk Azure, Google Cloud Platform, NetApp, Dell EMC, dan lainnya.

1. **Performa dan skalabilitas**

Kubernetes : Dengan rilis 1,6, Kubernetes menskalakan hingga 5.000 simpul node. Skalabilitas Kubernetes dibandingkan dengan Service Level Objectives (SLOs) berikut: Responsif API: 99% dari semua panggilan API kembali dalam waktu kurang dari 1 detik. Waktu mulai pod: 99% dari pod dan wadahnya (dengan gambar yang sudah ditarik) mulai dalam 5s.

Docker : Menurut posting blog Docker tentang penskalaan kelompok Swarm, Docker Swarm telah diskalakan dan kinerjanya diuji hingga 30.000 kontainer dan 1.000 node dengan 1 manajer Swarm.

1. **Meningkatkan Aplikasi dan Rollback**

Kubernetes : Pengontrol penyebaran mendukung strategi "pembaruan-bergulir" dan "menciptakan kembali". Pembaruan bergulir dapat menentukan jumlah maksimum pod yang tidak tersedia atau jumlah maksimum yang berjalan selama proses.

Docker : Pada saat peluncuran, menerapkan pembaruan bergulir ke layanan. Manajer Swarm memungkinkan Anda mengontrol penundaan antara penyebaran layanan ke berbagai set simpul, sehingga memperbarui hanya 1 tugas pada satu waktu.

**Kesimpulan**

Kubernetes mendukung permintaan yang lebih tinggi dengan lebih banyak kompleksitas sementara Docker Swarm menawarkan solusi sederhana yang cepat untuk memulai. Docker Swarm telah cukup populer di kalangan pengembang yang lebih suka penyebaran cepat dan kesederhanaan. Secara bersamaan, Kubernet digunakan dalam lingkungan produksi oleh berbagai perusahaan internet profil tinggi yang menjalankan layanan populer. Baik Kubernetes dan Docker Swarm dapat menjalankan banyak layanan yang sama tetapi mungkin memerlukan pendekatan yang sedikit berbeda untuk detail tertentu.