



Kubernetes

Apa Itu Kubernetes ?

Kubernetes adalah salah satu produk open source untuk sistem manajemen container bersifat open source. Aplikasi management open source ini adalah senjata yang diandalkan oleh Google dengan tujuan mendongkrak naiknya peringkat Google di pasar cloud hosting.

Pengertian lainnya dari aplikasi ini yaitu jenis platform open source dan sering digunakan pada manajemen container. Aplikasi ini dapat disingkat dengan huruf dan angka K8s.

Mengapa Kubernetes Di Butuhkan?

Dengan hadirnya teknologi baru yaitu **container**, maka seluruh aplikasi yang dimiliki akan di kemas sehingga ketika akan di jalankan dimanapun, aplikasi tersebut akan berjalan sama seperti pada saat mencobanya. Dengan adanya container maka tidak perlu lagi menginstall server dengan berbagai macam aplikasi, environment, dan kebutuhan lainnya, karena semua yang dibutuhkan oleh aplikasi tersebut sudah terpasang di container.

Untuk menjalankan container, tentunya membutuhkan sebuah aplikasi. Aplikasi yang paling umum digunakan adalah Docker. Docker di install di server, kemudian menjalankan *docker image* dan terbentuklah *container* dan aplikasi sudah berjalan dan bisa digunakan.

Tetapi untuk kebutuhan **production**, tidak cukup hanya sekedar aplikasi bisa berjalan, masih dibutuhkan sistem yang handal yang mampu membuat aplikasi selalu available, tidak ada downtime, memiliki security yang handal, bisa menerima banyak traffic, dan sebagainya. Untuk itu diperlukan lah sebuah orkestrasi (*orchestration*) yang bertugas untuk manajemen itu semua. Maka yang bisa kita gunakan untuk itu semua adalah **kubernetes**.

Kelebihan Menggunakan Kubernetes

1. Service discovery and Load Balancing

Kubernetes dapat mengekspose container menggunakan nama DNS atau menggunakan IP address server itu sendiri. Jika trafik ke container besar, kubernetes akan otomatis melakukan load balancing trafik (penyeimbangan beban) dan melakukan distribusi trafik sehingga aplikasi selalu stabil.

2. Storage orchestration

Kubernetes memungkinkan untuk *me-mount* sistem storage yang diinginkan, seperti lokal storage, atau storage dari cloud provider seperti aws, dan alicloud.

3. Automated rollouts and rollbacks

Deployment di kubernetes bisa menggunakan file YAML. File YAML tersebut untuk melakukan deployment lainnya atau untuk mengubah deployment yang telah ada. Atau bisa melakukan rollback jika memiliki file YAML yang terdahulu. Sehingga semua terdokumentasi dan semua bisa dilakukan dengan otomatis.

4. Automatic bin packing

Kubernetes memberi izin untuk memberikan container dengan CPU dan RAM secara spesifik. Tujuannya ketika container memiliki sumber daya yang ditentukan, Kubernetes dapat membuat keputusan yang lebih baik dalam mengelola sumber daya untuk container.

5. Self-healing

Kubernetes merestart container yang gagal, menggantikan dan menjadwalkan ulang container saat node worker mati/tidak menanggapi request, dan kubernetes tidak akan memberikan trafik ke container sampai container benar-benar siap untuk menerima request (ready to serve).

6. Secret and configuration management

Kubernetes memungkinkan untuk menyimpan data-data yang sensitif, seperti password, auth token dan ssh keys. Anda dapat membangun dan mengupdate *secret* tanpa harus membangun ulang container image dan tanpa mengekspose *secret* konfigurasi yang telah ada sebelumnya.

Konsep Kubernetes

Dalam Cluster Kubernetes terdapat beberapa node. Node merupakan Server, baik server fisik atau Virtual Private Server (VPS) yang digunakan untuk menjalankan kubernetes. Untuk membuat cluster kubernetes dibutuhkan setidaknya 1 server untuk **kubernetes-master** atau **node-master** dan 1 server untuk **kubernetes-node** atau **node-1**.

Kubernetes Master

Kubernetes master adalah server yang bertindak sebagai node-master. Kubernetes master menjalankan tiga komponen yaitu *kube-apiserver*, *kube-controller-manager*, dan *kube-scheduler*.

Komponen Master

1. **kube-apiserver**, merupakan *front-end* dari *control plane* Kubernetes dan merupakan komponen di node master yang mengekspos API Kubernetes. Komponen ini didesain agar dapat di-scale secara horizontal. Server API Kubernetes memvalidasi dan mengonfigurasi data untuk objek api yang mencakup pod, services, replication controller, dan lainnya.
2. **kube-controller-manager**, merupakan komponen di node master yang menjalankan kontroler. Secara logis, setiap kontroler adalah sebuah proses yang berbeda, tetapi untuk mengurangi kompleksitas, kontroler-kontroler ini dikompilasi menjadi sebuah *binary* yang dijalankan sebagai satu proses.
3. **kube-scheduler**, merupakan komponen di master yang bertugas mengamati pod yang baru dibuat dan belum di-assign ke suatu node dan kemudian akan memilih sebuah node dimana pod baru tersebut akan dijalankan.

Kubernetes Nodes

Kubernetes nodes adalah semua server non-master yang terdaftar di server master. Kubernetes nodes menjalankan dua komponen, yaitu kubelet dan kube-proxy.

1. kubelet, merupakan agen yang dijalankan pada setiap node di kluster dan bertugas memastikan kontainer dijalankan di dalam pod.
2. kube-proxy, bertugas membantu abstraksi service Kubernetes melakukan tugasnya. Hal ini terjadi dengan cara memelihara aturan-aturan jaringan (network rules) serta meneruskan koneksi yang ditujukan pada suatu host.

Kubernetes architecture

