

a. Perbedaan linux kernel dan linux distro

Linux kernel adalah inti dari sistem operasi linux yang mengatur komunikasi antara hardware dan software sehingga tanpa adanya kernel maka aplikasi tidak bisa berkomunikasi dengan CPU, RAM, penyimpanan atau perangkat lain. Kernel bekerja dengan cara menyediakan driver untuk mengenali hardware, menerjemahkan instruksi dari program ke bahasa yang dimengerti perangkat keras serta mengatur jalannya proses, penggunaan memori dan sistem file.

Sedangkan Linux distro adalah variasi sistem operasi linux. Keberadaan distro memudahkan pengguna agar tidak perlu menyusun sistem dari nol, dan tiap distro biasanya dibuat untuk tujuan tertentu, misalnya ubuntu untuk pemula, debian untuk server, atau linux untuk keamanan.

b. FHS atau Filesystem Hierarchy Standard adalah aturan standar penempatan lokasi file dan direktori yang ada pada sistem operasi. Sistem berkas diklasifikasikan berdasarkan dua parameter: shareable – unshareable dan static – variable. Dengan adanya FHS penggunaan di setiap distro tetap memakai pola dasar yang sama dan aplikasi bisa berjalan di banyak distro tanpa harus tahu letak file manual. Selain itu user atau admin menjadi tidak bingung file penting ada di folder mana. Berikut contoh kegunaan dari masing-masing direktori

- / = Direktori yang berguna untuk menampung semua file yang terdapat dalam sistem linux.
- /bin = Direktori yang berisi file-file yang dapat langsung dijalankan
- /boot = Direktori yang digunakan saat booting Linux.
- /dev = Direktori yang berisi file-file dari hardware yang dimiliki komputer. misal /dev/ adalah file dari harddrive dan /dev/fd0 adalah file dari floppy
- /etc = Direktori yang berisi file yang digunakan untuk mengkonfigurasi linux. File ini biasanya dalam bentuk file naskah atau teks.
- /home = direktori ini digunakan untuk menyimpan semua direktori pengguna
- /lib = Direktori yang berisi file-file library yang dibutuhkan oleh program pada direktori root.
- /mnt = Direktori yang berisi direktori yang dapat anda gunakan untuk memounting perangkat peripheral yang dimiliki PC.
- /proc = Direktori yang berisi buffer file yang digunakan untuk menginformasi proses yang sedang berjalan pada sistem linux
- /root = direktori home untuk root
- /sbin = Direktori yang berisi file-file yang dijalankan oleh sistem linux
- /tmp = Direktori yang berisi file-file sementara

- /usr = Direktori yang berisi file-file yang dapat dijalankan atau berorientasi kepada semua pengguna.
- /var = Direktori yang berisi file data yang dapat berubah-ubah saat linux sedang dalam proses.

c. sistem permission dan owner di linux

Di sistem Linux, setiap file/direktori punya hak akses(permissions) yang melekat pada 3 kategori:

- Owner merupakan pemilik file yang berisikan sebuah user.
- Group merupakan sebuah pengelompokan user dalam sistem linux.
- Other merupakan pengguna lain yang tidak termasuk sebagai owner dan group.

mengubah kepemilikan file, user dapat menggunakan perintah chown yang biasa digunakan untuk merubah kepemilikan user dan juga grup dari file atau folder.

Contoh: `chown <new_user_name>:<new_user_group> <filename>`

Di setiap file dan folder linux memiliki 3 izin untuk ketiga jenis pemilik yaitu:

- Read untuk melihat dan menyalin baik isi dari file atau susunan file
- Write untuk mengubah isi dari file/menambah dan menghapus
- Execute untuk menjalankan file/masuk ke dalam direktori

Untuk merubah permission dengan menggunakan chmod ada 2 cara yaitu dengan menggunakan absolute mode dan symbolic mode. Absolute mode direpresentasikan dalam bentuk numerik yaitu :

- - (no permission) = 0
- X (execute) = 1
- W (write) = 2
- R (Read) = 4

Contoh : `chmod 755 file.sh`

Sedangkan symbolic mode direpresentasikan dalam bentuk simbol:

- Izin direpresentasikan dalam bentuk alfabet(x, w, r)
- Target perubahan direpresentasikan dalam bentuk alfabet(u, g, o, a)
- Modifikasi direpresentasikan dalam bentuk simbol aritmatika(+, -, =)

Contoh: `chmod u+x file.txt`

d. Prinsip enkripsi pada SSH

SSH adalah cara aman untuk tersambung ke komputer lain lewat internet. Prinsip enkripsi pada ssh melibatkan tiga jenis utama enkripsi untuk mengamankan komunikasi antara klien dan server :

- Enkripsi simetris (kunci bersama) menggunakan satu kunci untuk mengenkripsi dan mendeskripsi data
- Enkripsi asimetris (kunci ganda) menggunakan sepasang kunci (publik dan privat)
- Hashing adalah enkripsi satu arah yang digunakan untuk memverifikasi integritas data

e. Perbedaan antara HTTP dan HTTPS

HTTP merupakan sebuah protokol jaringan aplikasi yang digunakan untuk mendistribusikan informasi antara server dengan client. Sedangkan HTTPS merupakan hasil pengembangan dari versi HTTP sebelumnya, HTTPS memiliki fungsi keamanan yang lebih ketat sehingga membuat client merasa aman dalam mengakses berbagai konten web. Secara default HTTP berjalan di port 80 sedangkan HTTPS di port 443. Kemudian untuk melakukan transport data, HTTP memanfaatkan plain-text sedangkan HTTPS memanfaatkan SSL Handshake terlebih dahulu kemudian melakukan enkripsi data berdasarkan SSL Certificate yang berhasil diverifikasi. Di HTTP, tools network dapat melihat secara jelas apapun yang ditransfer dengan menggunakan protokol HTTP, sementara di HTTPS tools tidak bisa melakukan inspeksi secara langsung selain melakukan intercept network dengan menggunakan custom SSL Certificate.

f. Docker OCI Compliance Standard

OCI (open container initiative) adalah proyek sumber terbuka yang bertujuan untuk menciptakan standar terbuka untuk kontainerisasi. Standar ini mencakup spesifikasi untuk format gambar kontainer (misalnya format .tar.gz yang digunakan Docker) dan runtime kontainer (misalnya, komponen yang menjalankan kontainer di sistem operasi)

Jadi Docker OCI Compliance standard artinya Docker patuh pada standar OCI. OCI pada Docker memastikan bahwa format dan spesifikasi gambar kontainer serta runtime yang digunakan oleh Docker compatible dengan standar terbuka. Ini berarti gambar Docker dapat dijalankan di berbagai platform dan runtime yang mendukung standar OCI, meningkatkan portabilitas dan interoperabilitas kontainer.

g. Perbedaan antara Container dan VM

Container dan VM adalah teknologi virtualisasi yang memungkinkan kita menjalankan beberapa aplikasi pada satu mesin fisik, tetapi mereka memiliki perbedaan mendasar.

VM memvirtualisasikan seluruh perangkat keras, termasuk sistem operasi, sementara kontainer memvirtualisasikan sistem operasi saja, memungkinkan berbagai kernel OS yang sama dengan host. Akibatnya kontainer lebih ringan, lebih cepat dan lebih efisien dalam penggunaan sumber daya dibanding VM. Selain itu, untuk portabilitas VM lebih sulit dipindahkan karena ukurannya yang besar dan dependensi sistem operasi. Namun VM menyediakan isolasi yang lebih kuat dibanding container karena setiap VM memiliki sistem operasi sendiri. Dan untuk keamanan VM juga dianggap

lebih aman karena isolasi yang lebih kuat, sedangkan container, meskipun menawarkan isolasi namun jika terjadi kerentanan pada kernel OS host maka kerentanannya bisa lebih tinggi.

h. Definisi dan manfaat dari image layer pada docker

Image layer pada Docker adalah bagian-bagian terpisah yang menyusun sebuah image Docker. Setiap layer mempresentasikan perubahan pada sistem berkas yang dilakukan oleh instruksi tertentu dalam DockerFile. Layer-layer ini ditumpuk untuk membuat image akhir, dan bersifat read-only setelah di buat.

Manfaat dari image layer pada docker:

- Penggunaan kembali yang efisien
Lapisan yang dibagikan antar gambar di cache dan tidak diduplikasi, sehingga mengurangi kebutuhan penyimpanan.
- Penerapan yang lebih cepat
Lapisan yang tidak diubah digunakan kembali, yang mempercepat pembangunan dan mengurangi waktu push/pull
- Pembaruan bertahap
Hanya lapisan yang dimodifikasi yang ditransfer, meminimalkan penggunaan bandwidth
- Konsistensi pembangunan
Setiap lapisan memetakan ke instruksi Dockerfile tertentu, yang membuat pembangunan dapat direproduksi
- Pemecahan masalah yang disederhanakan
Struktur berlapis membantu mengidentifikasi langkah mana yang menimbulkan masalah
- Modularitas dan portabilitas
Lapisan membantu menjaga modularitas, sehingga gambar lebih mudah dipindahkan antar lingkungan. Gambar yang sama berjalan dengan cara yang sama dimana-mana.

i. Kegunaan dari penggunaan docker volume dan network beserta contohnya

Docker volume berguna untuk menyimpan file-file atau data pengguna dilokasi yang relatif aman.

Contohnya penyimpanan basis data seperti MySQL dan PostgreSQL

Docker network berguna untuk mengatur container mana yang harus ada di dalam 1 ruang lingkup yang sama

Contohnya Aplikasi berbasis mikroservis

j. Definisi dan tujuan dari penggunaan web server dan reverse-proxy

Web server adalah sebuah program untuk menyediakan akses terhadap resource dengan menggunakan protokol HTTP atau HTTPS dan berjalan sebagai sebuah service pada komputer server. Bertujuan untuk menyimpan konten, keamanan dasar, mengelola permintaan serta menyajikan konten web.

Web server biasanya memiliki fitur load balancing dan reverse proxy. Reverse proxy adalah server yang bertindak sebagai perantara antara klien dengan satu atau lebih server web. Reverse proxy menerima permintaan dari klien dan meneruskan ke server web yang sesuai, lalu mengembalikan respons dari server web ke klien. Tujuan dari reverse proxy adalah untuk keamanan, performa, skalabilitas dan membatasi akses ke konten tertentu