

Cloud Computing

Cloud Infrastructure and Deployment

Raih Mimpi #TanpaBatas



Infrastruktur dan Deployment Cloud Computing

- ☐ Infrastruktur Cloud
- ☐ Pendahuluan Deployment
- ☐ Strategi Deployment
- ☐ Virtualisasi dan Kontainerisasi
- ☐ CI/CD
- ☐ Studi Kasus

Objektif sesi

- Memahami konsep dasar dan implementasi infrastruktur cloud
- Memahami cara kerja, manajemen, dan penerapan model deployment dalam cloud
- Menganalisis studi kasus di industri perbankan dan merancang solusi infrastruktur cloud yang efisien
- Membekali peserta dengan keterampilan praktis dalam deployment dan manajemen infrastruktur cloud

Expected output

- Peserta dapat menjelaskan konsep dan komponen penting dari infrastruktur cloud
- Peserta dapat merencanakan, melaksanakan, dan mengelola deployment dalam lingkungan cloud
- Peserta dapat menganalisis kasus nyata dalam industri perbankan dan merancang solusi infrastruktur cloud yang sesuai
- Peserta memiliki keterampilan praktis dalam penggunaan alat dan teknologi cloud terkini

Agenda 1

Infrastuktur Cloud



Definisi dan Komponen

Infrastruktur cloud adalah model penyediaan IT yang menyediakan akses jaringan on-demand ke pool sumber daya komputasi yang dapat dikonfigurasi, seperti jaringan, server, storage, aplikasi, dan layanan. Komponen ini dapat dengan cepat dialokasikan dan dikelola dengan upaya minimal, sering kali melalui sistem manajemen menggunakan dashboard atau interface API.

Hardware

Ini termasuk server fisik yang berfungsi sebagai mesin komputasi dan penyimpanan dalam infrastruktur cloud

Virtualisasi

Teknologi ini memungkinkan pembuatan versi virtual dari sumber daya fisik, seperti mesin virtual (VMs) yang berfungsi sebagai server individu dalam infrastruktur cloud

Networking

Ini mencakup semua elemen yang diperlukan untuk komunikasi antara komponen infrastruktur cloud, seperti switch, router, dan firewall

Storage

Ini adalah tempat data disimpan dalam infrastruktur cloud, baik dalam format block storage, file storage, atau object storage

Review

Model Layanan Cloud

Model layanan cloud merujuk pada cara spesifik layanan TI disediakan melalui infrastruktur cloud



**Infrastructure
as a Service**



**Platform as a
Service**



**Software as a
Service**



Model Layanan Cloud dalam Industri

Infrastructure as a Service (IaaS)

- Perusahaan bisa menyewa infrastruktur IT seperti server, storage dan jaringan dari penyedia cloud. Manfaat yang didapatkan adalah mengurangi biaya investasi awal dalam hardware
- Contoh : Perusahaan e-commerce menggunakan **AWS** untuk mengelola dan menyimpan data transaksi pelanggan

Platform as a Service (PaaS)

- PaaS memberikan platform dan *environment* yang memungkinkan pengembang untuk membangun aplikasi dan layanan via internet
- Contoh : Developer game mobile menggunakan **Google App Engine** untuk membangun, menguji dan mengelola aplikasi mereka

Software as a Service (SaaS)

- SaaS memungkinkan pengguna untuk terhubung dan menggunakan aplikasi berbasis cloud melalui internet.
- Contoh: Sekolah menggunakan **Google Classroom** untuk menyediakan material dan soal latihan / ujian untuk pembelajaran jarak jauh

Model Serving Cloud Computing

Private Cloud

- Private Cloud adalah model serving yang mendedikasikan seluruh infrasturktur kepada satu organisasi
- Keuntungan : Kontrol penuh atas data dan proses
- Kekurangan : Mahal, serta perlu pengetahuan teknis untuk mengelola dan maintenance cloud

Public Cloud

- Public Cloud adalah model serving yang memungkinkan user membayar hanya untuk apa yang digunakan oleh user, selain itu user mendapatkan fleksibilitas untuk meningkatkan / mengurangi sumber daya
- Keuntungan: Skalabilitas, tidak ada maintenace perangkat keras, biaya lebih murah dibandingkan private cloud
- Kekurangan: Kontrol lebih terbatas atas data dan proses, potensi masalah keamanan

Model Serving Cloud Computing

Hybrid Cloud

- Hybrid cloud merupakan kombinasi dari private cloud dan public cloud. Dalam kasus ini, data dan aplikasi dapat berpindah antara private dan public cloud
- Keuntungan : Fleksibilitas, kontrol atas data yang sensitif
- Kekurangan : Kompleks, memerlukan teknologi jaringan untuk memastikan semua elemen dapat berinteraksi

Community Cloud

- Community cloud memungkinkan banyak pelanggan bekerja pada proyek bersama dan aplikasi milik komunitas, di mana diperlukan infrastruktur cloud terpusat
- Keuntungan: Biaya dan manajemen dibagi antara anggota, keamanan yang disesuaikan dengan kebutuhan komunitas
- Kekurangan: Investasi awal yang tinggi, memerlukan kerjasama dan koordinasi antar komunitas

Top 10 Cloud Providers





Model Deployment Cloud Computing

Deployment dalam konteks cloud computing adalah proses memindahkan aplikasi atau perangkat lunak dari tahap pengembangan ke produksi, di mana aplikasi menjadi tersedia bagi pengguna akhir. Deployment melibatkan persiapan, pemasangan, pengujian, operasi, dan update aplikasi atau perangkat lunak dalam lingkungan produksi.

Life Cycle Deployment

Development

Di mana aplikasi atau perangkat lunak dikembangkan

Testing

Aplikasi atau perangkat lunak diuji untuk mengidentifikasi dan memperbaiki bug atau masalah

Deployment

Aplikasi atau perangkat lunak dipindahkan ke lingkungan produksi dan dibuat tersedia bagi pengguna

Maintenance

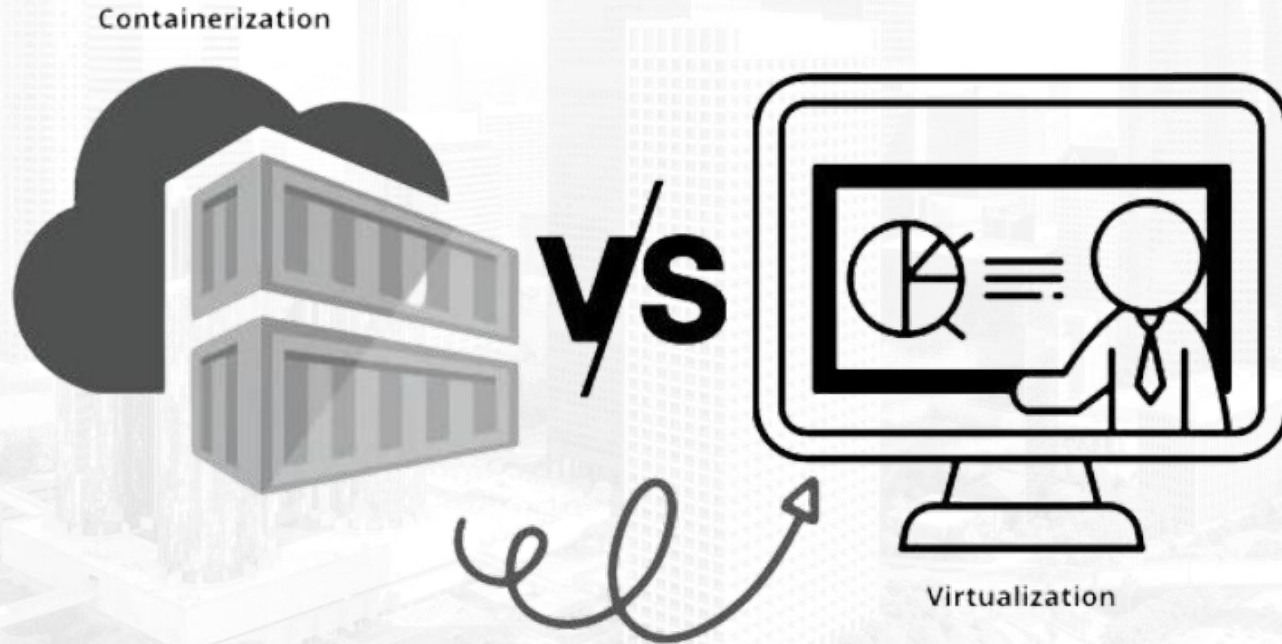
Aplikasi atau perangkat lunak diperbarui dan diperbaiki sepanjang waktu

Strategi Deployment

Strategi deployment digunakan untuk meminimalkan downtime dan risiko yang terkait dengan merilis versi baru aplikasi. Beberapa strategi populer meliputi:

- **Blue/Green Deployment:** Dalam model ini, ada dua lingkungan, "Blue" dan "Green", yang identik dan dapat ditukar. Versi baru aplikasi diterapkan ke lingkungan yang tidak aktif (misalnya, Green) dan setelah semua pengujian selesai, lalu lintas dialihkan dari lingkungan aktif saat ini (Blue) ke Green.
- **Canary Deployment:** Dalam strategi ini, versi baru aplikasi diterapkan ke subset kecil dari infrastruktur produksi untuk menilai kinerjanya sebelum diterapkan ke seluruh infrastruktur.
- **Rolling Deployment:** Versi baru aplikasi diterapkan ke infrastruktur produksi secara bertahap, menggantikan versi lama selama waktu tertentu.

Virtualisasi dan Kontainerisasi



Virtualisasi

Definisi:

Teknologi yang memungkinkan untuk menjalankan beberapa sistem operasi atau lingkungan virtual di atas perangkat keras fisik tunggal.

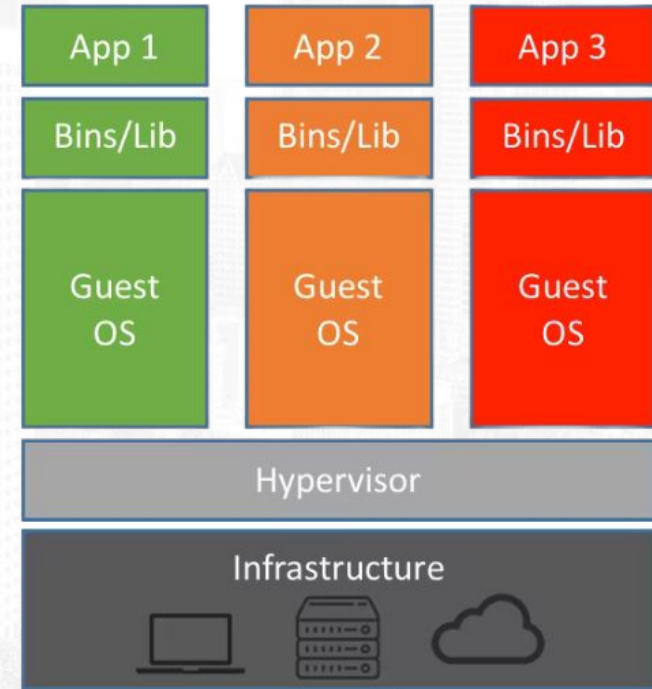
Komponen Utama:

Hypervisor: Perangkat lunak yang membuat dan mengelola mesin virtual (VM) di atas perangkat keras.

Mesin Virtual (VM): Entitas komputer virtual dengan sumber daya (CPU, memori, disk) yang dialokasikan dari mesin fisik.

Manfaat:

- Optimalisasi sumber daya perangkat keras.
- Isolasi aplikasi dan sistem operasi.
- Kemudahan dalam backup, migrasi, dan replikasi.



Kontainerisasi

Definisi:

Metode yang memungkinkan aplikasi dan dependensinya untuk dijalankan dalam kontainer yang terisolasi, namun berbagi kernel OS yang sama.

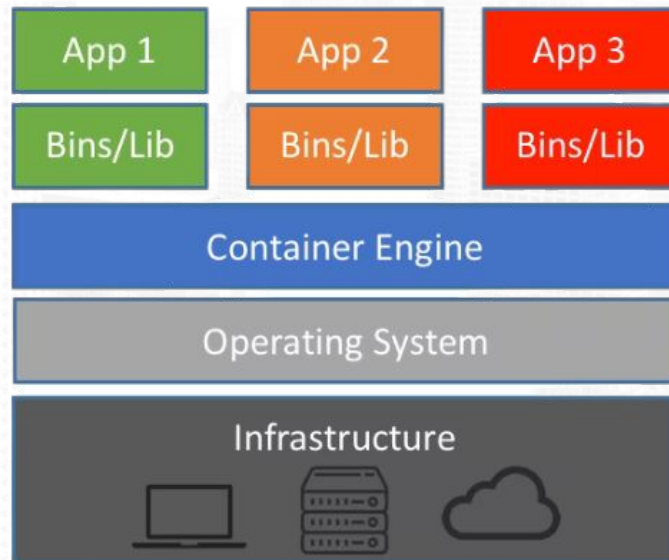
Komponen Utama:

Docker: Platform yang populer untuk mengembangkan, mengirim, dan menjalankan aplikasi dalam kontainer.

Kubernetes: Sistem orkestrasi kontainer terbuka yang otomatis menerapkan, menskalakan, dan mengelola aplikasi yang dikontainerisasi.

Manfaat:

- Efisiensi dan kecepatan yang lebih tinggi dibandingkan dengan VM.
- Konsistensi lingkungan antara pengembangan dan produksi.
- Scalability dan portabilitas aplikasi.



Virtualisasi vs Kontainersiasi

VIRTUALISASI

- Menjalankan banyak OS lengkap di atas perangkat keras.
- Overhead yang lebih tinggi karena setiap VM memiliki sistem operasi sendiri.
- Lebih aman karena VM sepenuhnya terisolasi.

KONTAINERISASI

- Semua kontainer berbagi OS yang sama, namun menjalankan proses dalam ruang lingkup terisolasi.
- Lebih ringan dan lebih cepat daripada VM.
- Lebih rentan jika ada kelemahan di level kernel OS.

Manajemen dan Operasi (Ops) Infrastruktur Cloud

Manajemen cloud melibatkan berbagai tugas, termasuk pengaturan dan pengelolaan sumber daya cloud, pelacakan penggunaan dan kinerja, serta pemecahan masalah dan pemeliharaan rutin

Otomatisasi dan Orkestrasi

Otomatisasi melibatkan penggunaan teknologi untuk melakukan tugas secara otomatis, yang dapat meningkatkan efisiensi dan mengurangi kesalahan manusia. Dalam konteks cloud, ini dapat mencakup otomatisasi tugas seperti penyebaran, pengelolaan, dan pemantauan sumber daya.



Tools otomatisasi : Terraform, Ansible, Kubernetes, dll

Disaster Recovery dan High Availability

Disaster recovery adalah strategi dan proses yang digunakan untuk memulihkan sistem dan data jika terjadi bencana atau kegagalan besar. Dalam cloud, ini bisa melibatkan replikasi data dan aplikasi di beberapa wilayah atau zona ketersediaan, serta penggunaan layanan dan teknologi khusus seperti snapshot dan backup.



Tools : Apache Cassandra, Hadoop, dll

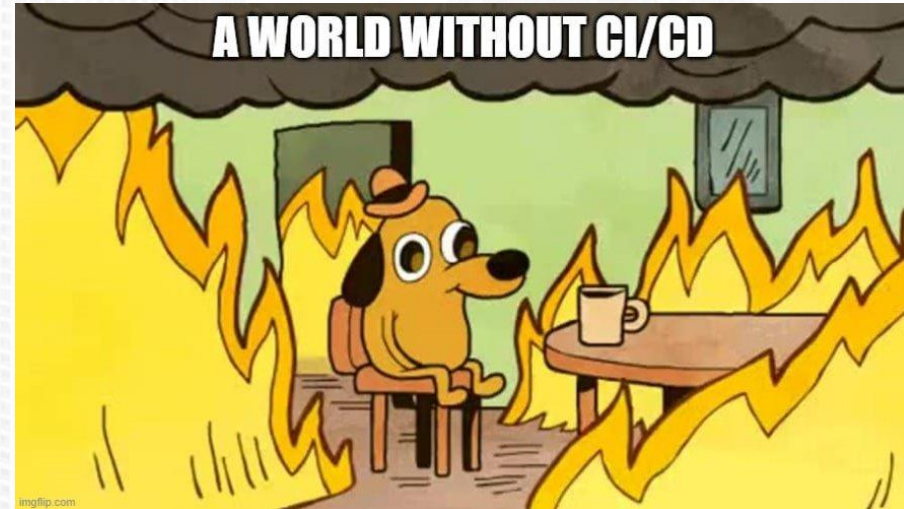


Agenda 2

Konsep CI/CD

Teknologi Deployment : CI/CD

Teknologi deployment mencakup serangkaian perangkat lunak dan praktik yang membantu dalam otomatisasi proses pengembangan, pengujian, dan deployment aplikasi. Konsep utama di sini adalah Continuous Integration (CI) dan Continuous Deployment (CD)



Teknologi Deployment : CI/CD

CI



Continuous **Integration**

CD



Continuous **Deployment**

ATAU

CD



Continuous **Delivery**

CI/CD Stack



Jenkins



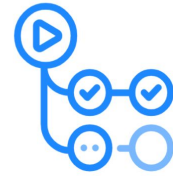
Azure Pipelines



CI/CD



Travis CI



GitHub Actions



TeamCity

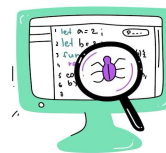
CI/CD Overview



DEPLOY



CODE
CHANGES



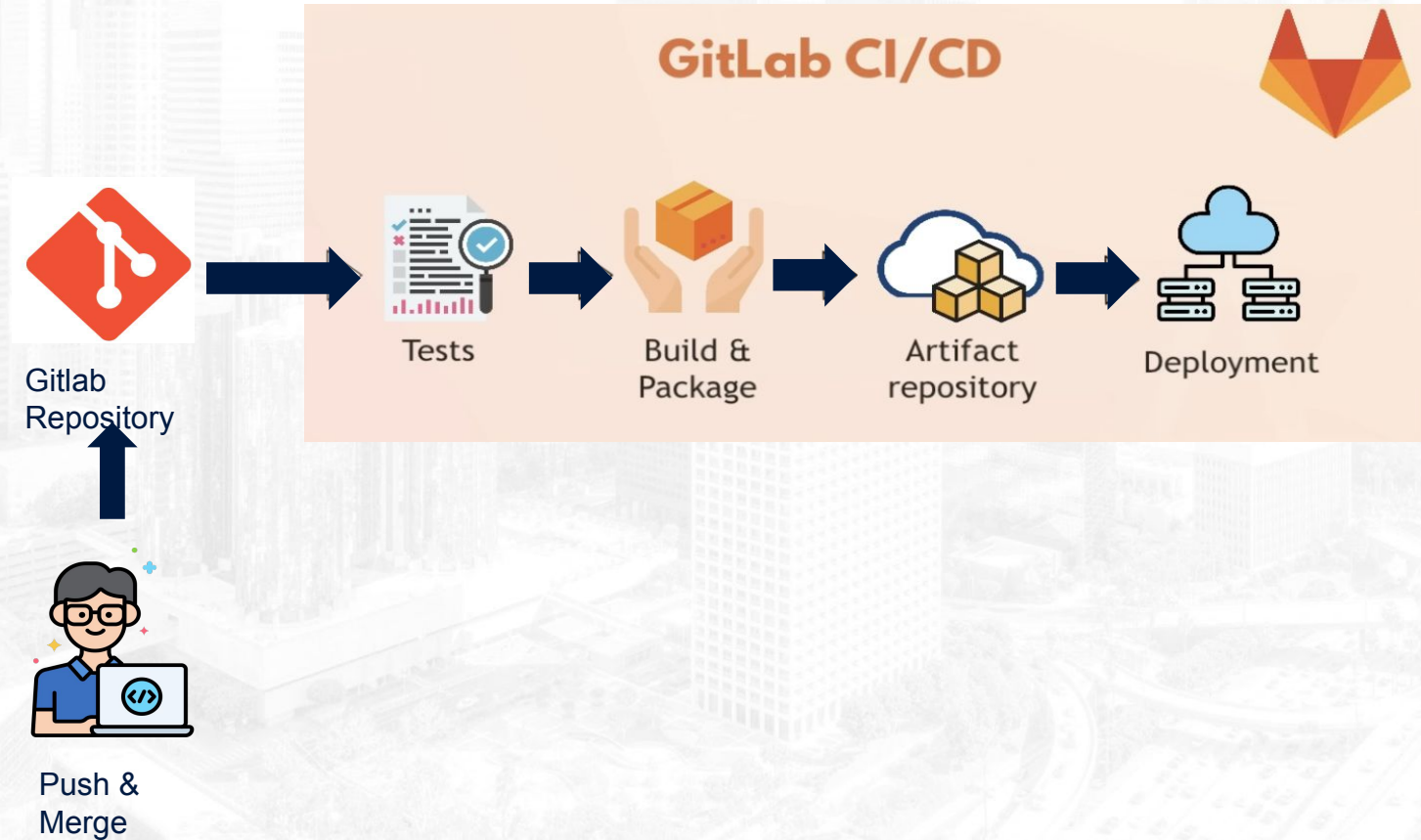
TEST



BUILD



CI/CD Overview



GitLab Architecture



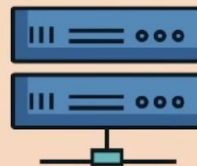
GitLab Instance

or

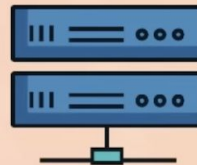
GitLab Server



job-1



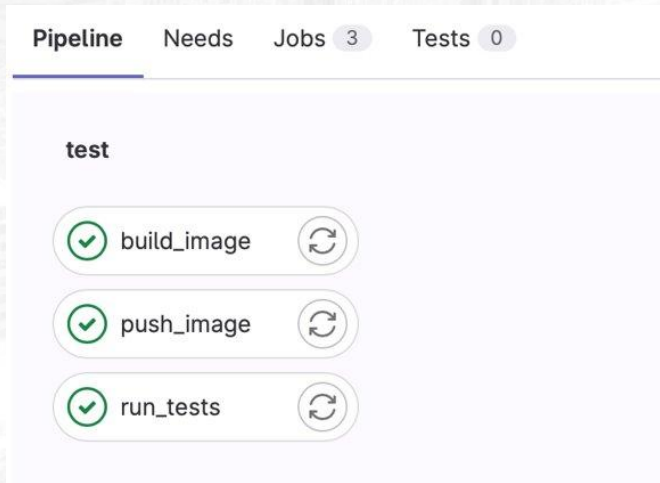
job-2



GitLab Runners

Konsep Dasar : Jobs

- Unit dasar dalam sebuah pipeline yang berisi satu atau beberapa task
- Dapat dijalankan secara simultan maupun sekuensial
- Independen antar Jobs



```
run_tests:
  before_script:
    - echo "Persiapan sebelum test ..."
  script:
    - echo "Afif Ganteng ..."
  after_script:
    - echo "menyelesaikan test"

build_image:
  script:
    - echo "membuat docker image ..."
    - echo "tagging docker ..."

push_image:
  script:
    - echo "Login Docker Registry ..."
    - echo "Push docker ke image registry"
```

Konsep Dasar : Stages

- Kumpulan jobs yang diorganisir
- Dapat berisi satu atau beberapa jobs
- Dijalankan secara sekuensial



stages:

- test
- build
- deploy

run_unit_tests:

```
stage: test
before_script:
  - echo "Persiapan sebelum test ..."
script:
  - echo "Afif Ganteng ..."
after_script:
  - echo "menyelesaikan test"
```

run_lint_tests:

```
stage: test
before_script:
  - echo "Persiapan sebelum test ..."
script:
  - echo "Afif Ganteng ..."
after_script:
  - echo "menyelesaikan test"
```

build_image:

```
stage: build
script:
  - echo "membuat docker image ..."
  - echo "tagging docker ..."
```

push_image:

```
stage: deploy
script:
  - echo "Login Docker Registry ..."
  - echo "Push docker ke image registry"
```

Hands on Deployment Cloud dengan Docker & CI/CD



Thank you