

Nama : Ni Made Cantika Dewi

NIM : 2201010733

Kelas : Q

### **UAS Cloud Computing**

1. Pendapat saya tentang elastisitas sebagai salah satu karakteristik utama dalam cloud computing? Elastisitas dalam cloud computing adalah kemampuan sistem untuk mengecilkan atau memperluas sumber daya infrastruktur secara potensial sebagaimana diperlukan untuk menyesuaikan dengan variasi beban kerja secara mandiri, memastikan efisiensi sumber daya. seperti pegas yang bisa meregang dan menyusut sesuai kebutuhan. Ini penting karena memungkinkan penyesuaian sumber daya komputasi sesuai dengan permintaan yang berubah. Menurut saya elastisitas menjadi penting dalam pengelolaan sumber daya karena dengan elastisitas, kita bisa mengoptimalkan penggunaan sumber daya dan menghindari pemborosan. Selain itu, ini juga membuat layanan lebih responsif terhadap lonjakan lalu lintas atau permintaan yang tidak terduga. Elastisitas membantu efisiensi dan cost-effectiveness dalam pengelolaan sumber daya cloud.
2. Pendapat saya mengenai perbandingan antara Infrastructure as a Service (IaaS) dan Software as a Service (SaaS) adalah IaaS memberikan infrastruktur dasar seperti server dan penyimpanan, sementara SaaS menyediakan aplikasi siap pakai. Sebuah organisasi sebaiknya memilih IaaS jika mereka membutuhkan kontrol lebih besar atas infrastruktur dan ingin mengelola sendiri sistem operasional, basis data, dan aplikasi. Ini cocok untuk proyek-proyek yang memerlukan kebebasan konfigurasi dan skala yang lebih fleksibel. Sementara itu, SaaS cocok untuk organisasi yang ingin fokus pada penggunaan aplikasi tanpa harus memikirkan pengelolaan infrastruktur di belakangnya. Ini sangat efisien untuk aplikasi umum seperti email, manajemen proyek, atau keuangan. Pilihan antara keduanya tergantung pada kebutuhan spesifik organisasi. Jika kontrol dan kustomisasi penting, IaaS bisa lebih baik. Jika kecepatan implementasi, keamanan, dan pemeliharaan minimalisasi adalah prioritas, SaaS bisa menjadi pilihan yang lebih baik.

3. Tanggapan saya terhadap konsep containerization menggunakan Docker adalah konsep containerization, terutama dengan menggunakan Docker, adalah gebrakan besar dalam dunia pengembangan dan pengelolaan aplikasi. Bayangkan kontainer sebagai paket yang berisi segala sesuatu yang diperlukan oleh aplikasi, dari kode sampai dependensi, dan bisa dijalankan secara konsisten di berbagai lingkungan. Docker mempermudah pembuatan dan menjalankan arsitektur layanan mikro terdistribusi, menerapkan kode dengan pipeline integrasi dan pengiriman berkelanjutan yang terstandarisasi, membangun sistem pemrosesan data dengan skalabilitas yang tinggi. Dengan Docker, proses pengembangan menjadi lebih konsisten dan dapat diulang. Developer dapat membangun dan menguji aplikasi di lingkungan lokal yang identik dengan lingkungan produksi, mengurangi risiko kesalahan akibat perbedaan konfigurasi.

Menurut saya keberadaan kontainer dapat mempermudah proses pengembangan dan pengelolaan aplikasi yaitu Pengelolaan aplikasi juga jauh lebih mudah. Kontainer memungkinkan pengelolaan sumber daya dengan lebih efisien, dapat dijalankan atau dihentikan dengan cepat, dan memudahkan skalabilitas horizontal. Ini membuat deployment lebih cepat dan lebih andal. Selain itu, kontainer juga membantu dalam memecah monolitik aplikasi menjadi komponen yang dapat dielaborasi dan di-deploy secara independen, meningkatkan fleksibilitas dan kemudahan pemeliharaan. Secara keseluruhan, Docker dan kontainerisasi membawa perubahan positif dalam siklus hidup pengembangan dan pengelolaan aplikasi.

4. Menurut saya redundansi dan ketersediaan tinggi menjadi faktor krusial dalam lingkungan cloud computing karena redundansi dan ketersediaan tinggi menjadi krusial dalam cloud computing karena organisasi mengandalkan infrastruktur yang tersedia secara online untuk menjalankan bisnis mereka. Ketika layanan atau aplikasi tidak tersedia, dapat berdampak signifikan pada produktivitas, reputasi, dan kepuasan pelanggan. Redundansi, seperti penggandaan server dan penyimpanan data, memberikan lapisan perlindungan jika ada kegagalan pada satu titik. Ini menghindari single points of failure dan memastikan bahwa jika satu komponen gagal, yang lainnya dapat mengambil alih dengan cepat.

Tantangan dalam mencapai tingkat ketersediaan yang diinginkan melibatkan kompleksitas sistem dan biaya. Menerapkan redundansi membutuhkan investasi tambahan dalam infrastruktur dan manajemen. Selain itu, membuat sistem benar-benar bebas dari downtime adalah suatu tantangan karena selalu ada kemungkinan kegagalan. Namun, dengan desain yang baik, otomatisasi, dan pemantauan yang efektif, organisasi dapat mengatasi tantangan ini. Penting untuk terus meningkatkan dan menguji kehandalan sistem agar dapat merespons perubahan dan menjaga tingkat ketersediaan yang tinggi.

5. Keamanan antara cloud public dan private yaitu cloud private menawarkan lebih banyak kontrol langsung atas keamanan karena dioperasikan secara eksklusif untuk satu organisasi. Sementara itu, cloud public memiliki keuntungan dalam skala, fleksibilitas, dan biaya, namun memerlukan kepercayaan terhadap penyedia layanan cloud.

Suatu organisasi harus memilih cloud private ketika keamanan dan privasi data sangat krusial, dan mereka memiliki kebutuhan untuk mematuhi regulasi ketat. Ini sering terjadi di industri keuangan, kesehatan, atau sektor public sedangkan cloud public cocok untuk organisasi yang menghargai skala, elastisitas, dan inovasi yang cepat. Jika data yang diolah bukan termasuk yang sangat rahasia dan jika organisasi memiliki kemampuan untuk mengelola keamanan menggunakan kontrol yang disediakan oleh penyedia cloud, maka cloud public bisa menjadi pilihan yang baik.

6. Pendapat saya mengenai perbedaan antara virtualisasi dan containerization dalam konteks pengembangan dan implementasi aplikasi adalah virtualisasi melibatkan menciptakan mesin virtual yang dapat menjalankan sistem operasi lengkap. Setiap mesin virtual memiliki kernel dan menyediakan isolasi penuh antar aplikasi. Ini bisa memakan banyak sumber daya. Sedangkan Containerisasi, seperti yang digunakan oleh Docker, lebih ringan. Kontainer berbagi kernel dari sistem host, membuatnya lebih efisien dan cepat. Mereka membungkus aplikasi bersama dengan dependensinya dan berbagi OS host, memastikan portabilitas yang tinggi antar lingkungan. Kelebihan docker dan kontainer termasuk kecepatan deployment, efisiensi penggunaan sumber daya, dan portabilitas. Mereka juga memungkinkan isolasi aplikasi, memastikan bahwa perubahan dalam satu kontainer tidak memengaruhi yang lain.

7. Tanggapan saya terhadap konsep skalabilitas horizontal dalam arsitektur cloud adalah skalabilitas horizontal dalam arsitektur cloud, atau kemampuan untuk menangani lonjakan lalu lintas dengan menambah lebih banyak instans, adalah konsep yang sangat positif. Ini memberikan fleksibilitas dan keandalan tinggi dalam mengelola beban kerja yang berubah-ubah. Dengan skalabilitas horizontal, sistem dapat dengan cepat menyesuaikan diri terhadap peningkatan permintaan dengan menambahkan lebih banyak instans atau node ke dalam infrastrukturnya. Ini memungkinkan pemanfaatan sumber daya yang lebih efisien dan menghindari bottleneck yang dapat terjadi pada sistem yang tidak dapat diskalakan. Manfaat signifikan dalam kemampuan sistem untuk menangani lonjakan lalu lintas dengan menambahkan lebih banyak instans adalah kemampuan untuk menjaga ketersediaan layanan tanpa mengorbankan kinerja. Saat terjadi lonjakan lalu lintas, sistem dapat secara otomatis menambahkan instans baru dan mendistribusikan beban kerja secara merata, mencegah kegagalan dan memberikan pengalaman pengguna yang lebih baik.
8. Tanggapan saya terhadap perbandingan antara Software as a Service (SaaS) dan Function as a Service (FaaS) (Software as a Service) dan FaaS (Function as a Service) adalah dua model layanan cloud yang berbeda dan ditujukan untuk kebutuhan yang berbeda pula. SaaS cocok untuk aplikasi yang menyediakan fungsionalitas end-to-end dan berjalan secara terus menerus. Misalnya, aplikasi manajemen proyek atau email. SaaS menyediakan akses langsung ke aplikasi yang siap pakai tanpa memerlukan pengelolaan infrastruktur oleh pengguna. FaaS cocok untuk skenario dimana ingin menjalankan potongan kode (fungsi) secara otomatis dan on-demand tanpa harus memikirkan infrastruktur di belakangnya. FaaS memungkinkan pengguna hanya membayar untuk eksekusi kode yang sebenarnya, sehingga lebih efisien untuk tugas-tugas spesifik yang memerlukan skala yang cepat dan efisien. Kapan atau waktu yang tepat dan lebih masuk akal menggunakan FaaS daripada SaaS atau sebaliknya tergantung pada kebutuhan spesifik proyek. Jika memerlukan aplikasi lengkap dengan antarmuka pengguna dan fungsionalitas end-to-end, SaaS mungkin lebih sesuai. Jika hanya perlu menjalankan potongan kode spesifik tanpa perlu mengelola infrastruktur, FaaS bisa menjadi pilihan yang lebih baik.

9. Menurut saya cara Docker Hub memfasilitasi manajemen container yaitu docker hub memainkan peran kunci dalam memfasilitasi manajemen kontainer, terutama dengan menggunakan Docker. Ini adalah repositori publik yang menyediakan tempat untuk menyimpan, berbagi, dan mendistribusikan kontainer Docker. Beberapa keuntungan utama dari menggunakan Docker Hub dan repositori publik serupa melibatkan kolaborasi, distribusi, dan efisiensi.
  - a. Kolaborasi dan Berbagi: Docker Hub memungkinkan pengembang dan tim untuk dengan mudah berbagi kontainer. Ini mempermudah kolaborasi antar pengembang dan tim yang bekerja pada proyek yang sama atau mirip.
  - b. Distribusi Global: Repositori publik seperti Docker Hub menyediakan infrastruktur untuk mendistribusikan kontainer secara global. Pengguna dapat dengan mudah mengakses dan mengunduh kontainer dari berbagai lokasi, meningkatkan ketersediaan dan aksesibilitas.
  - c. Integrasi dengan CI/CD: Docker Hub dapat diintegrasikan dengan alat Continuous Integration/Continuous Deployment (CI/CD), memungkinkan otomatisasi proses build, uji, dan distribusi kontainer. Ini membantu dalam menyederhanakan siklus hidup pengembangan dan penyebaran aplikasi.
  - d. Versi dan Tagging: Docker Hub mendukung versi dan tagging, memungkinkan pengguna untuk mengelola versi kontainer dengan mudah. Ini memudahkan dalam memastikan bahwa semua anggota tim menggunakan versi yang sama dari aplikasi atau layanan.
10. Menurut pendapat saya, langkah-langkah konkret yang dapat diambil untuk meningkatkan keamanan data dalam cloud computing menjadi krusial, dan ada beberapa langkah konkret yang dapat diambil oleh organisasi untuk meningkatkannya yaitu
  - a. Enkripsi Data: Gunakan enkripsi untuk melindungi data saat transit dan saat istirahat. Pastikan bahwa data yang disimpan di cloud atau dikirim antar sistem dienkripsi menggunakan protokol keamanan yang kuat.
  - b. Manajemen Akses yang Ketat: Terapkan kebijakan manajemen akses yang ketat. Berikan hak akses sesuai kebutuhan, dan pastikan bahwa hanya orang yang berwenang yang dapat mengakses data kritis.

- c. Auditing dan Pemantauan: Lakukan pemantauan dan audit secara teratur untuk melacak aktivitas pengguna dan mendeteksi potensi ancaman keamanan. Log aktivitas dan lakukan analisis untuk mendeteksi anomali atau tanda-tanda intrusi.
- d. Keselamatan Fisik dan Infrastruktur: Pastikan penyedia cloud memiliki langkah-langkah keamanan fisik yang solid untuk melindungi pusat data mereka. Selain itu, pilih penyedia yang memiliki keamanan infrastruktur yang kuat.
- e. Backup dan Pemulihan Bencana: Rutin melakukan backup data dan memiliki rencana pemulihan bencana yang teruji. Ini memastikan bahwa data dapat dipulihkan dengan cepat dalam kasus kejadian tidak terduga atau serangan.
- f. Pembaruan Keamanan Reguler: Pastikan semua sistem dan perangkat lunak terus diperbarui dengan patch keamanan terbaru untuk mengatasi kerentanan potensial.
- g. Kesadaran Pengguna: Lakukan pelatihan keamanan untuk anggota tim dan pengguna akhir. Kesadaran pengguna yang baik dapat mencegah serangan sosial dan membantu melindungi data.
- h. Penilaian Keamanan dan Pengujian: Lakukan penilaian keamanan secara teratur dan uji penetrasi untuk mengidentifikasi dan mengatasi potensi kerentanan.
- i. Ketentuan Kontrak yang Jelas: Pastikan kontrak dengan penyedia cloud mencakup ketentuan keamanan yang jelas dan memadai. Tentukan tanggung jawab dan kewajiban masing-masing pihak terkait keamanan data.
- j. Keamanan Multi-Faktor (MFA): Aktifkan autentikasi multi-faktor untuk meningkatkan lapisan keamanan. Ini membutuhkan lebih dari sekadar kata sandi untuk mengakses data atau sistem.

organisasi dapat memastikan bahwa data mereka tetap aman di lingkungan cloud dengan menerapkan langkah-langkah tersebut, organisasi dapat memperkuat keamanan data mereka di lingkungan cloud. Penting untuk memahami bahwa keamanan adalah upaya bersama antara penyedia cloud dan organisasi pengguna.