UJIAN AKHIR SEMESTER

Nama: Jose Hermenegildo Da Costa

Nim:19101423

1. Mengapa elastisitas menjadi penting dalam pengelolaan sumber daya?

Elastisitas adalah kemampuan program berbasis cloud untuk memerlukan lebih banyak atau lebih sedikit sumber daya. Anda dapat menskalakan pemrosesan komputer, memori, dan kapasitas penyimpanan dalam komputasi awan untuk menyesuaikan dengan perubahan permintaan. Berkat skalabilitas, Anda tidak perlu khawatir tentang rekayasa puncak atau perencanaan kapasitas . Anda tidak perlu membeli atau memelihara peralatan tambahan karena elastisitas cloud menyelamatkan Anda dari membayar kapasitas yang kurang dimanfaatkan atau sumber daya yang menganggur.

Komputasi elastis lebih efektif dibandingkan teknologi lokal dalam banyak situasi. Selain itu, biasanya otomatis dan menjaga layanan berjalan lancar dengan mencegah jeda dan gangguan. Hal ini juga disebut sebagai elastisitas cepat dalam komputasi awan.

2. Saas perangkat lunak sebagai Layanan,
   perangkat Lunak sebagai Layanan, juga dikenal sebagai layanan aplikasi cloud, mewakili opsi yang paling umum digunakan untuk bisnis di pasar cloud . SaaS memanfaatkan internet untuk mengirimkan aplikasi, yang dikelola oleh vendor pihak ketiga, kepada penggunanya. Mayoritas aplikasi SaaS dijalankan langsung melalui browser web Anda, yang berarti aplikasi tersebut tidak memerlukan pengunduhan atau penginstalan apa pun di sisi klien.

   Iaas Layanan infrastruktur cloud, yang dikenal sebagai Infrastruktur sebagai Layanan (IaaS), terbuat dari sumber daya komputasi yang sangat skalabel dan otomatis. IaaS sepenuhnya merupakan layanan mandiri untuk mengakses dan memantau komputer, jaringan, penyimpanan, dan layanan lainnya. IaaS memungkinkan bisnis untuk membeli sumber daya sesuai permintaan dan kebutuhan daripada harus membeli perangkat keras secara langsung.

   Contoh umum Saas dan Iass.

   Saas : Google Workspace, Dropbox, Salesforce, Cisco WebEx, Setuju, GoToMeeting

   Iass : DigitalOcean, Linode, Rackspace, Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE)

3. Containerization menggunakan Docker adalah konsep yang sangat efektif dalam memudahkan pengembangan dan pengelolaan aplikasi. Dengan kontainer, Anda dapat mengemas aplikasi beserta dependensinya dalam lingkungan yang terisolasi. Ini memungkinkan aplikasi berjalan konsisten di berbagai lingkungan, dari pengembangan hingga produksi, mengatasi masalah kompatibilitas yang sering muncul.Keberadaan kontainer mempermudah pengembang untuk menyediakan aplikasi dengan cepat, karena dapat dijamin bahwa lingkungan di mana aplikasi berjalan akan konsisten. Selain itu, kontainer juga memfasilitasi penskalaan aplikasi dan pengelolaan sumber daya dengan lebih efisien.

Docker juga memungkinkan penggunaan gambar (images) yang dapat digunakan kembali, mempercepat proses pengembangan dan implementasi. Dengan adanya Docker Hub, penyedia gambar kontainer, kolaborasi antar pengembang juga menjadi lebih mudah.

Secara keseluruhan, konsep containerization dengan Docker memberikan fleksibilitas, kecepatan, dan konsistensi dalam siklus hidup aplikasi, membuatnya menjadi alat yang sangat berharga dalam dunia pengembangan perangkat lunak

4. Redundansi dan ketersediaan tinggi adalah krusial dalam lingkungan cloud computing karena meminimalkan risiko kegagalan sistem dan memastikan kontinuitas layanan. Dengan adanya redundansi, jika satu komponen atau server mengalami masalah, beban kerja dapat dipindahkan secara otomatis ke sumber daya yang lain, menjaga ketersediaan layanan.
   Tingkat ketersediaan yang tinggi sangat penting untuk menghindari downtime, yang dapat berdampak negatif pada produktivitas, reputasi bisnis, dan kepercayaan pengguna. Dalam cloud computing, di mana aplikasi dan data disimpan di infrastruktur yang dapat diakses secara online, aspek ketersediaan sangat diperlukan.

   Meskipun demikian, mencapai tingkat ketersediaan yang diinginkan tidak selalu mudah. Beberapa tantangan melibatkan kompleksitas manajemen sumber daya, kebutuhan untuk mengatasi kegagalan perangkat keras atau lunak dengan cepat, dan perencanaan yang cermat untuk memastikan redundansi yang efektif. Pengelolaan dan pemantauan sistem secara proaktif menjadi kunci dalam mengatasi tantangan ini.

   5. Pertimbangan keamanan antara cloud public dan private melibatkan trade-off antara kontrol langsung dan manfaat skalabilitas serta efisiensi yang ditawarkan oleh setiap jenis cloud.

   Cloud private memberikan tingkat kontrol yang lebih besar karena infrastruktur dielola secara eksklusif oleh organisasi tersebut. Namun, keamanan ini juga bergantung pada kemampuan internal organisasi dalam mengelola dan mempertahankan lingkungan cloud sendiri.

   Di sisi lain, cloud public menyediakan akses ke infrastruktur yang dikelola oleh penyedia layanan cloud, mengurangi tanggung jawab langsung organisasi atas infrastruktur fisik. Penyedia cloud public biasanya menawarkan lapisan keamanan tinggi, tetapi kekhawatiran umum melibatkan masalah privasi dan keamanan data karena data berada di tangan pihak ketiga.

   Organisasi seharusnya memilih cloud private ketika mereka memiliki kebutuhan untuk mengontrol penuh atas lingkungan cloud, memiliki persyaratan keamanan yang sangat tinggi, atau tunduk pada regulasi ketat yang memerlukan kontrol lebih besar terhadap data. Sementara itu, cloud public cocok untuk organisasi yang membutuhkan skalabilitas, efisiensi biaya, dan fleksibilitas sumber daya tanpa harus mengelola infrastruktur fisik secara langsung.
   Banyak organisasi juga memilih pendekatan campuran (hybrid) di mana mereka memanfaatkan kedua model cloud, menggunakan cloud private untuk beban kerja yang lebih sensitif dan cloud public untuk kebutuhan yang lebih dinamis dan scalable. Keputusan ini sering didasarkan pada kebutuhan spesifik dan strategi bisnis organisasi.

6. Virtualisasi dan containerization adalah dua pendekatan yang berbeda dalam mengisolasi dan menyediakan lingkungan untuk aplikasi.
Virtualisasi melibatkan penggunaan hypervisor untuk membuat mesin virtual yang dapat menjalankan sistem operasi lengkap secara terisolasi di atas sistem fisik. Setiap mesin virtual memiliki kernel dan menggunakan sumber daya fisik yang terpisah.
Containerization, khususnya dengan Docker, menggunakan konsep kontainer yang berbagi kernel host dan mengisolasi proses dan resource aplikasi. Kontainer lebih ringan dan cepat dibandingkan mesin virtual karena tidak perlu menggulirkan sistem operasi penuh.

Kelebihan Docker dan kontainer termasuk efisiensi sumber daya yang tinggi, waktu startup yang cepat, dan portabilitas aplikasi antar lingkungan. Docker memfasilitasi pengepakan dan distribusi aplikasi bersama dengan dependensinya, memastikan konsistensi antar lingkungan.
Meskipun kedua pendekatan ini memiliki peran mereka, kelebihan Docker dan kontainer meliputi fleksibilitas, kemudahan pengelolaan, dan kemampuan untuk melakukan penskalaan aplikasi dengan cepat. Docker juga menyediakan ekosistem yang kaya dengan Docker Hub, memungkinkan pengguna untuk berbagi dan menggunakan gambar kontainer yang telah disiapkan sebelumnya.
Pilihan antara virtualisasi dan containerization tergantung pada kebutuhan spesifik proyek dan lingkungan pengembangan, tetapi Docker dan kontainer secara luas diadopsi dalam dunia pengembangan modern karena manfaat efisiensinya.

7. Konsep skalabilitas horizontal dalam arsitektur cloud adalah pendekatan yang memungkinkan sistem untuk menangani lonjakan lalu lintas dengan menambahkan lebih banyak instans secara horizontal. Hal ini memberikan manfaat signifikan dalam hal kemampuan sistem untuk bersifat elastis dan responsif terhadap perubahan beban kerja.

Keuntungan utama dari skalabilitas horizontal termasuk:

- Responsif terhadap Lonjakan Lalu Lintas: Dengan menambahkan lebih banyak instans, sistem dapat menangani lonjakan lalu lintas dengan lebih baik, menjaga ketersediaan layanan dan kinerja tanpa mengorbankan responsivitas.

- Efisiensi Sumber Daya: Instans dapat ditambahkan atau dikurangi sesuai kebutuhan, memungkinkan penggunaan sumber daya yang lebih efisien. Ini berkontribusi pada optimalisasi biaya dan meningkatkan efisiensi infrastruktur.

- Skalabilitas Otomatis: Skalabilitas horizontal dapat diotomatisasi, memungkinkan sistem untuk menyesuaikan jumlah instans berdasarkan aturan atau kebijakan tertentu. Ini mengurangi beban administratif dan meningkatkan kecepatan respons terhadap perubahan beban kerja.

- Peningkatan Ketersediaan: Dengan mendistribusikan beban kerja di antara beberapa instans, sistem dapat meningkatkan ketersediaan secara keseluruhan. Jika satu instans mengalami masalah, yang lain masih dapat menjalankan layanan.

Namun, penting untuk memperhitungkan desain dan konfigurasi yang tepat agar sistem dapat secara efektif menanggapi perubahan dinamis dalam beban kerja. Selain itu, pemantauan dan

manajemen yang cermat diperlukan untuk memastikan bahwa penambahan instans berjalan lancar dan sesuai dengan kebutuhan aplikasi.

8. Perbandingan antara Software as a Service (SaaS) dan Function as a Service (FaaS) mencerminkan perbedaan dalam jenis layanan yang mereka tawarkan.

SaaS (Software as a Service):

Memberikan aplikasi lengkap yang dapat diakses secara online.
Biasanya dirancang untuk memenuhi kebutuhan umum dan spesifik suatu fungsi.
Pengguna tidak perlu mengelola infrastruktur atau perangkat lunak, karena semuanya di-host dan dikelola oleh penyedia SaaS.

FaaS (Function as a Service):
Fokus pada eksekusi fungsi atau potongan kode yang berjalan secara independen.
Terutama digunakan untuk menangani fungsi kecil dan khusus yang diaktifkan oleh suatu kejadian tertentu.

Penyedia cloud secara otomatis menangani alokasi sumber daya dan skala fungsi.
Kapan Menggunakan FaaS atau SaaS:

Pilih SaaS jika:
Anda membutuhkan solusi aplikasi end-to-end.
Kebutuhan bisnis Anda umum dan dapat dilayani oleh solusi yang sudah ada.
Pengelolaan dan pemeliharaan infrastruktur tidak menjadi fokus utama.

Pilih FaaS jika:
Anda memiliki fungsi-fungsi kecil atau tugas yang bersifat event-driven.
Skalabilitas otomatis dan efisiensi biaya per eksekusi fungsi penting.
Ingin menghindari biaya tetap dan hanya membayar untuk eksekusi fungsi yang sebenarnya.
Keputusan antara FaaS dan SaaS tergantung pada karakteristik aplikasi dan kebutuhan bisnis spesifik. Terkadang, perusahaan menggunakan kombinasi dari kedua model ini, mengintegrasikan FaaS untuk menangani fungsi-fungsi spesifik dalam arsitektur SaaS mereka.

9. Docker Hub memfasilitasi manajemen kontainer dengan menyediakan repositori publik yang memungkinkan pengguna untuk menyimpan, berbagi, dan mengelola gambar kontainer. Beberapa keuntungan penggunaan Docker Hub dan repositori publik ini meliputi:

Akses ke Gambar Kontainer Publik: Docker Hub menyediakan akses ke repositori yang berisi berbagai gambar kontainer publik. Pengguna dapat dengan mudah mencari dan menggunakan gambar-gambar ini untuk mempercepat pengembangan aplikasi tanpa perlu membuat ulang gambar yang umum.

Kemudahan Berbagi dan Kolaborasi: Docker Hub memungkinkan pengguna untuk dengan mudah berbagi gambar kontainer mereka dengan anggota tim atau komunitas. Ini mendukung kolaborasi dan memungkinkan pengguna untuk mendapatkan manfaat dari kontribusi dari orang lain.

Integrasi dengan Alat Pengembangan: Docker Hub dapat diintegrasikan dengan alat pengembangan dan otomatisasi, seperti Docker Compose, Jenkins, atau Kubernetes. Ini mempermudah otomatisasi proses pengembangan, pengujian, dan penyebaran aplikasi.

Skalabilitas dan Ketersediaan Tinggi: Docker Hub memiliki infrastruktur yang dapat diandalkan, memberikan ketersediaan tinggi dan skalabilitas. Gambar-gambar kontainer dapat diakses dengan cepat dan dapat diandalkan untuk digunakan dalam berbagai skenario.

Repositori Privat Opsional: Docker Hub juga menyediakan repositori privat sebagai opsi berbayar, memungkinkan organisasi untuk menyimpan gambar-gambar kontainer secara pribadi dan aman.
Namun, penting untuk mempertimbangkan kebijakan keamanan dan privasi saat menggunakan repositori publik. Beberapa organisasi memilih menggunakan repositori privat untuk mengontrol akses dan keamanan gambar kontainer mereka, terutama jika aplikasi mereka membutuhkan lapisan keamanan yang lebih tinggi.

10. Untuk meningkatkan keamanan data dalam cloud computing, organisasi dapat mengambil beberapa langkah konkret:

Enkripsi Data:

Terapkan enkripsi untuk data selama penyimpanan, pemindahan, dan pengolahan.
Gunakan enkripsi end-to-end untuk melindungi data bahkan saat berada dalam transit.
Manajemen Akses yang Tepat:

Atur kebijakan manajemen akses yang ketat.
Berikan hak akses sesuai dengan prinsip least privilege untuk meminimalkan risiko akses yang tidak sah.

- Authentication dan Authorization:
- Monitoring dan Logging:
- Kemanan Infrastruktur Cloud:
- Patch dan Update Secara Teratur:
- Pengelolaan Identitas dan Akses (IAM):
- Penyimpanan Data yang Aman:
- Kebijakan Keamanan yang Jelas:
- Audit dan Penilaian Keamanan Reguler:

Dengan mengintegrasikan langkah-langkah ini, organisasi dapat meningkatkan keamanan data mereka di lingkungan cloud. Penting untuk memahami persyaratan keamanan spesifik pada platform cloud yang digunakan dan memastikan kepatuhan dengan standar keamanan industri.