

**Name : Jenibelle Wieven**  
**NIM : 2602105632**  
**Major : Data Science**  
**Class : LC09**  
**Subject : Big Data Infrastructure and Technology**

**1. Pelajari dan jelaskan cara pemilihan region dalam pembangunan cloud architecture, dan berikan contoh untuk tiap poin.**

*Proximity to Users :*

Memilih region yang dekat dengan mayoritas pengguna meningkatkan kecepatan akses aplikasi.

Contoh: Jika mayoritas pengguna berada di AS, region di Virginia atau California dapat meningkatkan kinerja.

*Compliance and Data Governance :*

Kepatuhan regulasi data mempengaruhi pilihan region.

Contoh: Untuk mematuhi GDPR, aplikasi yang menangani data Eropa memerlukan region di Eropa.

*Resilience and Disaster Recovery:*

Memilih region di zona geografis yang berbeda meningkatkan ketersediaan dan pemulihan bencana.

Contoh: Memilih region di Virginia dan Ohio memberikan redundancy dan pemulihan yang lebih baik.

*Cost :*

Biaya infrastruktur cloud bervariasi antar region.

Contoh: Region Asia-Pasifik mungkin lebih mahal daripada AS, sehingga perlu mempertimbangkan biaya.

*Service Availability :*

Memastikan semua layanan diperlukan tersedia di region yang dipilih.

Contoh: Layanan keamanan khusus mungkin hanya tersedia di beberapa region.

**2. Jelaskan cara pemilihan AMI EC2 secara detail.**

*Identifikasi Kebutuhan Aplikasi :*

Langkah awal adalah mengidentifikasi kebutuhan aplikasi atau tugas kerja yang akan dijalankan pada instance EC2. Ini mencakup menentukan sistem operasi yang dibutuhkan (contohnya, Linux atau Windows), perangkat lunak tambahan

yang diperlukan, konfigurasi jaringan, serta spesifikasi sumber daya seperti kapasitas RAM, jumlah CPU, dan ukuran penyimpanan.

*Telusuri dan Pilih AMI :*

Login ke AWS Management Console dan akses layanan EC2. Di halaman dashboard EC2, cari menu "Instances" dan klik opsi "Launch Instance". Selanjutnya, Anda akan diminta untuk memilih Amazon Machine Image (AMI). AWS menawarkan beragam pilihan AMI yang meliputi yang disediakan oleh AWS, mitra, atau yang telah Anda buat sebelumnya.

*Filter dan Cari AMI yang Sesuai :*

Manfaatkan fitur filter untuk memfilter AMI yang cocok dengan kebutuhan Anda. Anda dapat melakukan penyaringan berdasarkan sistem operasi (seperti Amazon Linux, Ubuntu, Windows), arsitektur (64-bit atau 32-bit), fitur-fitur khusus (contohnya, AMI yang sudah dilengkapi dengan perangkat lunak tertentu seperti database, server web), dan kriteria lainnya.

*Periksa Detail AMI :*

Setelah menemukan AMI yang potensial, pastikan untuk memeriksa detailnya dengan teliti. Tinjau deskripsi AMI, termasuk versi sistem operasi, tanggal rilis, perawatan, dan informasi lain yang relevan. Pastikan AMI ini memenuhi kebutuhan Anda dan sesuai dengan standar keamanan serta keandalan yang diperlukan.

*Ulas Kebijakan Harga :*

Pastikan untuk memperhatikan biaya yang terkait dengan penggunaan AMI. Beberapa AMI mungkin memerlukan biaya tambahan berdasarkan lisensi perangkat lunak yang terpasang di dalamnya. Pastikan Anda memahami dan mempertimbangkan biaya ini dalam anggaran Anda.

*Pilih AMI dan Lanjutkan :*

Setelah Anda yakin dengan pilihan AMI Anda, pilih AMI tersebut dan lanjutkan dengan konfigurasi instance EC2 Anda. Di langkah berikutnya, Anda akan diminta untuk memilih jenis instance, konfigurasi jaringan, penyimpanan, dan opsi lainnya sebelum akhirnya meluncurkan instance EC2 menggunakan AMI yang telah dipilih.

**3. Pelajari dan tuliskan secara detail mengenai konsep *instance type* dari EC2 beserta *usecase* untuk tiap tipe.**

*General Purpose Instances :*

Menyediakan performa yang seimbang dengan rasio CPU dan memori yang baik untuk berbagai beban kerja umum.

Use Case: Cocok untuk aplikasi web, server back-end, basis data relasional, dan aplikasi enterprise yang memerlukan performa yang seimbang antara CPU dan memori.

#### *Compute Optimized Instances :*

Menawarkan performa komputasi yang tinggi dengan CPU yang kuat dan jumlah memori yang moderate.

Use Case: Ideal untuk beban kerja yang membutuhkan pemrosesan komputasi intensif seperti analisis data, pengolahan batch, rendering, dan simulasi ilmiah.

#### *Memory Optimized Instances :*

Dirancang untuk aplikasi yang memerlukan banyak memori, tipe ini menawarkan rasio tinggi antara CPU dan memori.

Use Case: Cocok untuk basis data in-memory, analisis big data, caching, dan aplikasi yang memerlukan banyak memori.

#### *Storage Optimized Instances :*

Tipe ini dioptimalkan untuk aplikasi yang memerlukan kapasitas penyimpanan yang besar dan performa tinggi dari SSD atau NVMe.

Use Case: Cocok untuk basis data NoSQL, sistem file distribusi, aplikasi analitik, dan beban kerja yang membutuhkan akses cepat ke data yang disimpan.

#### *Accelerated Computing Instances :*

Menawarkan akselerasi komputasi dengan menggunakan GPU atau FPGA.

Use Case: Cocok untuk aplikasi yang memerlukan pemrosesan paralel seperti rendering grafis, pembelajaran mesin, pemrosesan gambar dan video, serta simulasi ilmiah yang intensif.

#### *Burstable Performance Instances :*

Menyediakan akses burstable ke sumber daya CPU untuk aplikasi yang memerlukan performa CPU sesekali.

Use Case: Cocok untuk beban kerja yang memiliki aktivitas CPU yang tidak konsisten seperti aplikasi pengembangan dan uji coba, situs web dengan beban lalu lintas yang tidak teratur, dan aplikasi kecil yang memerlukan performa CPU sesekali.