

Panduan Ujian AWS Certified Solutions Architect - Associate (SAA-C03)

Pengantar

Ujian AWS Certified Solutions Architect - Associate (SAA-C03) ditujukan untuk individu yang memegang peran *solutions architect*. Ujian memvalidasi kemampuan kandidat untuk merancang solusi berdasarkan Kerangka Kerja AWS Well-Architected.

Ujian memvalidasi kemampuan kandidat untuk menyelesaikan tugas-tugas berikut:

- Merancang yang menggabungkan layanan AWS untuk memenuhi persyaratan bisnis saat ini dan kebutuhan yang diproyeksikan di masa depan
- Merancang arsitektur yang aman, tangguh, berkinerja tinggi, dan dioptimalkan biaya
- Meninjau solusi yang ada dan tentukan perbaikan

Deskripsi kandidat target

Kandidat target harus memiliki setidaknya 1 tahun pengalaman langsung merancang solusi *cloud* yang menggunakan layanan AWS.

Lihat Lampiran untuk daftar teknologi dan konsep yang mungkin muncul pada ujian, daftar layanan dan fitur AWS dalam lingkup, dan daftar layanan dan fitur AWS di luar cakupan.

Konten ujian

Tipe jawaban

Ada dua jenis pertanyaan dalam ujian:

- **Pilihan berganda:** Memiliki satu jawaban benar dan tiga jawaban salah (distraktor)
- **Jawaban berganda:** Memiliki dua atau lebih jawaban yang benar dari lima atau lebih pilihan jawaban

Pilih satu atau lebih jawaban yang paling melengkapi pernyataan atau menjawab pertanyaan. Distraktor, atau jawaban yang salah, adalah pilihan jawaban yang mungkin akan dipilih oleh kandidat yang memiliki pengetahuan atau keterampilan yang tidak memadai. Distraktor umumnya merupakan jawaban yang masuk akal dan cocok dengan pokok bahasan.

Pertanyaan yang tidak dijawab dinilai sebagai salah, tidak ada sanksi untuk terkaan. Ujian mencakup 50 pertanyaan yang akan memengaruhi nilai Anda.

Konten tanpa nilai

Ujian ini mencakup 15 pertanyaan tanpa nilai yang tidak memengaruhi nilai Anda. AWS mengumpulkan informasi tentang kinerja pada pertanyaan-pertanyaan yang tidak dinilai ini untuk mengevaluasi pertanyaan-pertanyaan ini untuk digunakan kelak sebagai pertanyaan yang dinilai. Pertanyaan-pertanyaan yang tidak dinilai ini tidak teridentifikasi pada saat ujian.

Hasil ujian

Ujian AWS Certified Solutions Architect - Associate (SAA-C03) memiliki penunjukan lulus atau gagal. Ujian ini dinilai dengan patokan standar minimum yang ditetapkan oleh profesional AWS yang mengikuti praktik terbaik dan pedoman industri sertifikasi.

Hasil Anda dalam ujian dilaporkan dalam skala nilai 100-1.000. Nilai kelulusan minimum adalah 720. Nilai Anda menunjukkan kinerja Anda secara keseluruhan dalam ujian dan apakah Anda lulus. Model penilaian skala membantu menyamakan nilai di beberapa formulir ujian yang mungkin memiliki tingkat kesulitan yang sedikit berbeda.

Laporan nilai Anda dapat berisi tabel klasifikasi kinerja Anda di setiap tingkat bagian. Ujian menggunakan model penilaian kompensasi, yang berarti Anda tidak perlu mencapai nilai kelulusan di setiap bagian. Anda hanya harus lulus dalam ujian secara keseluruhan.

Setiap bagian ujian memiliki bobot tertentu, sehingga beberapa bagian memiliki lebih banyak soal daripada bagian yang lain. Tabel klasifikasi berisi informasi umum yang menyoroti kekuatan dan kelemahan Anda. Berhati-hatilah saat Anda menafsirkan umpan balik tingkat bagian.

Garis besar konten

Panduan ujian CLF-C02 ini mencakup pembobotan, domain konten, dan pernyataan tugas untuk ujian. Panduan ini tidak memberikan daftar lengkap dari konten ujian. Namun, tersedia konteks tambahan untuk masing-masing pernyataan tugas untuk membantu Anda mempersiapkan ujian.

Ujian ini memiliki domain konten dan bobot sebagai berikut:

- Domain 1: Rancang Arsitektur Aman (30% dari konten yang diberi nilai)
- Domain 2: Rancang Arsitektur Tangguh (26% dari konten yang diberi nilai)
- Domain 3: Rancang Arsitektur Berkinerja Tinggi (24% dari konten yang diberi nilai)
- Domain 4: Rancang Arsitektur dengan Biaya Optimal (20% dari konten yang diberi nilai)

Domain 1: Rancang Arsitektur Aman

Pernyataan Tugas 1.1: Rancang akses aman ke sumber daya AWS.

Pengetahuan tentang:

- Kontrol akses dan manajemen di beberapa akun
- Layanan akses dan identitas gabungan AWS (misalnya, AWS Identity and Access Management [IAM], AWS IAM Identity Center [AWS Single Sign-On])
- Infrastruktur global AWS (misalnya, Regional AWS, Availability Zone)
- Praktik terbaik keamanan AWS (misalnya, prinsip hak istimewa paling sedikit)
- Model tanggung jawab bersama AWS

Keterampilan dalam:

- Menerapkan praktik terbaik keamanan AWS untuk pengguna IAM dan pengguna *root* (misalnya, autentikasi multi-faktor [MFA])
- Merancang model otorisasi fleksibel yang mencakup pengguna, grup, *role*, dan kebijakan IAM
- Merancang strategi kontrol akses berbasis peran (misalnya, AWS Security Token Service [AWS STS], peralihan *role*, akses lintas akun)
- Merancang strategi keamanan untuk beberapa akun AWS (misalnya, AWS Control Tower, kebijakan kontrol layanan [SCP])

- Menentukan penggunaan kebijakan sumber daya yang tepat untuk layanan AWS
- Menentukan kapan harus menggabungkan layanan direktori dengan IAM role

Pernyataan Tugas 1.2: Rancang *workload* dan aplikasi yang aman.

Pengetahuan tentang:

- Konfigurasi aplikasi dan keamanan kredensial
- *Endpoint* layanan AWS
- Port kontrol, protokol, dan lalu lintas jaringan di AWS
- Akses aplikasi yang aman
- Layanan keamanan dengan kasus penggunaan yang sesuai (misalnya, Amazon Cognito, Amazon GuardDuty, Amazon Macie)
- Vektor ancaman eksternal ke AWS (misalnya, DDoS, injeksi SQL)

Keterampilan dalam:

- Merancang arsitektur VPC dengan komponen keamanan (misalnya, *security group*, tabel rute, ACL jaringan, *gateway* NAT)
- Menentukan strategi segmentasi jaringan (misalnya, menggunakan subnet publik dan subnet privat)
- Mengintegrasikan layanan AWS untuk mengamankan aplikasi (misalnya, AWS Shield, AWS WAF, IAM Identity Center, AWS Secrets Manager)
- Mengamankan koneksi jaringan eksternal ke dan dari AWS Cloud (misalnya, VPN, AWS Direct Connect)

Pernyataan Tugas 1.3: Tentukan kontrol keamanan data yang sesuai.

Pengetahuan tentang:

- Akses dan tata kelola data
- Pemulihan data
- Retensi dan klasifikasi data
- Enkripsi dan manajemen kunci yang sesuai

Keterampilan dalam:

- Menyelaraskan teknologi AWS untuk memenuhi persyaratan kepatuhan
- Mengenkripsi data saat istirahat (misalnya, AWS Key Management Service [AWS KMS])
- Mengenkripsi data dalam perjalanan (misalnya, AWS Certificate Manager [ACM] menggunakan TLS)
- Menerapkan kebijakan akses untuk kunci enkripsi
- Menerapkan cadangan data dan replikasi
- Menerapkan kebijakan untuk akses data, siklus hidup, dan perlindungan
- Memutar kunci enkripsi dan memperbarui sertifikat

Domain 2: Rancang Arsitektur Tangguh

Pernyataan Tugas 2.1: Rancang arsitektur yang dapat diskalakan dan *loosely coupled*.

Pengetahuan tentang:

- Pembuatan dan manajemen API (misalnya, Amazon API Gateway, REST API)
- AWS Managed Services dengan kasus penggunaan yang sesuai (misalnya, AWS Transfer Family, Amazon Simple Queue Service [Amazon SQS], Secrets Manager)
- Strategi *Caching*
- Prinsip-prinsip desain untuk layanan mikro (misalnya, *workload stateless* dibandingkan dengan *workload stateful*)
- Arsitektur berbasis *event*
- Penskalaan vertikal dan penskalaan horizontal
- Cara menggunakan akselerator edge dengan tepat (misalnya, jaringan pengiriman konten [CDN])
- Cara memigrasikan aplikasi ke dalam kontainer
- Konsep *load balancing* (misalnya, Application Load Balancer)
- Arsitektur multi-tingkat
- Konsep antrean dan pesan (misalnya, terbitkan/berlangganan)
- Teknologi dan pola nirserver (misalnya, AWS Fargate, AWS Lambda)
- Jenis penyimpanan dengan karakteristik terkait (misalnya, objek, file, blok)
- Orkestrasi kontainer (misalnya, Amazon Elastic Container Service [Amazon ECS], Amazon Elastic Kubernetes Service [Amazon EKS])
- Kapan menggunakan replika baca

- Orkestrasi alur kerja (misalnya, AWS Step Functions)

Keterampilan dalam:

- Merancang arsitektur berbasis acara, layanan mikro, dan/atau multi-tingkat berdasarkan persyaratan
- Menentukan strategi penskalaan untuk komponen yang digunakan dalam rancangan arsitektur
- Menentukan layanan AWS yang diperlukan untuk mencapai *loose coupling* berdasarkan persyaratan
- Menentukan kapan harus menggunakan kontainer
- Menentukan kapan harus menggunakan teknologi dan pola nirserver
- Merekomendasikan komputasi, penyimpanan, jaringan, dan teknologi basis data yang sesuai berdasarkan persyaratan
- Menggunakan layanan AWS yang dibuat khusus untuk *workload*

Pernyataan Tugas 2.2: Rancang arsitektur yang berketersediaan tinggi dan/atau toleran terhadap kesalahan.

Pengetahuan tentang:

- Infrastruktur global AWS (misalnya, Availability Zone, Regional AWS, Amazon Route 53)
- AWS Managed Services dengan kasus penggunaan yang sesuai (misalnya, Amazon Comprehend, Amazon Polly)
- Konsep jaringan dasar (misalnya, tabel rute)
- Strategi pemulihan bencana (DR) (misalnya, pencadangan dan pemulihan, lampu pilot, siaga hangat, *failover* Aktif-Aktif, tujuan titik pemulihan [RPO], tujuan waktu pemulihan [RTO])
- Pola rancangan terdistribusi
- Strategi *failover*
- Infrastruktur *immutable*
- Konsep *load balancing* (misalnya, Application Load Balancer)
- Konsep *proxy* (misalnya, Amazon RDS Proxy)
- *Service quotas* dan *throttling* (misalnya, cara mengonfigurasi kuota layanan untuk workload di lingkungan siaga)
- Opsi dan karakteristik penyimpanan (misalnya, daya tahan, replikasi)
- Visibilitas *workload* (misalnya, AWS X-Ray)

Keterampilan dalam:

- Menentukan strategi otomatisasi untuk memastikan integritas infrastruktur
- Menentukan layanan AWS yang diperlukan untuk menyediakan arsitektur yang sangat tersedia dan/atau toleran terhadap kesalahan di seluruh Regional AWS atau Availability Zone
- Mengidentifikasi metrik berdasarkan kebutuhan bisnis untuk memberikan solusi yang sangat tersedia
- Menerapkan rancangan untuk mengurangi titik tunggal kegagalan
- Menerapkan strategi untuk memastikan daya tahan dan ketersediaan data (misalnya, cadangan)
- Memilih strategi DR yang tepat untuk memenuhi persyaratan bisnis
- Menggunakan layanan AWS yang meningkatkan keandalan aplikasi *legacy* dan aplikasi yang tidak dibangun untuk *cloud* (misalnya, ketika perubahan aplikasi tidak dimungkinkan)
- Menggunakan layanan AWS yang dibuat khusus untuk *workload*

Domain 3: Rancang Arsitektur Berkinerja Tinggi

Pernyataan Tugas 3.1: Tentukan solusi penyimpanan berkinerja tinggi dan/atau dapat diskalakan.

Pengetahuan tentang:

- Solusi penyimpanan hibrida untuk memenuhi kebutuhan bisnis
- Layanan penyimpanan dengan kasus penggunaan yang sesuai (misalnya, Amazon S3, Amazon Elastic File System [Amazon EFS], Amazon Elastic Block Store [Amazon EBS])
- Jenis penyimpanan dengan karakteristik terkait (misalnya, objek, file, blok)

Keterampilan dalam:

- Menentukan layanan penyimpanan dan konfigurasi yang memenuhi tuntutan kinerja
- Menentukan layanan penyimpanan yang dapat diskalakan untuk mengakomodasi kebutuhan masa depan

Pernyataan Tugas 3.2: Rancang solusi komputasi berkinerja tinggi dan elastis.

Pengetahuan tentang:

- Layanan komputasi AWS dengan kasus penggunaan yang sesuai (misalnya, AWS Batch, Amazon EMR, Fargate)
- Konsep komputasi terdistribusi yang didukung oleh infrastruktur global AWS dan layanan edge
- Konsep antrean dan pesan (misalnya, terbitkan/berlangganan)
- Kemampuan skalabilitas dengan kasus penggunaan yang sesuai (misalnya, Amazon EC2 Auto Scaling, AWS Auto Scaling)
- Teknologi dan pola nirserver (misalnya, Lambda, Fargate)
- Orkestrasi kontainer (misalnya, Amazon ECS, Amazon EKS)

Keterampilan dalam:

- *Decoupling workload* agar komponen dapat menskalakan secara independen
- Mengidentifikasi metrik dan kondisi untuk melakukan tindakan penskalaan
- Memilih opsi dan fitur komputasi yang sesuai (misalnya, tipe instans EC2) untuk memenuhi persyaratan bisnis
- Memilih jenis dan ukuran sumber daya yang sesuai (misalnya, jumlah memori Lambda) untuk memenuhi persyaratan bisnis

Pernyataan Tugas 3.3: Tentukan solusi basis data berkinerja tinggi.

Pengetahuan tentang:

- Infrastruktur global AWS (misalnya, Regional AWS, Availability Zone)
- Strategi dan layanan *caching* (misalnya, Amazon ElastiCache)
- Pola akses data (misalnya, intensif-baca dibandingkan dengan intensif-tulis)
- Perencanaan kapasitas basis data (misalnya, unit kapasitas, tipe instans, Provisioned IOPS)
- Koneksi basis data dan *proxy*
- Mesin basis data dengan kasus penggunaan yang sesuai (misalnya, migrasi heterogen, migrasi homogen)
- Replikasi basis data (misalnya, replika baca)
- Jenis dan layanan basis data (misalnya, nirserver, relasional dibandingkan dengan non-relasional, *in-memory*)

Keterampilan dalam:

- Mengonfigurasi replika baca untuk memenuhi persyaratan bisnis
- Merancang arsitektur basis data
- Menentukan mesin basis data yang sesuai (misalnya, MySQL dibandingkan dengan PostgreSQL)
- Menentukan jenis basis data yang sesuai (misalnya, Amazon Aurora, Amazon DynamoDB)
- Mengintegrasikan *caching* untuk memenuhi persyaratan bisnis

Pernyataan Tugas 3.4: Tentukan arsitektur jaringan berkinerja tinggi dan/atau dapat diskalakan.

Pengetahuan tentang:

- Layanan jaringan *edge* dengan kasus penggunaan yang sesuai (misalnya, Amazon CloudFront, AWS Global Accelerator)
- Bagaimana merancang arsitektur jaringan (misalnya, tingkat subnet, perutean, pembuatan alamat IP)
- Konsep *load balancing* (misalnya, Application Load Balancer)
- Opsi koneksi jaringan (misalnya, AWS VPN, Direct Connect, AWS PrivateLink)

Keterampilan dalam:

- Membuat topologi jaringan untuk berbagai arsitektur (misalnya, global, hibrida, multi-tingkat)
- Menentukan konfigurasi jaringan yang dapat menskalakan untuk mengakomodasi kebutuhan masa depan
- Menentukan penempatan sumber daya yang tepat untuk memenuhi persyaratan bisnis
- Memilih strategi *load balancing* yang tepat

Pernyataan Tugas 3.5: Tentukan solusi *ingestion* data dan transformasi berkinerja tinggi.

Pengetahuan tentang:

- Layanan analitik dan visualisasi data dengan kasus penggunaan yang sesuai (misalnya, Amazon Athena, AWS Lake Formation, Amazon QuickSight)
- Pola *ingestion* data (misalnya, frekuensi)

- Layanan transfer data dengan kasus penggunaan yang sesuai (misalnya, AWS DataSync, Gateway AWS Storage)
- Layanan transformasi data dengan kasus penggunaan yang sesuai (misalnya, AWS Glue)
- Akses aman ke titik akses *ingestion*
- Ukuran dan kecepatan yang dibutuhkan untuk memenuhi kebutuhan bisnis
- Layanan *streaming* data dengan kasus penggunaan yang sesuai (misalnya, Amazon Kinesis)

Keterampilan dalam:

- Membangun dan mengamankan danau data
- Merancang arsitektur *streaming* data
- Merancang solusi transfer data
- Menerapkan strategi visualisasi
- Memilih opsi komputasi yang sesuai untuk pemrosesan data (misalnya, Amazon EMR)
- Memilih konfigurasi yang sesuai untuk *ingestion*
- Mengubah data antar-format (misalnya, .csv ke .parquet)

Domain 4: Merancang Arsitektur dengan Biaya Optimal

Pernyataan Tugas 4.1: Rancang solusi penyimpanan dengan biaya optimal.

Pengetahuan tentang:

- Opsi akses (misalnya, *Bucket S3* dengan penyimpanan objek Pemohon Membayar)
- Fitur layanan manajemen biaya AWS (misalnya, tanda alokasi biaya, penagihan multi-akun)
- Alat manajemen biaya AWS dengan kasus penggunaan yang sesuai (misalnya, AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Layanan penyimpanan AWS dengan kasus penggunaan yang sesuai (misalnya, Amazon FSx, Amazon EFS, Amazon S3, Amazon EBS)
- Strategi cadangan
- Opsi penyimpanan blok (misalnya, jenis volume *hard disk drive* [HDD], tipe volume *solid state drive* [SSD])
- Siklus hidup data

- Opsi penyimpanan hibrida (misalnya, DataSync, Transfer Family, Storage Gateway)
- Pola akses penyimpanan
- Tingkatan penyimpanan (misalnya, tingkatan dingin untuk penyimpanan objek)
- Jenis penyimpanan dengan karakteristik terkait (misalnya, objek, file, blok)

Keterampilan dalam:

- Merancang strategi penyimpanan yang sesuai (misalnya, unggahan *batch* ke Amazon S3 dibandingkan dengan unggahan individual)
- Menentukan ukuran penyimpanan yang benar untuk *workload*
- Menentukan metode biaya terendah untuk mentransfer data untuk *workload* ke penyimpanan AWS
- Menentukan kapan penskalaan otomatis penyimpanan diperlukan
- Mengelola siklus hidup objek S3
- Memilih solusi cadangan dan/atau pengarsipan yang sesuai
- Memilih layanan yang sesuai untuk migrasi data ke layanan penyimpanan
- Memilih tingkat penyimpanan yang sesuai
- Memilih siklus hidup data yang benar untuk penyimpanan
- Memilih layanan penyimpanan yang paling hemat biaya untuk *workload*

Pernyataan Tugas 4.2: Rancang solusi komputasi dengan biaya optimal.

Pengetahuan tentang:

- Fitur layanan manajemen biaya AWS (misalnya, tanda alokasi biaya, penagihan multi-akun)
- Alat manajemen biaya AWS dengan kasus penggunaan yang sesuai (misalnya, Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Infrastruktur global AWS (misalnya, Regional AWS, Availability Zone)
- Opsi pembelian AWS (misalnya, Instans Spot, Instans Cadangan, Savings Plans)
- Strategi komputasi terdistribusi (misalnya, pemrosesan *edge*)
- Opsi komputasi hibrida (misalnya, AWS Outposts, AWS Snowball Edge)
- Tipe instans, *family*, dan ukuran (misalnya, memori dioptimalkan, komputasi dioptimalkan, virtualisasi)

- Optimalisasi pemanfaatan komputasi (misalnya, kontainer, komputasi nirserver, layanan mikro)
- Strategi penskalaan (misalnya, penskalaan otomatis, hibernasi)

Keterampilan dalam:

- Menentukan strategi *load balancing* yang tepat (misalnya, Application Load Balancer [Lapisan 7] dibandingkan dengan Network Load Balancer [Lapisan 4] dibandingkan dengan Gateway Load Balancer)
- Menentukan metode dan strategi penskalaan yang tepat untuk *workload* elastis (misalnya, horizontal dibandingkan dengan vertikal, hibernasi EC2)
- Menentukan layanan komputasi AWS yang hemat biaya dengan kasus penggunaan yang sesuai (misalnya, Lambda, Amazon EC2, Fargate)
- Menentukan ketersediaan yang diperlukan untuk berbagai kelas *workload* (misalnya, *workload* produksi, *workload* non-produksi)
- Memilih *family* instans yang sesuai untuk *workload*
- Memilih ukuran instans yang sesuai untuk *workload*

Pernyataan Tugas 4.3: Rancang solusi basis data dengan biaya optimal.

Pengetahuan tentang:

- Fitur layanan manajemen biaya AWS (misalnya, tanda alokasi biaya, penagihan multi-akun)
- Alat manajemen biaya AWS dengan kasus penggunaan yang sesuai (misalnya, Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Strategi *Caching*
- Kebijakan retensi data
- Perencanaan kapasitas basis data (misalnya, unit kapasitas)
- Koneksi basis data dan *proxy*
- Mesin basis data dengan kasus penggunaan yang sesuai (misalnya, migrasi heterogen, migrasi homogen)
- Replikasi basis data (misalnya, replika baca)
- Jenis dan layanan basis data (misalnya, relasional dibandingkan dengan non-relasional, Aurora, DynamoDB)

Keterampilan dalam:

- Merancang kebijakan pencadangan dan retensi yang sesuai (misalnya, frekuensi *snapshot*)
- Menentukan mesin basis data yang sesuai (misalnya, MySQL dibandingkan dengan PostgreSQL)
- Menentukan layanan basis data AWS yang hemat biaya dengan kasus penggunaan yang sesuai (misalnya, DynamoDB dibandingkan dengan Amazon RDS, nirserver)
- Menentukan jenis basis data AWS yang hemat biaya (misalnya, format deret waktu, format kolom)
- Migrasi skema basis data dan data ke lokasi yang berbeda dan/atau mesin basis data yang berbeda

Pernyataan Tugas 4.4: Rancang arsitektur jaringan yang berbiaya optimal.

Pengetahuan tentang:

- Fitur layanan manajemen biaya AWS (misalnya, tanda alokasi biaya, penagihan multi-akun)
- Alat manajemen biaya AWS dengan kasus penggunaan yang sesuai (misalnya, Cost Explorer, AWS Budgets, AWS Cost and Usage Report)
- Konsep *load balancing* (misalnya, Application Load Balancer)
- *Gateway* NAT (misalnya, biaya instans NAT dibandingkan dengan biaya *gateway* NAT)
- Konektivitas jaringan (misalnya, jalur pribadi, jalur khusus, VPN)
- Perutean jaringan, topologi, dan *peering* (misalnya, AWS Transit Gateway, *peering* VPC)
- Layanan jaringan dengan kasus penggunaan yang sesuai (misalnya, DNS)

Keterampilan dalam:

- Mengonfigurasi jenis *gateway* NAT yang sesuai untuk jaringan (misalnya, satu *gateway* NAT bersama dibandingkan dengan *gateway* NAT untuk setiap Availability Zone)
- Mengonfigurasi koneksi jaringan yang sesuai (misalnya, Direct Connect dibandingkan dengan VPN dibandingkan dengan internet)

- Mengonfigurasi rute jaringan yang sesuai untuk meminimalkan biaya transfer jaringan (misalnya, Wilayah ke Wilayah, Availability Zone ke Availability Zone, privat ke publik, Global Accelerator, VPC *endpoint*)
- Menentukan kebutuhan strategis untuk jaringan pengiriman konten (CDN) dan *caching edge*
- Meninjau *workload* yang ada untuk pengoptimalan jaringan
- Memilih strategi *throttling* yang tepat
- Memilih alokasi lebar pita yang sesuai untuk perangkat jaringan (misalnya, satu VPN dibandingkan dengan beberapa VPN, kecepatan Direct Connect)

Lampiran

Teknologi dan konsep yang mungkin muncul pada ujian

Daftar berikut berisi teknologi dan konsep yang mungkin muncul pada ujian. Daftar ini tidak lengkap dan dapat berubah sewaktu-waktu. Urutan dan penempatan item dalam daftar ini tidak menunjukkan bobot atau nilai pentingnya dalam ujian:

- Komputasi
- Manajemen biaya
- Basis Data
- Pemulihan bencana
- Kinerja tinggi
- Management and Governance
- Layanan mikro dan pengiriman komponen
- Migrasi dan transfer data
- Jaringan, konektivitas, dan pengiriman konten
- Ketahanan
- Keamanan
- Prinsip rancangan nirserver dan berbasis peristiwa
- Penyimpanan

Layanan dan fitur dalam lingkup AWS

Daftar berikut berisi layanan dan fitur AWS yang berada dalam cakupan ujian. Daftar ini tidak lengkap dan dapat berubah sewaktu-waktu. Penawaran AWS muncul dalam kategori yang sejalan dengan fungsi utama penawaran:

Analitik:

- Amazon Athena
- AWS Data Exchange
- AWS Data Pipeline
- Amazon EMR
- AWS Glue
- Amazon Kinesis
- AWS Lake Formation
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)

- Amazon OpenSearch Service
- Amazon QuickSight
- Amazon Redshift

Integrasi Aplikasi:

- Amazon AppFlow
- AWS AppSync
- Amazon EventBridge
- Amazon MQ
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Queue Service (Amazon SQS)
- AWS Step Functions

Manajemen Biaya AWS:

- AWS Budgets
- AWS Cost and Usage Report
- AWS Cost Explorer
- Savings Plans

Komputasi:

- AWS Batch
- Amazon EC2
- Amazon EC2 Auto Scaling
- AWS Elastic Beanstalk
- AWS Outposts
- AWS Serverless Application Repository
- VMware Cloud on AWS
- AWS Wavelength

Kontainer:

- Amazon ECS Anywhere
- Amazon EKS Anywhere
- Amazon EKS Distro
- Amazon Elastic Container Registry (Amazon ECR)

- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Service (Amazon EKS)

Basis Data:

- Amazon Aurora
- Amazon Aurora Nirserver
- Amazon DocumentDB (dengan kompatibilitas MongoDB)
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Keyspaces (untuk Apache Cassandra)
- Amazon Neptune
- Amazon Quantum Ledger Database (Amazon QLDB)
- Amazon RDS
- Amazon Redshift

Alat Pengembang:

- AWS X-Ray

Web dan Seluler *Front-End*:

- AWS Amplify
- Amazon API Gateway
- AWS Device Farm
- Amazon Pinpoint

Machine Learning:

- Amazon Comprehend
- Amazon Forecast
- Amazon Fraud Detector
- Amazon Kendra
- Amazon Lex
- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker
- Amazon Textract

- Amazon Transcribe
- Amazon Translate

Management and Governance:

- AWS Auto Scaling
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS Command Line Interface (AWS CLI)
- AWS Compute Optimizer
- AWS Config
- AWS Control Tower
- Dasbor AWS Health
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Service untuk Prometheus
- Konsol Manajemen AWS
- AWS Organizations
- AWS Proton
- AWS Service Catalog
- AWS Systems Manager
- AWS Trusted Advisor
- Alat AWS Well-Architected

Layanan Media:

- Amazon Elastic Transcoder
- Amazon Kinesis Video Streams

Migrasi dan Transfer:

- AWS Application Discovery Service
- AWS Application Migration Service
- AWS Database Migration Service (AWS DMS)
- AWS DataSync
- AWS Migration Hub

- AWS Snow Family
- AWS Transfer Family

Jaringan dan Pengiriman Konten:

- VPN Klien AWS
- Amazon CloudFront
- AWS Direct Connect
- Elastic Load Balancing (ELB)
- AWS Global Accelerator
- AWS PrivateLink
- Amazon Route 53
- AWS Site-to-Site VPN
- AWS Transit Gateway
- Amazon VPC

Keamanan, Identitas, dan Kepatuhan:

- AWS Artifact
- AWS Audit Manager
- AWS Certificate Manager (ACM)
- AWS CloudHSM
- Amazon Cognito
- Amazon Detective
- AWS Directory Service
- AWS Firewall Manager
- Amazon GuardDuty
- Pusat Identitas IAM AWS (AWS Single Sign-On)
- AWS Identity and Access Management (AWS IAM)
- Amazon Inspector
- AWS Key Management Service (AWS KMS)
- Amazon Macie
- AWS Network Firewall
- AWS Resource Access Manager (AWS RAM)
- AWS Secrets Manager
- AWS Security Hub

- AWS Shield
- AWS WAF

Nirserver:

- AWS AppSync
- AWS Fargate
- AWS Lambda

Penyimpanan:

- AWS Backup
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx (untuk semua jenis)
- Amazon S3
- Amazon S3 Glacier
- *Gateway* AWS Storage

Layanan dan fitur di luar cakupan AWS

Daftar berikut berisi layanan dan fitur AWS yang berada di luar cakupan ujian. Daftar ini tidak lengkap dan dapat berubah sewaktu-waktu. Penawaran AWS yang sama sekali tidak terkait dengan peran pekerjaan target untuk ujian dikecualikan dari daftar ini:

Analitik:

- Amazon CloudSearch

Integrasi Aplikasi:

- Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

AR dan VR:

- Amazon Sumerian

Blockchain:

- Amazon Managed Blockchain

Komputasi:

- Amazon Lightsail

Basis Data:

- Amazon RDS on VMware

Alat Pengembang:

- AWS Cloud9
- AWS Cloud Development Kit (AWS CDK)
- AWS CloudShell
- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- Amazon CodeGuru
- AWS CodeStar
- Amazon Corretto
- Simulator *Fault Injection* AWS (AWS FIS)
- Alat dan SDK AWS

Web dan Seluler *Front-End*:

- Amazon Location Service

Teknologi *Game*:

- Amazon GameLift
- Amazon Lumberyard

Internet untuk Segala (IoT)

- Semua layanan

Machine Learning:

- Apache MXNet on AWS
- Amazon Augmented AI (Amazon A2I)
- AWS DeepComposer

- AWS Deep Learning AMIs (DLAMI)
- AWS Deep Learning Containers
- AWS DeepLens
- AWS DeepRacer
- Amazon DevOps Guru
- Amazon Elastic Inference
- Amazon HealthLake
- AWS Inferentia
- Amazon Lookout for Equipment
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon Monitron
- AWS Panorama
- Amazon Personalize
- PyTorch on AWS
- Amazon SageMaker Data Wrangler
- Amazon SageMaker Ground Truth
- TensorFlow on AWS

Management and Governance:

- AWS Chatbot
- AWS Console Mobile Application
- AWS Distro for OpenTelemetry
- AWS OpsWorks

Layanan Media:

- Peralatan dan Perangkat Lunak AWS Elemental
- AWS Elemental MediaConnect
- AWS Elemental MediaConvert
- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaStore
- AWS Elemental MediaTailor
- Amazon Interactive Video Service (Amazon IVS)

Migrasi dan Transfer:

- Migration Evaluator

Jaringan dan Pengiriman Konten:

- AWS App Mesh
- AWS Cloud Map

Teknologi Kuantum:

- Amazon Braket

Robotika:

- AWS RoboMaker

Satelit:

- AWS Ground Station

Survei

Seberapa berguna panduan ujian ini? Beri tahu kami dengan [mengikuti survei kami](#).