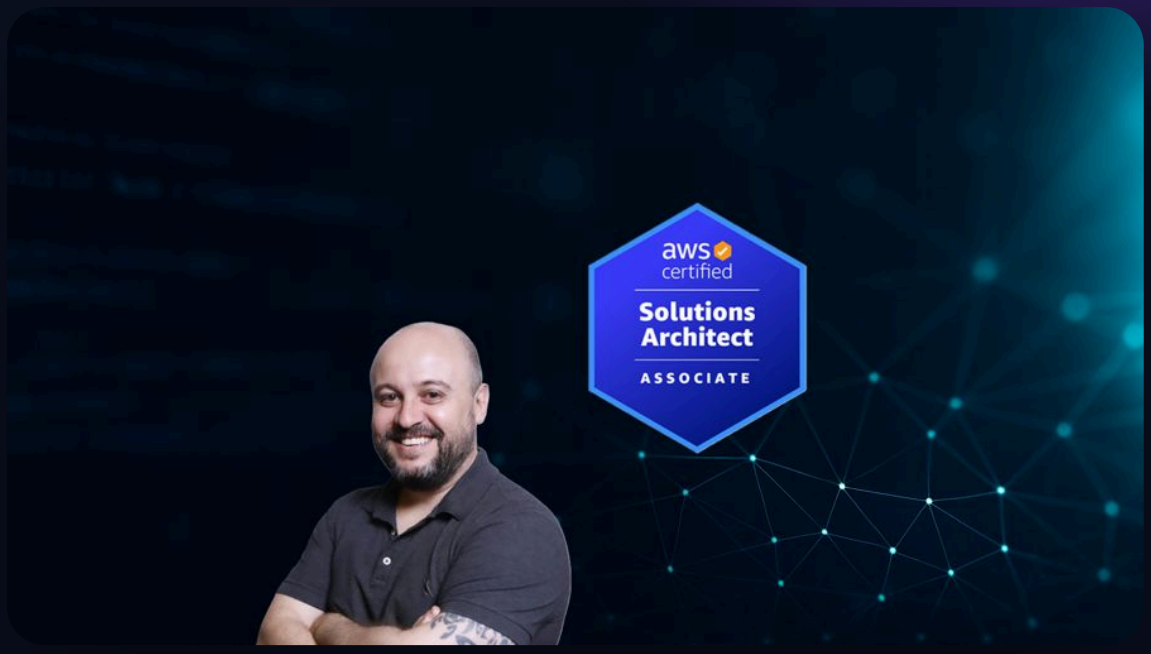


Guia de preparação SAA-C03

Tudo o que a prova cobra, organizado por domínio: conceitos-chave, comparações que confundem, pegadinhas reais, gatilhos do enunciado, questões comentadas e um plano para chegar pronto ao exame.

4 capítulos por domínio	**28** questões comentadas	**72%** nota de corte
6 provas completas no curso		

ANTES DE COMEÇAR

Como usar este guia

Este guia foi feito para ser estudado junto com as 6 provas completas do curso SAA-C03: o guia explica o que cai e onde estão as armadilhas; as provas treinam o ritmo e mostram o que você ainda não domina.

Quando	O que fazer
Semana 1	Faça o teste diagnóstico (página seguinte) e leia os capítulos 1 e 2. No fim da semana, faça a Prova 1 do curso no tempo do exame.
Semana 2	Leia os demais capítulos, começando pelos domínios em que você errou mais no diagnóstico. Faça as provas 2 e 3.
Semana 3	Releia os alertas de pegadinha e as tabelas de comparação. Faça as provas 4 e 5 e revise cada erro pela explicação.
Semana 4	Estude a folha de cola, faça a Prova 6 e refaça as provas abaixo de 80%. Agende o exame quando estiver acima de 80% em todas (a nota de corte é 72%).

Mapa do exame

Cada prova do curso tem 65 questões e 130 minutos, no formato do exame. Referência de aprovação: 72%.

Domínio (capítulo)	Peso	Neste guia
1. Domínio 1: Criação de arquiteturas seguras	30%	18 conceitos · 14 pegadinhas
2. Domínio 2: Criação de arquiteturas resilientes	26%	18 conceitos · 14 pegadinhas
3. Domínio 3: Criação de arquiteturas de alto desempenho	24%	18 conceitos · 14 pegadinhas
4. Domínio 4: Criação de arquiteturas com custo otimizado	20%	17 conceitos · 14 pegadinhas

Como ler os capítulos: os **tópicos-chave** são o que mais cai; a **tabela de memorização** separa o que a prova costuma confundir; os **alertas de prova** mostram as armadilhas das alternativas erradas; os **gatilhos** ligam palavras do enunciado à resposta; e o **checklist** diz se você está pronto para o domínio.

Estude um pouco todo dia, de graça

Instagram @strecht.simulados

Um post de prova por dia: questão comentada, pegadinhas e o "não confunda" de cada certificação.

Seguir no Instagram →

YouTube Strecht Simulados

Shorts diários com pegadinhas de prova e, aos sábados, um simulado comentado.

Inscrever-se no canal →

DIAGNÓSTICO

Você já sabe isso?

Responda às 12 questões sem consultar nada. As respostas, com a explicação e o capítulo para revisar, estão no fim do guia. Comece o estudo pelos capítulos em que errou.

D1 · Domínio 1: Criação de arquiteturas seguras

Uma seguradora tem dezenas de microsserviços internos atrás de Application Load Balancers internos, com nomes no domínio privado corp.internal resolvido por uma hosted zone privada do Route 53. A segurança exige TLS em todas as chamadas, com certificados confiáveis só dentro da empresa e renovação automática, sem operar uma CA própria em servidores. Qual solução atende a esses requisitos?

A. Solicitar certificados públicos do AWS Certificate Manager com validação por DNS para os nomes do domínio corp.internal
B. Criar uma CA privada no AWS Private CA e emitir pelo ACM certificados privados para os ALBs internos, renovados pelo ACM
C. Carregar certificados no IAM como server certificates e associá-los aos listeners HTTPS dos ALBs internos de cada microsserviço
D. Gerar certificados autoassinados com OpenSSL numa instância EC2 e importá-los no ACM para associá-los aos load balancers internos

D2 · Domínio 1: Criação de arquiteturas seguras

Uma healthtech tem 25 contas no AWS Organizations, com cargas em quatro Regiões, e a conta de segurança já é a administradora delegada do AWS Security Hub CSPM. A diretoria quer verificações automáticas e contínuas das configurações contra boas práticas da AWS, como buckets públicos, volumes EBS sem criptografia, SSH aberto e IMDSv1 em uso, com pontuação de segurança e uma visão única dos resultados das quatro Regiões na conta de segurança, com o MENOR esforço operacional. Qual solução atende a esses requisitos?

A. Habilitar o Amazon GuardDuty nas contas e Regiões e usar a quantidade de achados de severidade alta como indicador da postura de configuração
B. Habilitar o padrão AWS Foundational Security Best Practices no Security Hub CSPM das contas e configurar a agregação entre Regiões com uma Região inicial
C. Habilitar o Amazon Inspector nas contas para varrer instâncias, imagens de contêiner e funções Lambda e consolidar os achados de vulnerabilidade na conta de segurança
D. Escrever regras personalizadas do AWS Config com uma função Lambda por verificação em cada conta e Região e consultar os resultados por um agregador do Config

D3 · Domínio 1: Criação de arquiteturas seguras

Uma auditoria numa corretora encontrou volumes EBS não criptografados criados por equipes que lançam instâncias por scripts, pelo console e por templates diferentes. A nova política exige que cada volume EBS e cada cópia de snapshot criados daqui em diante em sa-east-1 sejam criptografados com uma chave do AWS KMS gerenciada pelo cliente, sem depender de cada equipe lembrar da opção. Qual solução atende a esse requisito com o MENOR esforço operacional?

A. Exigir nos templates do CloudFormation e nos launch templates o parâmetro Encrypted como true e revisar os scripts das equipes num pipeline de revisão de código
B. Habilitar a criptografia por padrão do EBS em sa-east-1 em cada conta e definir a chave KMS gerenciada pelo cliente como chave padrão de criptografia do EBS
C. Criar uma regra do AWS Config encrypted-volumes com remediação que crie um snapshot criptografado de cada volume não conforme e substitua o volume original
D. Ativar a criptografia por padrão do EBS mantendo a chave aws/ebs e criptografar os volumes existentes com o comando modify-volume da AWS CLI

D4 · Domínio 2: Criação de arquiteturas resilientes

Uma aplicação de negociação em tempo real está implantada em duas zonas de disponibilidade dentro de uma mesma região da AWS, utilizando Auto Scaling Group e Application Load Balancer. Durante um incidente regional, toda a região se torna indisponível por várias horas, causando interrupção total do serviço. A equipe de arquitetura precisa projetar uma solução que minimize o tempo de indisponibilidade em caso de falha regional completa, mantendo o RTO em poucos minutos e aceitando um RPO de até 5 minutos. Qual abordagem é MAIS adequada para atender a esses requisitos de resiliência?

A. Manter uma região e usar AWS Backup com cópias diárias entre regiões, restaurando à mão
B. Usar Multi-Region ativo-passivo com failover automático no Route 53 e dados replicados de forma assíncrona e contínua
C. Implementar uma arquitetura Multi-Region Active-Active com balanceamento de tráfego via Amazon Route 53 usando políticas de geolocalização e replicação de dados síncrona entre regiões.
D. Configurar um segundo Auto Scaling Group em outra região com instâncias em modo 'standby' e replicar os dados de forma assíncrona usando backups diários do Amazon RDS.

D5 · Domínio 2: Criação de arquiteturas resilientes

Uma empresa de seguros roda uma API em 30 instâncias EC2 de um grupo do Auto Scaling. Há dois anos a equipe aplica patches e ajustes de configuração diretamente nas instâncias em execução, e hoje cada servidor está um pouco diferente do outro: falhas aparecem só em alguns nós e voltar a uma versão anterior é demorado. A empresa quer que cada instância seja idêntica e reproduzível, com rollback rápido para a versão anterior da imagem. Qual abordagem atende a esses requisitos?

A. Habilitar regras do AWS Config que detectem diferenças de configuração entre as instâncias e enviar os desvios encontrados à equipe de operações para correção
B. Gerar a AMI com patches em um pipeline do EC2 Image Builder, publicar uma nova versão do launch template e substituir as instâncias por instance refresh
C. Aplicar os patches com o Patch Manager do AWS Systems Manager em janelas de manutenção e usar associações do State Manager para manter a configuração das instâncias em execução
D. Criar uma AMI a partir da instância de produção mais atualizada, usá-la como imagem dourada do grupo e repetir o processo a cada novo ciclo de patches

D6 · Domínio 2: Criação de arquiteturas resilientes

A plataforma de uma rede de farmácias é um monolito Java em instâncias EC2 atrás de um Application Load Balancer, servindo dezenas de rotas para os aplicativos móveis. A equipe quer extrair o módulo de receitas para um microsserviço no Amazon ECS sem uma migração de uma vez só, sem mudar a URL usada pelos aplicativos e com a possibilidade de devolver o tráfego ao monolito rapidamente. Qual solução atende a esses requisitos?

A. Usar o Amazon CloudFront com o monolito como origem e uma CloudFront Function que responda com redirecionamento HTTP 301 de /receitas para o domínio do novo serviço
B. Adicionar ao listener do ALB uma regra de caminho /receitas/* que encaminhe ao target group do serviço ECS, mantendo a regra padrão no target group do monolito
C. Criar registros ponderados no Amazon Route 53 que enviem parte do tráfego do domínio da aplicação para um balanceador do novo microsserviço
D. Reescrever o monolito inteiro como microsserviços no Amazon ECS e trocar o DNS da aplicação para o novo ambiente em uma única janela

D7 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma plataforma educacional online executa sua aplicação web em instâncias Amazon EC2 dentro de um Auto Scaling group atrás de um Application Load Balancer (ALB). Em horários de pico, a latência para os usuários aumenta significativamente, mesmo sem atingir 100% de utilização de CPU nas instâncias. A equipe de arquitetura identificou que muitas requisições são idênticas e leem dados raramente alterados a partir de um banco de dados Amazon RDS. Qual solução é mais adequada para melhorar o desempenho da aplicação e reduzir a latência para os usuários, mantendo a escalabilidade?

A. Configurar o Amazon CloudFront como CDN na frente do ALB e habilitar cache de respostas estáticas e dinâmicas quando possível
B. Migrar o banco de dados Amazon RDS para uma instância maior com mais CPU e memória, mantendo a mesma arquitetura de aplicação
C. Substituir o ALB por um Network Load Balancer (NLB) para reduzir a latência de rede e aumentar o throughput das conexões TCP
D. Adicionar mais instâncias EC2 ao Auto Scaling group e aumentar o número máximo de instâncias para lidar com o aumento de requisições

D8 · Domínio 3: Criação de arquiteturas de alto desempenho

Um escritório de engenharia vai migrar o NAS do datacenter para a AWS. Estações Linux acessam os projetos por NFS e estações Windows abrem os mesmos arquivos por SMB; os usuários recuperam versões anteriores por snapshots e mais de 70% dos dados não são abertos há meses. A empresa quer latência baixa para os arquivos ativos e custo menor para os dados frios, sem manter servidores de arquivos. Qual solução atende a esses requisitos?

A. Criar um Amazon FSx for Lustre vinculado a um bucket do Amazon S3 como repositório de dados e exportar os arquivos frios para o bucket
B. Criar um Amazon FSx for NetApp ONTAP com volumes multiprotocolo, snapshots agendados e política de tiering para o capacity pool
C. Criar um Amazon EFS com lifecycle para a classe Infrequent Access e conectar as estações Windows ao sistema de arquivos pelo protocolo SMB
D. Criar um Amazon FSx for Windows File Server com deduplicação e cópias de sombra, montando os mesmos compartilhamentos nas estações Linux por NFS

D9 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma empresa está executando um workload de analytics que lê grandes volumes de dados (centenas de TB) armazenados no Amazon S3 usando Amazon Athena. Os analistas reclamam que as consultas estão lentas e o custo por consulta aumentou significativamente. Os dados atualmente são armazenados em arquivos CSV sem nenhum particionamento. O Solutions Architect precisa melhorar a performance das consultas e reduzir custos, com o mínimo de esforço operacional. Qual é a MELHOR solução?

A. Migrar os dados para um cluster Redshift e usar o Redshift Spectrum para acessar o S3
B. Converter os dados para o formato Parquet compactado e criar partições nas tabelas do Athena com base em colunas de alto filtro, como data ou região.
C. Mover os CSV para a classe S3 Express One Zone para reduzir a latência de leitura, sem mudar o formato
D. Criar um Amazon ElastiCache for Redis e armazenar em cache os resultados das consultas mais frequentes, sem alterar o formato nem a organização dos dados no S3.

D10 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma startup mantém um ambiente de desenvolvimento em uma VPC com poucas instâncias em sub-redes privadas de três zonas de disponibilidade. Elas acessam a internet por IPv4 para baixar dependências de repositórios públicos (npm e PyPI) algumas vezes por semana, somando poucos GB por mês. O ambiente não tem requisito de alta disponibilidade e a equipe aceita administrar uma instância. Hoje há um NAT gateway zonal em cada zona. Qual solução reduz ao MÍNIMO o custo de saída para a internet nesse ambiente?

A. Substituir os três NAT gateways zonais por um NAT gateway regional para que a AWS gerencie a expansão automática entre as zonas
B. Manter um único NAT gateway zonal e apontar para ele as rotas padrão das sub-redes privadas das três zonas
C. Substituir os NAT gateways por um egress-only internet gateway e apontar para ele a rota padrão IPv4 das sub-redes privadas
D. Substituir os NAT gateways por uma NAT instance pequena em sub-rede pública, com a verificação de origem e destino desligada

D11 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma editora de livros digitais usa um AWS Global Accelerator na frente de um ALB em us-east-1 para servir o site e os downloads de e-books, arquivos estáticos que mudam pouco, a leitores do mundo todo, com 40 TB de saída por mês. Os leitores acessam por nome DNS, nenhum parceiro precisa liberar IPs fixos em firewall e não há outra Região para failover. A diretoria quer reduzir o custo de entrega mantendo ou melhorando a latência. Qual solução atende a esses requisitos com o MENOR custo?

A. Trocar o Global Accelerator por um Network Load Balancer com Elastic IPs na Região, que mantém endereços fixos para os leitores com um custo menor
B. Substituir o Global Accelerator por uma distribuição do Amazon CloudFront com o ALB como origem e políticas de cache para os e-books e os arquivos do site
C. Habilitar o S3 Transfer Acceleration num bucket com os e-books e apontar os links de download para o endpoint acelerado do bucket nas páginas do site
D. Manter o Global Accelerator e adicionar um segundo endpoint group em outra Região, com uma cópia da aplicação, para reduzir a distância até os leitores

D12 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma agência de publicidade dá a cada projeto de cliente uma conta própria na AWS Organizations, com orçamento mensal fechado. Quando o gasto real de uma conta atingir 100% do orçamento, a empresa quer impedir automaticamente o lançamento de novas instâncias EC2 e parar as instâncias de desenvolvimento do projeto, sem desenvolver código. Qual solução atende a esses requisitos?

A. Criar um alarme de faturamento do Amazon CloudWatch por conta com notificação SNS que acione uma função Lambda escrita pela equipe para parar as instâncias
B. Criar em cada conta um orçamento do AWS Budgets com budget actions em 100% que apliquem uma política IAM negando ec2:RunInstances e parem as instâncias de dev
C. Criar um orçamento no AWS Budgets com notificação por e-mail em 100% e um runbook para a equipe de nuvem aplicar uma SCP de negação na conta assim que receber o aviso
D. Habilitar o AWS Cost Anomaly Detection em cada conta de projeto e enviar os alertas de anomalia por Amazon SNS para o gerente responsável

01

CAPÍTULO 1 · 30% DA PROVA

Domínio 1: Criação de arquiteturas seguras

Aqui eu preciso dominar identidade (IAM, roles, STS, IAM Identity Center, Cognito), governança multi-conta (SCP, RCP, Control Tower, Firewall Manager), proteção de rede (security group, network ACL, VPC endpoints, WAF, Shield, Network Firewall) e proteção de dados (KMS, ACM, Secrets Manager, S3 Object Lock). É o domínio de maior peso (30%) e cai como cenário com uma restrição decisiva: sem credenciais de longo prazo, sem tráfego pela internet, controle preventivo e não só detectivo, menor esforço operacional. Onde mais erro é confundir os serviços de detecção (GuardDuty, Inspector, Macie, Security Hub CSPM, Detective) e os detalhes de KMS, como a chave gerenciada pela AWS que não pode ser compartilhada com outra conta. Duas regras resolvem metade do domínio: role com credencial temporária sempre vence chave de acesso, e Deny explícito sempre vence Allow.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

SCP (Service Control Policy)

Define o máximo de permissões que usuários e roles das contas-membro podem usar; nunca concede permissão, então ainda é preciso uma política do IAM com Allow. Não afeta a conta de gerenciamento nem service-linked roles, mas atinge o usuário raiz das contas-membro. Uma exceção se faz na própria SCP, com condição como ArnNotLike em aws:PrincipalArn.

RCP (Resource Control Policy)

Limita quem pode acessar os recursos da organização, inclusive principals de fora dela. Exemplo cobrado: negar ações do S3 quando aws:PrincipalOrgID for diferente do ID da organização, com exceção para aws:PrincipalIsAWSService. SCP controla o que os SEUS principals fazem; RCP controla quem acessa os SEUS recursos.

Permissions boundary

Teto de permissões aplicado a um usuário ou role, usado para delegar a criação de roles sem permitir escalada de privilégio. Não concede permissão e não anula um Deny de SCP. Vale para todas as sessões da role ao mesmo tempo, por isso não serve para restringir jobs simultâneos com escopos diferentes.

Política de sessão no AssumeRole

Política passada na chamada AssumeRole que reduz as permissões apenas daquela sessão; o resultado é a interseção, nunca amplia. É a resposta quando vários jobs usam a mesma role, mas cada execução deve enxergar só um prefixo do S3, sem alterar a role.

ABAC com session tags

A federação passa a tag do usuário como session tag e a política compara aws:ResourceTag/projeto com aws:PrincipalTag/projeto. Escala para muitos projetos sem editar políticas a cada novo projeto, ao contrário de uma role ou grupo por projeto (RBAC). Identidades federadas não são membros de grupos do IAM.

Confused deputy: ExternalId x SourceArn

Fornecedor terceiro que atende vários clientes e assume uma role na sua conta: trust policy com sts:ExternalId. Serviço da AWS assumindo a role ou invocando o recurso (S3 chamando Lambda, EventBridge Scheduler): condições aws:SourceArn e aws:SourceAccount. MFA não resolve confused deputy.

IAM Identity Center

Acesso de pessoas da força de trabalho ao console e à CLI de várias contas, federado ao IdP corporativo via SAML 2.0 com MFA exigido no IdP. Permission sets diferentes podem ser atribuídos ao mesmo grupo por conta (PowerUserAccess em desenvolvimento, ReadOnlyAccess em produção). Não é para aplicações nem pipelines de CI/CD.

Amazon Cognito: user pool e identity pool

User pool autentica usuários finais (provedores sociais e SAML, managed login) e emite JWT validado por authorizer do API Gateway ou pela ação de autenticação do ALB. Identity pool troca o token por credenciais temporárias da AWS para acessar S3 ou DynamoDB direto, com escopo por usuário usando a variável cognito-identity.amazonaws.com:sub.

Provedor OIDC + AssumeRoleWithWebIdentity

Pipeline externo, como GitHub Actions, assume uma role sem chave de acesso: cria-se no IAM um provedor OIDC para token.actions.githubusercontent.com e uma trust policy condicionada em aud e sub (repositório e branch). Pipeline não consegue digitar MFA e Identity Center não restringe por repositório.

Chave KMS gerenciada pelo cliente x gerenciada pela AWS

A chave do cliente tem key policy editável, pode ser desabilitada, auditada no CloudTrail, rotacionada automaticamente (padrão de 365 dias, mesmo ARN) e liberada para outras contas. As chaves gerenciadas pela AWS (aws/s3, aws/ebs, aws/rds, aws/dynamodb) têm key policy que não pode ser alterada e não podem ser usadas por outra conta, então snapshots e objetos cifrados com elas não se compartilham.

Envelope encryption (GenerateDataKey)

Para dados grandes ou cifragem no cliente, chama-se GenerateDataKey, cifra-se localmente com a chave de dados em texto claro e guarda-se a cópia cifrada da chave junto ao arquivo. A operação Encrypt direta aceita no máximo 4.096 bytes. O key material de uma chave KMS nunca sai do serviço em texto claro.

AWS CloudHSM

HSM dedicado de locatário único, sob controle exclusivo do cliente, com modo FIPS e biblioteca PKCS #11. Chave do KMS com key material importado dá controle sobre a origem, mas continua em HSMs multilocatários operados pela AWS e sem PKCS #11.

Security group x network ACL

Security group é stateful, só tem regras de permissão (negação implícita) e pode referenciar outro security group como origem. Network ACL é stateless, aceita permitir e negar, usa só blocos CIDR, vale para a sub-rede inteira e é avaliada pela ordem: a primeira regra que casa decide. Servidor atrás de NACL precisa de regra de SAÍDA para as portas efêmeras 1024-65535.

VPC endpoints: gateway x interface

Gateway endpoint existe só para S3 e DynamoDB, não tem custo e atende apenas tráfego originado na própria VPC (não atravessa peering nem Transit Gateway). Os demais serviços (KMS, Secrets Manager, SQS, SSM, Firehose) usam interface endpoint do PrivateLink, com DNS privado habilitado para não mudar a aplicação. A bucket policy restringe a origem com aws:SourceVpce ou aws:SourceVpc.

S3 Object Lock, legal hold e MFA Delete

Modo compliance: ninguém, nem o usuário raiz, apaga a versão ou reduz a retenção. Modo governance: quem tem s3:BypassGovernanceRetention consegue remover. Legal hold protege sem data definida, até ser removido. MFA Delete só pode ser habilitado pelo usuário raiz, pela CLI ou API.

IMDSv2

Exigir IMDSv2 (HttpTokens required), mantendo o limite de saltos em 1, mitiga SSRF que rouba credenciais da role da instância. Security group e NACL não filtram o endereço link-local do serviço de metadados, e o GuardDuty só detecta, não bloqueia.

Serviços de detecção e postura

GuardDuty detecta ameaça ativa (cripto, comando e controle, credencial roubada); Inspector acha CVEs em EC2, imagens do ECR e Lambda; Macie descobre dados sensíveis no S3; Security Hub CSPM avalia postura contra padrões como o AWS Foundational Security Best Practices; Detective investiga causa raiz de um achado; IAM Access Analyzer aponta recursos compartilhados com quem é de fora; AWS Config registra configuração e conformidade.

Proteção de borda: WAF, Shield e Firewall Manager

AWS WAF protege só recursos HTTP (CloudFront, ALB, API Gateway): regras gerenciadas, rate-based, IP sets e geo match. Shield Advanced acrescenta Shield Response Team, proteção de custos por DDoS e mitigação automática de camada 7. Firewall Manager aplica WAF, security groups e DNS Firewall em muitas contas, inclusive em recursos futuros.

Macetes para memorizar

💡 Avaliação de políticas

SCP põe o teto, IAM entrega a chave, Deny em qualquer camada fecha a porta.

💡 Security group x NACL

SG = Sempre Guarda a resposta (stateful, só permite). NACL = Nada Acompanha a Conexão (stateless, permite e nega, ordem numérica).

💡 Gateway endpoint

Gateway é SD: S3 e DynamoDB. Todo o resto é interface com DNS privado.

💡 Detecção

GIMS-D: GuardDuty ameaça, Inspector vulnerabilidade, Macie dado sensível, Security Hub postura, Detective investigação.

💡 Chaves aws/

Chave com nome aws/ não viaja: key policy travada, não sai da conta.

💡 CloudFront e ACM

Certificado do CloudFront mora na Virgínia: us-east-1.

💡 Object Lock

Compliance é cimento, governance é gaveta com chave, legal hold é cadeado sem data.

Tabela de memorização: não confunda

Quando é...	...e quando é
SCP Restringir o que usuários e roles das contas-membro podem fazer, como negar Regiões ou impedir que admins desliguem CloudTrail e GuardDuty.	**RCP** Impedir que principals de fora da organização acessem buckets e outros recursos da organização, mesmo que uma bucket policy os libere.

Security group Controle por instância, stateful, com origem igual a outro security group (banco aceitando só o SG da aplicação).	**Network ACL** Barreira para a sub-rede inteira, com DENY de um IP ou faixa (número menor que o ALLOW), independente do security group de cada instância.
sts:ExternalId Fornecedor terceiro, em outra conta, assume role em nome de vários clientes.	**aws:SourceArn e aws:SourceAccount** Principal de serviço da AWS (S3, EventBridge Scheduler) usa a role ou invoca a função; evita que recursos de outras contas usem o mesmo caminho.
Secrets Manager Rotação automática nativa, replicação do segredo para outra Região e acesso entre contas por política de recurso.	**Parameter Store SecureString** Configuração e segredo simples lido pela role da instância, sem exigência de rotação automática; o Parameter Store não replica entre Regiões.
Object Lock compliance Nem administrador nem usuário raiz pode apagar ou encurtar a retenção (auditoria, log imutável).	**Object Lock governance** Retenção com exceções controladas por quem tem s3:BypassGovernanceRetention.
AD Connector Usar o AD existente sem copiar diretório para a AWS (WorkSpaces, Client VPN, MFA via RADIUS); é só um proxy.	**AWS Managed Microsoft AD** Domínio gerenciado na AWS, confiança de floresta com o AD local ou RDS for SQL Server com autenticação do Windows (AD Connector não serve ao RDS).
EC2 Instance Connect Endpoint SSH e RDP privados sem bastion, sem IP público e sem instalar agente nem mudar o perfil da instância.	**Session Manager** Acesso sem porta aberta e com os comandos de cada sessão registrados no CloudWatch Logs, aceitando SSM Agent e VPC endpoints do SSM.

Números que caem

Item	Valor
Portas efêmeras na regra de saída da NACL	**TCP 1024-65535**
Período de espera para excluir chave KMS	**7 a 30 dias (depois é irreversível)**
Rotação automática da chave gerenciada pelo cliente	**Padrão de 365 dias, mesmo ARN**
Operação Encrypt direta no KMS	**Até 4.096 bytes (4 KB)**
Token de autenticação IAM do RDS	**15 minutos**
CloudTrail Event history	**Eventos de gerenciamento dos últimos 90 dias (leituras de objetos são eventos de dados e não aparecem)**
Limite de saltos com IMDSv2	**Manter em 1**

Região do certificado ACM usado no CloudFront	**us-east-1**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Acrescentar Allow na SCP ou anexar a SCP à conta de gerenciamento para resolver falta de permissão.
Como não cair: SCP nunca concede nada e não afeta a conta de gerenciamento; a permissão vem de política do IAM e a carga controlada por SCP deve ficar em conta-membro.

⚠ **Alerta de prova:** Compartilhar objetos, snapshots EBS ou RDS cifrados com aws/s3, aws/ebs ou aws/rds liberando a outra conta na bucket policy ou na key policy.
Como não cair: A key policy de chave gerenciada pela AWS não pode ser editada. Copie ou recriptografe com chave gerenciada pelo cliente e libere a conta na key policy.

⚠ **Alerta de prova:** Liberar a conta parceira só na bucket policy de um bucket com SSE-KMS de chave do cliente.
Como não cair: Sem kms:Decrypt liberado na key policy a leitura falha com acesso negado; são precisas bucket policy E key policy.

⚠ **Alerta de prova:** Criar no security group uma regra de negação para uma faixa de IP.
Como não cair: Security group só tem regras de permissão. Para bloquear faixa use DENY na network ACL com número menor que o da regra ALLOW.

⚠ **Alerta de prova:** Na NACL, liberar as portas efêmeras na entrada ou mexer no security group para as respostas aos clientes.
Como não cair: As respostas saem da sub-rede: a NACL stateless precisa de regra de SAÍDA 1024-65535. Security group já devolve a resposta sozinho.

⚠ **Alerta de prova:** Exigir MFA com Deny e condição Bool em aws:MultiFactorAuthPresent igual a false.
Como não cair: Em requisições com chave de acesso de longo prazo a chave de condição nem existe; use BoolIfExists. Allow condicionado não restringe enquanto outra política concede a ação.

⚠ **Alerta de prova:** Criar gateway endpoint para Secrets Manager, KMS, SQS ou Firehose.
Como não cair: Gateway endpoint é só para S3 e DynamoDB; o resto usa interface endpoint com DNS privado. Sem DNS privado, o nome padrão continua resolvendo para IP público.

⚠ **Alerta de prova:** Ligar a criptografia em uma instância RDS ou em um EFS já existente, ou criar réplica de leitura criptografada a partir de origem sem criptografia.
Como não cair: RDS: snapshot, cópia criptografada com a chave KMS e restauração. EFS: novo sistema de arquivos criptografado e cópia (replicação do EFS). A réplica herda o estado de criptografia da origem.

⚠ **Alerta de prova:** Usar API key do API Gateway como autenticação de parceiros.
Como não cair: API key serve a usage plans (cotas e throttling). Autenticação forte: authorizer do Cognito (JWT e escopos), autorização IAM com SigV4 ou mTLS em domínio personalizado.

⚠ **Alerta de prova:** Associar web ACL do WAF a NLB, a instâncias EC2 ou contar com ele para tráfego UDP.
Como não cair: WAF só protege recursos HTTP: CloudFront, ALB e API Gateway. NLB, EC2 e UDP ficam com NACL, security group ou Shield.

⚠ **Alerta de prova:** Usar a restrição geográfica nativa do CloudFront e uma regra Allow no WAF para abrir exceção à equipe de QA.
Como não cair: A restrição nativa bloqueia antes de a requisição chegar à web ACL. País com exceção = regra Block com geo match AND NOT IP set no próprio WAF.

⚠ **Alerta de prova:** Confiar em Object Lock ou Backup Vault Lock em modo governance, ou em MFA Delete, contra administrador comprometido.
Como não cair: Governance pode ser removido por quem tem permissão. Use modo compliance e, para backups, cópia para cofre em conta dedicada com Vault Lock compliance.

⚠ **Alerta de prova:** Escolher AWS Config, Access Analyzer ou alarme quando o enunciado pede impedir.
Como não cair: Config e Access Analyzer são detectivos. Prevenção = SCP, RCP, S3 Block Public Access no nível da conta, declarative policies.

⚠ **Alerta de prova:** Usar certificado do ACM emitido em sa-east-1 ou o *.cloudfront.net para um domínio próprio no CloudFront.
Como não cair: Certificado associado ao CloudFront precisa estar em us-east-1; o ALB usa o certificado da própria Região. Certificado importado não renova sozinho: monitore o evento ACM Certificate Approaching Expiration.

Como cai na prova

📝 **Como cai:** Organização com dezenas de contas quer impedir uma ação (Região, desligar CloudTrail, criar chave KMS) inclusive em contas futuras.
Como resolver: 1) Viu impedir = controle preventivo. 2) Ação de principals da organização = SCP na OU (com NotAction para serviços globais e exceção pela role de segurança). 3) Acesso de fora aos recursos = RCP. 4) Descarte Config, Access Analyzer e scripts, que só detectam ou corrigem depois.

📝 **Como cai:** Outra conta (DR, auditoria, parceiro) precisa ler dados cifrados no S3, EBS ou RDS.
Como resolver: 1) Verifique a chave: se for aws/, a resposta exige copiar com chave do cliente. 2) Libere a conta na key policy. 3) Libere o recurso (bucket policy ou compartilhamento de snapshot manual). 4) Snapshot automatizado do RDS precisa virar manual antes.

📝 **Como cai:** Instâncias em sub-redes privadas sem NAT precisam falar com um serviço da AWS sem passar pela internet.
Como resolver: 1) S3 ou DynamoDB = gateway endpoint (grátis). 2) Outro serviço = interface endpoint com DNS privado. 3) Para restringir o bucket à VPC, bucket policy com aws:SourceVpce. 4) Várias VPCs com um endpoint só = interface endpoint central + Transit Gateway + DNS privado.

📝 **Como cai:** Escolher o serviço de segurança certo entre cinco nomes parecidos.
Como resolver: Pergunte o que o enunciado quer: comportamento malicioso acontecendo (GuardDuty), pacote vulnerável (Inspector), onde estão dados pessoais (Macie), configuração contra padrão com pontuação (Security Hub CSPM), reconstruir o incidente (Detective), recurso compartilhado com fora (Access Analyzer).

📝 **Como cai:** API precisa autenticar clientes: parceiros B2B, usuários de app ou outras contas internas.
Como resolver: Parceiro com identidade forte = mTLS em domínio personalizado ou Cognito com JWT e escopos + usage plans. Usuário final = Cognito user pool. Interno entre contas sem internet = API privada via PrivateLink + autorização IAM (SigV4) + resource policy. API key nunca é autenticação.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
sem chaves de acesso de longo prazo, credenciais temporárias	**Role do IAM: instance profile no EC2, task role no ECS, EKS Pod Identity ou IRSA, OIDC para pipelines.**
fornecedor terceiro que atende vários clientes assume role	**Trust policy com sts:ExternalId.**
funcionários com o login corporativo em várias contas	**IAM Identity Center federado via SAML 2.0, permission sets por conta.**
usuários finais de aplicativo móvel acessando S3 ou DynamoDB direto	**Cognito user pool + identity pool com role usando cognito-identity.amazonaws.com:sub.**

limitar requisições por IP, SQL injection, XSS	**AWS WAF com regra rate-based e regras gerenciadas no CloudFront, ALB ou API Gateway.**
Shield Response Team, proteção de custo por DDoS, mitigação automática de camada 7	**AWS Shield Advanced.**
filtrar saída por nome de domínio (FQDN)	**AWS Network Firewall com domain list antes do NAT gateway; bloqueio na resolução DNS = Route 53 Resolver DNS Firewall.**
firewall de terceiros inline, transparente e altamente disponível	**Gateway Load Balancer + GWLB endpoints em VPC de inspeção.**
auditor pediu relatório SOC 2 ou atestado PCI da AWS	**AWS Artifact.**
CVE em EC2, ECR ou Lambda / CPF e cartão no S3 / malware em upload no S3	**Inspector / Macie / GuardDuty Malware Protection for S3.**
SSH/RDP sem bastion, sem IP público e sem agente	**EC2 Instance Connect Endpoint; com auditoria de comandos, Session Manager.**
domínio interno (corp.internal) com TLS e renovação gerenciada	**AWS Private CA + certificado privado do ACM.**
campo sensível do formulário que só um serviço do backend pode ler	**Criptografia em nível de campo do CloudFront.**
Direct Connect com exigência de IPsec ponta a ponta	**Site-to-Site VPN sobre o Direct Connect; MACsec é camada 2 no link dedicado.**

🎯 **Se você lembrar de só uma coisa deste capítulo...**

- Deny explícito vence qualquer Allow, e SCP nunca concede permissão nem afeta a conta de gerenciamento.
- Chave gerenciada pela AWS (aws/s3, aws/ebs, aws/rds) não pode ser compartilhada entre contas; troque para chave do cliente.
- Gateway endpoint só existe para S3 e DynamoDB.
- Security group não tem regra de negação; NACL é stateless e precisa de saída para portas efêmeras.
- Config, Access Analyzer e GuardDuty detectam; SCP, RCP e Block Public Access impedem.

Essenciais para a prova (checklist)

☐ Sei explicar por que uma SCP com Allow não dá acesso e como abrir exceção para uma role na própria SCP.

☐ Sei diferenciar sts:ExternalId de aws:SourceArn/aws:SourceAccount e quando usar cada um.

☐ Sei escolher entre IAM Identity Center, Cognito user pool, identity pool e provedor OIDC pelo tipo de usuário.

☐ Sei montar o acesso entre contas a um bucket com SSE-KMS (bucket policy + key policy + chave do cliente).

☐ Sei criptografar um RDS ou EFS que nasceu sem criptografia.

☐ Sei associar GuardDuty, Inspector, Macie, Security Hub CSPM, Detective e Access Analyzer ao problema certo.

☐ Sei configurar NACL para bloquear uma faixa e liberar respostas às portas efêmeras.

☐ Sei quando usar gateway endpoint, interface endpoint com DNS privado e bucket policy com aws:SourceVpce.

☐ Sei a diferença entre Object Lock compliance, governance, legal hold e MFA Delete.

☐ Sei que WAF só protege CloudFront, ALB e API Gateway e o que o Shield Advanced acrescenta.

Questões de revisão do capítulo

R1.1 · Domínio 1: Criação de arquiteturas seguras

Uma rede de academias opera em sa-east-1 e já usa um certificado público do AWS Certificate Manager, solicitado nessa Região, no listener HTTPS do seu Application Load Balancer para www.exemplo.com.br. Ao criar uma distribuição do Amazon CloudFront na frente do ALB com o mesmo nome de domínio, o certificado não aparece na lista de certificados disponíveis. Qual ação permite servir HTTPS com o domínio próprio no CloudFront?

A. Usar o certificado padrão *.cloudfront.net da distribuição e criar no DNS um registro CNAME de www.exemplo.com.br para o domínio da distribuição
B. Solicitar no ACM em us-east-1 um certificado público para www.exemplo.com.br com validação por DNS e associá-lo à distribuição, mantendo o de sa-east-1 no ALB
C. Criar a distribuição do CloudFront na Região sa-east-1 para que ela passe a enxergar os certificados do ACM emitidos nessa mesma Região
D. Emitir um certificado privado pelo AWS Private CA em sa-east-1 para www.exemplo.com.br e associá-lo à distribuição do CloudFront como certificado personalizado

R1.2 · Domínio 1: Criação de arquiteturas seguras

Uma API pública recebe rajadas de requisições de poucos endereços IP, caracterizando abuso. A empresa quer bloquear automaticamente os IPs que excederem um limite de requisições em cinco minutos. Qual solução atende?

A. Grupo de segurança com limite de conexões
B. AWS Shield Standard com alarme por e-mail
C. Network ACL com regra de negação por IP fixo
D. Regra baseada em taxa no AWS WAF

R1.3 · Domínio 1: Criação de arquiteturas seguras

Um aplicativo de telemedicina permite que pacientes enviem exames em PDF que ficam em um bucket S3 privado. Cada paciente autenticado no backend deve enviar o arquivo direto do navegador para o S3, sem passar pelo servidor, com acesso válido por poucos minutos e restrito a uma chave de objeto específica. O bucket deve continuar com o S3 Block Public Access ativado. Qual solução atende a esses requisitos?

A. Configurar CORS no bucket permitindo o método PUT a partir do domínio do aplicativo e enviar os arquivos ao S3 sem assinatura
B. Criar um usuário IAM para cada paciente com permissão s3:PutObject no prefixo dele e entregar as chaves de acesso ao navegador do aplicativo
C. Gerar no backend uma URL pré-assinada de PUT para a chave do paciente, com expiração de 5 minutos, usando as credenciais da role do backend
D. Adicionar uma bucket policy que permita s3:PutObject ao principal * quando o cabeçalho aws:Referer for o domínio do aplicativo

R1.4 · Domínio 1: Criação de arquiteturas seguras

Uma empresa de mídia tem 70 contas no AWS Organizations, com distribuições do CloudFront e ALBs criados pelas equipes de produto. Após um ataque de injeção de SQL, a segurança exige que cada ALB e cada distribuição, inclusive os criados no futuro, tenham uma web ACL do AWS WAF com o grupo de regras gerenciadas contra SQL injection, e que nenhum security group permita SSH de 0.0.0.0/0. As equipes não podem remover essas proteções. Qual solução atende a esses requisitos com o MENOR esforço operacional?

A. Anexar uma SCP que negue elasticloadbalancing:CreateLoadBalancer e cloudfront:CreateDistribution a quem não tiver a tag de aprovação da equipe de segurança
B. Designar uma conta administradora do AWS Firewall Manager e criar políticas do AWS WAF e de auditoria de security groups com remediação automática no escopo da organização
C. Criar em cada conta uma regra do AWS Config que detecte ALBs e distribuições sem web ACL e enviar os achados por e-mail às equipes responsáveis pela correção
D. Publicar a web ACL por um StackSet do CloudFormation em cada conta da organização e pedir às equipes de produto que associem a web ACL aos novos ALBs e distribuições que criarem

Respostas comentadas

R1.1. B) Solicitar no ACM em us-east-1 um certificado público para www.exemplo.com.br com validação por DNS e associá-lo à distribuição, mantendo o de sa-east-1 no ALB

Por quê: Para usar um certificado do ACM em uma distribuição do CloudFront, ele precisa ser solicitado ou importado na Região us-east-1. Recursos regionais, como ALB e API Gateway regional, usam certificados da própria Região. Certificados públicos do ACM validados por DNS são renovados automaticamente enquanto o registro de validação existir.

✗ A) Incorreta. O certificado *.cloudfront.net não cobre www.exemplo.com.br; os navegadores acusariam nome de certificado incompatível ao acessar pelo domínio próprio.

✗ C) Incorreta. O CloudFront é um serviço global e não se cria a distribuição em uma Região; a exigência é que o certificado do ACM esteja em us-east-1.

✗ D) Incorreta. Um certificado privado não é confiável para os navegadores do público e, além disso, o certificado associado ao CloudFront precisa estar em us-east-1.

> **Dica de prova:** Certificado do ACM para CloudFront = sempre em us-east-1. ALB usa o certificado da sua Região.

R1.2. D) Regra baseada em taxa no AWS WAF

Por quê: Regras baseadas em taxa do AWS WAF contam as requisições de cada IP em uma janela de tempo e bloqueiam automaticamente os que ultrapassam o limite. Grupos de segurança não limitam taxa, Network ACLs exigem listar IPs um a um e o Shield Standard protege contra DDoS de rede, sem regras de taxa configuráveis por IP.

✗ A) Incorreta. Grupos de segurança não controlam taxa de requisições.

✗ B) Incorreta. Protege contra DDoS de rede, sem regras de taxa por IP.

✗ C) Incorreta. Exige conhecer e listar os IPs com antecedência.

> **Dica de prova:** Limitar requisições por IP: regra baseada em taxa do AWS WAF.

R1.3. C) Gerar no backend uma URL pré-assinada de PUT para a chave do paciente, com expiração de 5 minutos, usando as credenciais da role do backend

Por quê: Uma URL pré-assinada do S3 inclui a assinatura de um principal com permissão para a operação (GET, PUT etc.) e uma expiração. Quem tem a URL executa a operação até o prazo, sem credenciais próprias, e o bucket continua privado. Se ela for gerada com credenciais temporárias, expira junto com elas.

✗ A) Incorreta. O CORS só permite que o navegador faça requisições entre origens; ele não concede permissão de gravação, e requisições sem assinatura a um bucket privado são negadas.

✗ B) Incorreta. Usuários IAM por paciente não escalam e expor chaves de longo prazo no navegador é um risco grave de segurança.

✗ D) Incorreta. Uma política com principal * torna o bucket público e é bloqueada pelo Block Public Access; o cabeçalho Referer pode ser forjado.

> **Dica de prova:** Upload ou download temporário por usuário, com bucket privado = URL pré-assinada gerada pelo backend.

R1.4. B) Designar uma conta administradora do AWS Firewall Manager e criar políticas do AWS WAF e de auditoria de security groups com remediação automática no escopo da organização

Por quê: AWS Firewall Manager administra de forma central, em várias contas, proteções de AWS WAF, AWS Shield Advanced, security groups e network ACLs, AWS Network Firewall e Route 53 Resolver DNS Firewall. Você define a política uma vez e o serviço a aplica automaticamente às contas e recursos no escopo, inclusive os criados depois. Pré-requisitos: AWS Organizations, uma conta administradora do Firewall Manager e o AWS Config habilitado nas contas.

✗ A) Incorreta. A SCP bloqueia a criação de recursos, mas não associa web ACL nem corrige security groups; ela não garante as proteções exigidas.

✗ C) Incorreta. O AWS Config só detecta; a correção continua dependendo das equipes e nada impede que a proteção seja removida depois.

✗ D) Incorreta. O StackSet distribui a web ACL, mas a associação aos recursos novos depende de cada equipe, e não há reaplicação se alguém removê-la.

Dica de prova: WAF e security groups padronizados em muitas contas, valendo para recursos futuros = Firewall Manager.

02

CAPÍTULO 2 · 26% DA PROVA

Domínio 2: Criação de arquiteturas resilientes

Aqui eu preciso dominar desacoplamento (SQS, SNS, EventBridge, Step Functions, Lambda assíncrona), alta disponibilidade em várias AZs (ALB, Auto Scaling, RDS Multi-AZ, EFS, NAT gateway) e recuperação de desastres entre Regiões (pilot light, warm standby, Aurora Global Database, Route 53, ARC). Vale 26% e cai muito como fila que processa em dobro, banco que não pode cair ou Região inteira fora com RTO e RPO definidos. Onde mais erro é trocar delay queue por visibility timeout, achar que réplica de leitura faz failover automático e esquecer que EBS, instance store e EFS One Zone ficam presos a uma AZ. O raciocínio que acerta: tire o estado da instância, coloque uma fila entre quem recebe e quem processa e escolha a estratégia de DR pelo RTO e pelo custo pedidos.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

SQS: delay queue, visibility timeout e long polling

Delay queue (DelaySeconds) esconde a mensagem logo após o envio, por até 15 minutos. Visibility timeout esconde a mensagem depois que um consumidor a recebe, para evitar processamento duplicado. Long polling (ReceiveMessageWaitTimeSeconds até 20 s) só reduz respostas vazias e custo de chamadas.

SQS com Lambda e DLQ

O visibility timeout da fila deve ser pelo menos 6 vezes o timeout da função; a Lambda rejeita timeout de função maior que o visibility timeout. A redrive policy envia à DLQ após maxReceiveCount recebimentos, e a AWS recomenda pelo menos 5 para não descartar falhas temporárias.

SQS Standard x FIFO

Standard entrega pelo menos uma vez, então o consumidor deve ser idempotente (gravação condicional com o ID do evento). FIFO garante ordem por MessageGroupId e deduplica envios repetidos em um intervalo de 5 minutos, mas não deduplica retentativas do consumidor. Não existe conversão de fila Standard para FIFO. Mensagem máxima de 1 MiB; carga maior vai para o S3 com a Extended Client Library.

SNS fan-out e filter policy

Um evento para vários consumidores independentes e duráveis: tópico SNS com uma fila SQS por sistema. A filter policy na assinatura entrega só os eventos de interesse sem mudar o produtor. O SNS não guarda mensagens; tópico SNS padrão não aceita fila FIFO, e tópico FIFO entregando para fila padrão perde a ordem exata.

Amazon EventBridge

Barramento com regras que filtram por conteúdo e entregam a alvos variados, como Step Functions e API destinations com autenticação. Archive e replay reprocessam eventos já entregues; retry e DLQ só cobrem falha de entrega. EventBridge Scheduler cria milhões de agendamentos pontuais com fuso horário.

Step Functions Standard x Express

Standard é para fluxos longos, não idempotentes e com espera humana via .waitForTaskToken, com Retry para erro temporário e Catch para desviar ou compensar (saga). Express dura no máximo 5 minutos, é cobrado por execução, duração e memória e serve a milhões de execuções curtas e idempotentes.

Lambda assíncrona: destino on-failure x DLQ

O destino on-failure recebe o registro completo da invocação, com requisição e resposta; a DLQ guarda só o evento e atributos do erro. Filas e tópicos FIFO não são suportados como destino. A invocação assíncrona faz no máximo duas novas tentativas e aceita idade máxima do evento de 6 horas.

Simultaneidade reservada x provisionada

Reservada põe um teto na escala da função (protege o banco e as outras funções) e não tem custo. Provisionada mantém ambientes pré-inicializados para eliminar cold start e é cobrada à parte; não limita a escala.

Estratégias de DR

Backup e restauração: horas, mais barato. Pilot light: dados replicados continuamente e aplicação desligada até o desastre, RTO de dezenas de minutos. Warm standby: pilha reduzida sempre ligada, RTO de minutos. Multi-site ativo-ativo: capacidade total nas Regiões, quase zero, mais caro.

RDS Multi-AZ, Multi-AZ DB cluster e réplica de leitura

Multi-AZ de instância tem standby síncrona que não atende leitura e faz failover automático pelo mesmo endpoint. Multi-AZ DB cluster (RDS MySQL/PostgreSQL) tem 1 escrita + 2 leitoras em três AZs, com failover mais rápido. Réplica de leitura é assíncrona, escala leitura e exige promoção pelo operador.

Aurora: failover tiers, endpoints e Global Database

A prioridade tier-0 é a primeira candidata à promoção e tier-15 a última; a aplicação grava pelo endpoint de cluster, que segue a nova primária. Aurora Global Database dá RPO de segundos e RTO de cerca de um minuto entre Regiões, com leituras locais.

Route 53: políticas e health checks

Failover (ativo-passivo com health check), weighted (dividir X% do tráfego), latency, geolocation, multivalue answer (vários IPs saudáveis na mesma resposta, com health check por valor) e calculated health check (regra N de M componentes). ARC routing control dá failover decidido por pessoas com safety rules.

Zonal shift do ARC

Afasta temporariamente o tráfego do ALB de uma AZ degradada cujos health checks continuam verdes, por prazo definido e reversível. Com zonal autoshift, a AWS inicia o shift quando detecta o problema.

Auto Scaling + ALB em várias AZs

Instance refresh troca a AMI com porcentagem mínima saudável; health check do ELB no ASG com grace period evita tráfego a instâncias ainda inicializando; lifecycle hook de término + deregistration delay concluem requisições longas no scale-in. Cross-zone ativo e capacidade mínima planejada para N-1 zonas absorvem a perda de uma AZ.

Estado fora da instância

Sessões vão para ElastiCache ou DynamoDB e arquivos para S3 ou EFS; sticky sessions não substituem sessão externa. Instance store só sobrevive a reboot. EBS existe em uma AZ: Multi-Attach (io1/io2) e Elastic Volumes não mudam de zona; para outra AZ, snapshot e novo volume.

NAT gateway regional x zonal

NAT gateway regional dá alta disponibilidade com o mínimo de operação para todas as AZs. Zonais exigem um por AZ, com tabela de rotas por AZ. Instância NAT única é ponto único de falha; NAT gateway privado não dá acesso à internet.

AWS Backup e recuperação no ponto exato

Backup contínuo dá restauração point-in-time com RPO de minutos; replicação copia corrupção e não protege contra dado errado. Restore testing prova que o backup restaura, de forma agendada e sem script. DynamoDB PITR restaura sempre em tabela nova. AWS Elastic Disaster Recovery dá RPO de segundos e RTO de minutos para servidores.

Retry com backoff exponencial e jitter

Throttling de API da AWS se resolve com o modo de repetição padrão dos SDKs (backoff exponencial, jitter e limite de tentativas). Intervalo fixo e repetição mais rápida sincronizam os clientes e agravam o efeito manada.

Macetes para memorizar

SQS

Delay é ANTES de entregar; Visibility é DEPOIS de receber; Long polling é ESPERAR na leitura.

Concorrência da Lambda

Reservada põe Rédea (teto, grátis); Provisionada Pré-aquece (paga).

DR

Escada de temperatura: Backup frio, Piloto aceso, Morno ligado pequeno, Ativo quente. Quanto mais quente, mais rápido e mais caro.

RDS

Multi-AZ = Mantém no Ar; Read Replica = Reforço de Leitura.

Aurora tiers

Tier zero é o primeiro da fila da promoção.

Step Functions

Express é expresso: até 5 minutos e não espera humano.

Armazenamento local

Instance store só sobrevive ao reboot; EBS mora numa AZ só.

Tabela de memorização: não confunda

Quando é...	...e quando é
Delay queue (DelaySeconds) Adiar o primeiro processamento de toda mensagem nova por até 15 minutos, sem mudar produtor nem consumidor.	**Visibility timeout** Esconder a mensagem enquanto um consumidor a processa; com Lambda, pelo menos 6 vezes o timeout da função.
Simultaneidade reservada Limitar quantas execuções da função rodam ao mesmo tempo para proteger banco e outras funções, sem custo.	**Simultaneidade provisionada** Eliminar cold start mantendo ambientes inicializados, pagando por eles o tempo todo.

RDS Multi-AZ Alta disponibilidade com standby síncrona e failover automático pelo mesmo endpoint.	**Réplica de leitura** Escalar leituras e relatórios; replicação assíncrona e promoção manual (entre Regiões serve de base para DR).
Pilot light RTO de horas ou dezenas de minutos aceitável, dados replicados e nenhum servidor de aplicação ligado na Região de recuperação.	**Warm standby** RTO e RPO de minutos, com uma versão reduzida da pilha sempre em execução e escalada no failover.
SNS com filas SQS Fan-out simples para filas e Lambdas, com filter policy por atributos.	**EventBridge** Roteamento por conteúdo para alvos variados (Step Functions, API destinations com chave), com archive e replay.
Step Functions Standard Fluxo longo, aprovação humana com .waitForTaskToken, saga com compensações e histórico de execução.	**Step Functions Express** Milhões de execuções curtas e idempotentes, até 5 minutos, onde a cobrança por transição pesaria.
Route 53 weighted Migrar gradualmente mandando 10%, 50% e depois 100% do tráfego para o novo ambiente.	**Route 53 failover** Página de contingência ou Região secundária que só recebe tráfego quando o primário fica não saudável.

Números que caem

Item	Valor
DelaySeconds máximo do SQS	**15 minutos**
Espera do long polling	**Até 20 segundos**
Tamanho máximo de mensagem do SQS	**1 MiB**
Visibility timeout com Lambda	**Pelo menos 6 vezes o timeout da função**
maxReceiveCount recomendado na redrive policy	**Pelo menos 5**
Janela de deduplicação do SQS FIFO	**5 minutos**
Timeout máximo da Lambda	**900 segundos (15 minutos)**
Duração máxima do Step Functions Express	**5 minutos**
Lambda assíncrona	**No máximo 2 novas tentativas; idade máxima do evento de 6 horas**
Retentativas HTTP/S do SNS	**No máximo uma hora**
Prioridade de failover do Aurora	**tier-0 (primeira) a tier-15 (última)**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Aumentar o visibility timeout para que mensagens novas fiquem ocultas antes do primeiro processamento.
Como não cair: O visibility timeout só começa quando um consumidor recebe a mensagem. Atraso inicial = delay queue; acima de 15 minutos = EventBridge Scheduler.

⚠ **Alerta de prova:** Migrar para FIFO com deduplicação para acabar com créditos duplicados.
Como não cair: FIFO deduplica envios repetidos em 5 minutos, não reentregas após falha do consumidor. Torne o consumidor idempotente com gravação condicional (attribute_not_exists).

⚠ **Alerta de prova:** Uma única fila SQS lida pelos três sistemas que precisam de todos os eventos.
Como não cair: Cada mensagem é consumida por um só consumidor. Vários consumidores independentes = SNS com uma fila SQS por sistema (SNS FIFO para SQS FIFO se exigir ordem).

⚠ **Alerta de prova:** Usar réplica de leitura como solução de alta disponibilidade com failover automático.
Como não cair: A réplica é assíncrona e a promoção é manual, com troca de endpoint. Failover automático entre AZs = Multi-AZ.

⚠ **Alerta de prova:** Usar a standby do Multi-AZ de instância para dividir leituras ou colocar RDS Proxy para ela atender tráfego.
Como não cair: A standby de Multi-AZ de instância não atende leitura. Standbys legíveis = Multi-AZ DB cluster; leitura extra = réplicas.

⚠ **Alerta de prova:** Converter o EBS para io2 Multi-Attach ou usar Elastic Volumes para usar o volume em outra AZ.
Como não cair: Multi-Attach vale só na mesma AZ e Elastic Volumes muda tipo, tamanho e desempenho, não a zona. Snapshot e novo volume na AZ de destino.

⚠ **Alerta de prova:** Configurar EC2 auto recovery para sobreviver à falha da zona.
Como não cair: Auto recovery volta a instância só na mesma AZ. Instância única resiliente a AZ = Auto Scaling group com mínimo e máximo 1 em várias AZs e dados no EFS ou S3.

⚠ **Alerta de prova:** Aumentar o timeout da Lambda para 40 minutos ou orquestrá-la com Step Functions Express para ganhar tempo.
Como não cair: A Lambda para em 900 segundos e o Express dura no máximo 5 minutos. Processo longo = tarefa do ECS no Fargate ou AWS Batch.

⚠ **Alerta de prova:** Usar réplica entre Regiões ou replicação do S3 para se proteger de dados corrompidos pela aplicação.
Como não cair: Réplicas copiam a corrupção quase na hora. Para voltar a um momento anterior: backup contínuo (PITR) do AWS Backup ou PITR do DynamoDB.

⚠ **Alerta de prova:** Usar .waitForTaskToken ou estado Wait de dias em fluxo Express.
Como não cair: Callback com token e espera longa só no fluxo Standard; Express é Request Response e dura até 5 minutos.

⚠ **Alerta de prova:** Deixar health checks mais agressivos para detectar falha que não derruba o servidor.
Como não cair: Falha de integração com instâncias saudáveis se detecta com métrica de negócio (pedidos por minuto, dados ausentes como violação) ou canário do CloudWatch Synthetics; AZ degradada com health check verde = zonal shift.

⚠ **Alerta de prova:** Aumentar o deregistration delay para que instâncias novas só recebam tráfego quando prontas.
Como não cair: Deregistration delay vale para targets saindo do target group. Entrada segura = health check HTTP em endpoint de prontidão + health check do ELB no ASG com grace period.

⚠ **Alerta de prova:** Pedir aumento da cota de vCPUs em us-east-1 para garantir o failover em us-west-2, ou trocar para Spot achando que não há cota.
Como não cair: Cotas de vCPU do EC2 são por Região, e Spot tem cota própria. Peça antes o aumento no Service Quotas da Região de DR.

⚠ **Alerta de prova:** Usar SNS sozinho com mais retentativas como buffer para consumidores instáveis.
Como não cair: O SNS não armazena mensagens; a política HTTP/S tenta por no máximo uma hora. Inscreva uma fila SQS no tópico e use DLQ.

Como cai na prova

📝 **Como cai:** Desastre regional com RTO, RPO e orçamento definidos; pede a estratégia de DR mais adequada.
Como resolver: 1) Leia RTO e RPO. 2) Horas e menor custo = backup e restauração; dados replicados e aplicação desligada = pilot light; minutos = warm standby; quase zero = ativo-ativo. 3) Elimine opções que violam restrição explícita, como não manter servidores de aplicação ligados. 4) Multi-AZ nunca resolve falha de Região.

📝 **Como cai:** Fila com processamento duplicado, mensagens perdidas ou picos derrubando o backend.
Como resolver: 1) Picos = fila SQS entre quem recebe e quem processa. 2) Duplicidade com SQS + Lambda = visibility timeout 6x; duplicidade por natureza da Standard = consumidor idempotente. 3) Falhas a isolar = DLQ via redrive policy. 4) Ordem por cliente = FIFO com MessageGroupId.

📝 **Como cai:** Aplicação em uma única instância com sessões em memória e arquivos em disco local precisa sobreviver à falha de AZ.
Como resolver: 1) Externalize sessão (ElastiCache ou DynamoDB) e arquivos (S3 ou EFS regional). 2) Coloque Auto Scaling group em várias AZs atrás de ALB. 3) Banco com RDS Multi-AZ. 4) Descarte snapshots, auto recovery e EFS One Zone, que não sobrevivem à AZ.

📝 **Como cai:** Implantação ou scale-in derruba requisições em andamento.
Como resolver: 1) Troca de AMI = instance refresh com porcentagem mínima saudável. 2) ECS = minimumHealthyPercent 100%, health checks e circuit breaker. 3) Scale-in = deregistration delay + lifecycle hook de término. 4) Blue/green atrás de um ALB = target groups ponderados (Route 53 ponderado depende do TTL).

📝 **Como cai:** Escolher a política do Route 53 para um requisito de tráfego.
Como resolver: Dividir porcentagem = weighted; contingência = failover com health check; vários IPs saudáveis na resposta = multivalue answer; saúde da Região depende de N de M serviços = calculated health check; troca decidida por pessoas com proteção = ARC routing control.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
mensagens invisíveis por alguns minutos logo após o envio	**Delay queue (DelaySeconds até 15 minutos).**
muitas respostas vazias, custo alto de ReceiveMessage	**Long polling com ReceiveMessageWaitTimeSeconds de 20 segundos.**
arquivo de centenas de MB na mensagem da fila	**SQS Extended Client Library: objeto no S3, ponteiro na mensagem.**
SQS + Lambda processando a mesma mensagem duas vezes	**Visibility timeout de pelo menos 6 vezes o timeout da função.**
Lambda que só repassa a requisição para SQS, EventBridge, Kinesis ou Step Functions	**Integração direta do API Gateway com role do IAM (HTTP API SQS-SendMessage).**
reprocessar eventos que já foram entregues e processados errado	**Archive + replay do EventBridge.**

aprovação humana que pode levar dias no meio do fluxo	**Step Functions Standard com .waitForTaskToken.**
RPO de segundos e RTO de cerca de um minuto entre Regiões, com leitura local	**Aurora Global Database.**
RPO zero entre Regiões	**DynamoDB global tables com consistência forte multirregião e capacidade ativa nas Regiões.**
dados replicados continuamente, aplicação desligada, menor custo	**Pilot light.**
AZ degradada mas health checks verdes	**Zonal shift do Amazon Application Recovery Controller.**
detectar falha na jornada do usuário de madrugada, sem tráfego	**Canário do CloudWatch Synthetics com alarme.**
descobrir qual microsserviço atrasa a requisição	**AWS X-Ray (trace map).**
aplicação com JMS, AMQP ou MQTT migrando sem reescrever	**Amazon MQ em modo active/standby em duas AZs.**

🎯 **Se você lembrar de só uma coisa deste capítulo...**

- Réplica de leitura não faz failover automático; Multi-AZ faz.
- SQS Standard entrega pelo menos uma vez: consumidor idempotente.
- Lambda passa de 15 minutos? Fargate ou AWS Batch.
- EBS, instance store e EFS One Zone ficam em uma AZ só.
- Replicação não protege contra corrupção; backup contínuo protege.

Essenciais para a prova (checklist)

- ☐ Sei diferenciar delay queue, visibility timeout e long polling e os limites de cada um.
- ☐ Sei montar fan-out SNS + SQS com filter policy e a combinação SNS FIFO + SQS FIFO.
- ☐ Sei quando usar Step Functions Standard ou Express, Retry, Catch e .waitForTaskToken.
- ☐ Sei escolher entre backup e restauração, pilot light, warm standby e ativo-ativo pelo RTO e pelo custo.
- ☐ Sei a diferença entre RDS Multi-AZ de instância, Multi-AZ DB cluster e réplica de leitura.
- ☐ Sei usar failover tiers e endpoint de cluster no Aurora e quando escolher Aurora Global Database.
- ☐ Sei escolher a política do Route 53 e quando usar ARC routing control ou zonal shift.
- ☐ Sei deixar uma aplicação stateful resiliente à perda de AZ.
- ☐ Sei a diferença entre simultaneidade reservada e provisionada e entre destino on-failure e DLQ na Lambda.

Questões de revisão do capítulo

R2.1 · Domínio 2: Criação de arquiteturas resilientes

Uma fintech provisiona toda a infraestrutura com AWS CloudFormation. Durante incidentes, operadores às vezes alteram security groups e parâmetros direto no console, e depois atualizações de stack falham ou desfazem essas mudanças sem aviso. A auditoria pede que a empresa identifique continuamente quais stacks divergiram do template e gere um alerta, sem bloquear as mudanças emergenciais. Qual solução atende a esses requisitos com o MENOR esforço operacional?

A. Habilitar a proteção contra encerramento nas stacks para impedir que os recursos provisionados sejam alterados fora do CloudFormation
B. Gerar um change set antes de cada atualização de stack e revisar o resumo de mudanças para descobrir o que foi alterado pelo console desde a última implantação
C. Ativar a regra gerenciada cloudformation-stack-drift-detection-check do AWS Config para detectar drift e marcar como não conformes as stacks divergentes
D. Aplicar uma stack policy que negue Update:* aos recursos de rede para bloquear as alterações feitas pelo console nos security groups

R2.2 · Domínio 2: Criação de arquiteturas resilientes

Uma empresa precisa de um banco relacional com RPO de segundos e RTO de cerca de um minuto em caso de falha de uma Região inteira, além de leituras locais de baixa latência em outra Região. Qual solução atende a esses requisitos?

A. RDS Multi-AZ DB cluster com snapshots copiados pelo AWS Backup para outra Região
B. Amazon Aurora Global Database com um cluster secundário na outra Região
C. Snapshots automáticos do RDS copiados diariamente para outra Região
D. Réplica de leitura entre Regiões do RDS for MySQL, promovida pelo operador no desastre

R2.3 · Domínio 2: Criação de arquiteturas resilientes

Uma rede de farmácias está migrando o site de e-commerce do data center para uma nova pilha na AWS atrás de um ALB. A diretoria quer enviar 10% dos clientes para a nova pilha na primeira semana, depois 50% e por fim 100%, podendo voltar o tráfego para o data center em minutos se houver problemas. O DNS do domínio já está no Route 53. Qual configuração atende a esses requisitos?

A. Criar registros ponderados com o mesmo nome para o data center e o ALB e ajustar os pesos de 90/10 para 50/50 e depois 0/100
B. Criar registros com política de failover, com o data center como primário e o ALB como secundário, ajustando os health checks a cada etapa
C. Criar registros de resposta com vários valores com os IPs do data center e do ALB e ajustar a proporção de valores devolvidos
D. Criar registros com roteamento por latência para o data center e para o ALB, deixando o Route 53 escolher o destino mais rápido

R2.4 · Domínio 2: Criação de arquiteturas resilientes

Uma instituição de ensino protege bancos Amazon RDS e volumes Amazon EBS com planos do AWS Backup. A auditoria passou a exigir evidência periódica de que os backups realmente restauram e de quanto tempo a restauração leva, e a equipe não quer manter scripts nem pagar por recursos restaurados esquecidos. Qual solução atende a esses requisitos com o MENOR esforço operacional?

A. Criar um plano de restore testing do AWS Backup com frequência, seleção de recursos e validação, que exclui os recursos restaurados
B. Criar uma função Lambda agendada pelo EventBridge que restaure um ponto de recuperação, rode testes, grave a duração e exclua os recursos
C. Criar um framework do AWS Backup Audit Manager com controles de frequência e retenção de backup e gerar relatórios mensais para a auditoria
D. Habilitar o AWS Backup Vault Lock em modo compliance no cofre para garantir que os pontos de recuperação fiquem íntegros e restauráveis

Respostas comentadas

R2.1. C) Ativar a regra gerenciada cloudformation-stack-drift-detection-check do AWS Config para detectar drift e marcar como não conformes as stacks divergentes

Por quê: A detecção de drift do CloudFormation identifica recursos alterados fora da stack (propriedades modificadas ou recursos excluídos) e mostra as diferenças por propriedade. Para controle contínuo, a regra gerenciada cloudformation-stack-drift-detection-check do AWS Config automatiza a verificação, apoiando a integridade da infraestrutura como código sem impedir intervenções emergenciais.

✗ A) Incorreta. A proteção contra encerramento só impede que a stack seja excluída; não detecta nem impede alterações nos recursos feitas fora do CloudFormation.

✗ B) Incorreta. O change set compara o template novo com o que a stack registra, não com a configuração real dos recursos; ele não revela o drift e depende de revisão humana a cada implantação.

✗ D) Incorreta. A stack policy protege recursos contra atualizações feitas pela própria stack; ela não alcança mudanças pelo console e, se alcançasse, bloquearia as correções emergenciais que devem continuar possíveis.

> **Dica de prova:** Saber se alguém mudou recurso fora do CloudFormation = detecção de drift (automatizada pela regra do AWS Config). Stack policy e termination protection não detectam drift.

R2.2. B) Amazon Aurora Global Database com um cluster secundário na outra Região

Por quê: O Aurora Global Database replica os dados para clusters secundários em outras Regiões pela camada de armazenamento, com atraso típico abaixo de um segundo, e permite promover uma Região secundária em cerca de um minuto. Os clusters secundários também atendem leituras locais de baixa latência. Réplicas de leitura entre Regiões do RDS usam a replicação do motor, com atraso variável e promoção mais lenta, e snapshots copiados entre Regiões têm RPO de horas.

✗ A) Incorreta. O Multi-AZ DB cluster protege contra falha de AZ dentro da Região; snapshots copiados para outra Região dão RPO de horas e restauração lenta, sem leituras locais na segunda Região.

✗ C) Incorreta. Cópias diárias de snapshot levam a um RPO de até um dia, e restaurar o banco a partir do snapshot leva bem mais que um minuto.

✗ D) Incorreta. A réplica entre Regiões atende leituras locais, mas a replicação do binlog é assíncrona e o atraso pode crescer sob carga; a promoção é feita pelo operador e leva minutos, além de exigir apontar a aplicação para o novo endpoint.

> **Dica de prova:** RPO de segundos e RTO de cerca de um minuto entre Regiões, com leituras locais: Aurora Global Database.

R2.3. A) Criar registros ponderados com o mesmo nome para o data center e o ALB e ajustar os pesos de 90/10 para 50/50 e depois 0/100

Por quê: O roteamento ponderado do Route 53 associa vários recursos ao mesmo nome e distribui as consultas na proporção do peso de cada registro dividido pela soma dos pesos. É o padrão para migrações graduais e testes de novas versões entre ambientes diferentes, como data center e AWS; peso 0 tira um destino de circulação.

✗ B) Incorreta. O failover envia todo o tráfego ao primário enquanto ele está saudável; não há como dividir 10% ou 50% entre os ambientes.

✗ C) Incorreta. A resposta com vários valores devolve registros saudáveis de forma aleatória, sem pesos, e o ALB não tem IPs fixos para cadastrar.

✗ D) Incorreta. O roteamento por latência escolhe o destino de menor latência para cada usuário, sem controle sobre a porcentagem enviada a cada ambiente.

> **Dica de prova:** Mandar X% do tráfego por DNS para outro ambiente: roteamento ponderado (weighted).

R2.4. A) Criar um plano de restore testing do AWS Backup com frequência, seleção de recursos e validação, que exclui os recursos restaurados

Por quê: Restore testing é o recurso do AWS Backup que avalia de forma automática e periódica a viabilidade de restauração: o plano define frequência e janela, os recursos protegidos entram por tipo, ARN ou tag, o serviço infere os metadados de restauração, monitora a duração e apaga os recursos de teste ao final. Os resultados servem como evidência de conformidade.

✗ B) Incorreta. A função funciona, mas é código próprio para manter, exatamente o que a equipe quer evitar, e a limpeza dos recursos fica por conta dela.

✗ C) Incorreta. Controles de frequência e retenção comprovam que os backups existem conforme a política, mas não executam restaurações nem medem o tempo de restauração.

✗ D) Incorreta. O Vault Lock impede a exclusão ou alteração dos backups (WORM), mas não prova que eles restauram nem mede o tempo de restauração.

> **Dica de prova:** Provar que o backup restaura, de forma agendada e sem script = AWS Backup restore testing.

03

CAPÍTULO 3 · 24% DA PROVA

Domínio 3: Criação de arquiteturas de alto desempenho

Aqui eu preciso escolher o armazenamento certo (tipos de EBS, instance store, EFS, família FSx, S3 e suas acelerações), o banco certo para cada padrão de acesso (DynamoDB com GSI e DAX, Aurora, Redshift, Neptune, DocumentDB, Keyspaces, OpenSearch), a camada de cache, o balanceador e a computação que escala sozinha. Vale 24% e cai como sintoma de lentidão com uma restrição de custo ou de esforço: o enunciado descreve o gargalo e a resposta é o serviço feito para ele. Onde mais erro é confundir CloudFront com Global Accelerator e Transfer Acceleration, achar que réplica ou RDS Proxy escala gravação e esquecer que na Lambda a CPU vem junto com a memória. O hábito que acerta: identificar se o gargalo é leitura, gravação, IOPS, throughput, latência de rede ou CPU antes de olhar as alternativas.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Tipos de volume EBS

gp3 entrega 3.000 IOPS e 125 MiB/s de base e permite provisionar IOPS e throughput sem aumentar o tamanho (até 80.000 IOPS por volume). gp2 só ganha desempenho crescendo (3 IOPS por GiB) e para em 250 MiB/s. io2 Block Express dá latência abaixo de 1 ms e durabilidade de 99,999%. st1 é HDD para I/O sequencial grande e sc1 é HDD frio; nenhum dos dois serve a banco transacional.

Instance store NVMe

Armazenamento local da instância, o mais rápido para scratch, cache e arquivos intermediários. Os dados não persistem a parada, hibernação ou recuperação da instância e não aceitam snapshot do EBS.

Amazon EFS e modos de throughput

Sistema de arquivos NFS regional, compartilhado entre instâncias de várias AZs e clientes Linux. Elastic Throughput escala sozinho com picos imprevisíveis; Provisioned cobra o valor fixado o tempo todo; em Bursting o throughput cresce com o tamanho armazenado. O modo de desempenho não pode ser trocado depois da criação e o Max I/O é de geração anterior.

Família Amazon FSx

FSx for Windows File Server: SMB, integração com AD, ACLs NTFS, shadow copies e Multi-AZ. FSx for Lustre: HPC e ML com latência abaixo de 1 ms, vinculado a bucket S3 por data repository association. FSx for NetApp ONTAP: mesmos arquivos por NFS e SMB, snapshots e tiering para o capacity pool.

AWS Storage Gateway

S3 File Gateway expõe NFS ou SMB no data center gravando objetos no S3 com cache local. Volume Gateway expõe iSCSI: cached mantém o primário no S3 e só o mais acessado localmente; stored mantém tudo local com backup assíncrono. Tape Gateway apresenta biblioteca de fitas virtuais.

Desempenho no S3

Leitura global de conteúdo estático = CloudFront e chaves espalhadas por vários prefixos. Objeto grande lido na mesma Região = byte-range fetches em paralelo. Upload de clientes distantes = Transfer Acceleration com multipart. Latência de um dígito de milissegundo para computação numa AZ = directory bucket do S3 Express One Zone.

Lambda: memória, CPU, /tmp e SnapStart

A CPU cresce na proporção da memória; em 1.769 MB a função tem o equivalente a um vCPU, então função limitada por CPU acelera ao aumentar a memória. O armazenamento efêmero /tmp vai até 10.240 MB e não cresce com a memória. SnapStart reduz o cold start de Java sem pagar ambientes ociosos, em versão publicada; simultaneidade provisionada garante latência, mas cobra o tempo todo.

Estratégias de cache

Lazy loading só atualiza o cache no cache miss; write-through atualiza a chave a cada gravação no banco; TTL equilibra os dois. Sorted sets do ElastiCache (Valkey ou Redis OSS) resolvem placar e ranking em tempo real; Memcached não tem estruturas ordenadas nem replica dados.

DynamoDB: índices, DAX e capacidade

GSI cria novo padrão de consulta em tabela existente, com outra chave de partição; LSI só na criação da tabela e com a mesma chave de partição. DAX é cache em memória com latência de microssegundos para leituras repetitivas. On-demand absorve tráfego imprevisível e warm throughput pré-aquece a tabela antes de um salto com data marcada.

Aurora para desempenho

Endpoints personalizados separam réplicas por tipo de carga (BI x aplicação); o reader endpoint usa todas. Aurora Auto Scaling adiciona réplicas conforme a carga de leitura. Aurora Serverless v2 escala em incrementos de meia ACU, inclusive a instância gravadora, sem trocar classe.

Analytics serverless e data lake

Athena cobra pelos bytes lidos: Parquet comprimido, partições nas colunas de filtro e bucketing nas de alta cardinalidade (reescritos com CTAS). Glue crawlers catalogam no Data Catalog e Lake Formation aplica permissão por linha e coluna com data filters. Quick Sight com SPICE evita que cada clique gere consulta no Athena.

Amazon Redshift

Data warehouse colunar com processamento paralelo massivo para SQL sobre bilhões de linhas. Redshift Spectrum consulta dados frios no S3 junto com tabelas locais por schema externo no Glue Data Catalog. Concurrency Scaling acrescenta capacidade nos picos de consultas sem redimensionar o cluster.

Streaming

Kinesis Data Streams retém o fluxo para vários consumidores lerem no próprio ritmo e reprocessarem; modo on-demand dispensa gerenciar shards; enhanced fan-out só aumenta leitura. Data Firehose entrega em lotes para S3, Redshift, OpenSearch e HTTP, não para RDS. MSK para Kafka, Managed Service for Apache Flink para processamento em tempo real e Kinesis Video Streams para vídeo de câmeras.

ALB, NLB e GWLB

ALB é camada 7: regras host-header e path-pattern, certificados por SNI, versão de protocolo gRPC, target groups ponderados. NLB é camada 4: TCP e UDP, milhões de requisições por segundo e Elastic IP por AZ. GWLB insere appliances de inspeção de forma transparente com GENEVE.

CloudFront x Global Accelerator

CloudFront é CDN: cache na borda, Origin Shield, CloudFront Functions para lógica leve por requisição e Lambda@Edge para lógica pesada. Global Accelerator usa a rede global da AWS com IPs estáticos, aceita UDP e faz failover rápido entre Regiões; não faz cache e seus endpoints são ALB, NLB, instâncias EC2 e Elastic IPs.

Placement groups

Cluster: instâncias próximas em uma AZ, baixa latência para HPC e MPI (mesmo tipo, uma solicitação). Spread: cada instância em hardware distinto, até sete em execução por AZ, para poucas instâncias críticas. Partition: grupos em racks separados para HDFS e Cassandra.

Famílias de instância EC2

C para CPU intensiva, R e X para memória (swap e paginação), M equilibrada, T para CPU baixa com picos, G para gráficos e inferência barata, P para treino pesado de ML. Banda de rede é atributo do tipo: instâncias Nitro com ENA, sufixo n para otimizadas para rede; mais ENIs não somam banda.

Escalabilidade automática

Target tracking em RequestCountPerTarget para web e em backlog por instância (mensagens visíveis divididas pelos consumidores) para workers de fila; CPU não reflete atraso na fila. Scheduled scaling para picos com hora marcada, warm pool para instâncias de inicialização longa e capacity provider com managed scaling no ECS em EC2.

Macetes para memorizar

Lambda

Memória é o pedal da CPU; /tmp é outro botão.

EBS

gp3 = 3 mil IOPS e 125 MiB/s já inclusos no preço; gp2 só cresce engordando.

FSx

Windows fala SMB, Lustre corre (HPC), ONTAP fala as duas línguas.

Bordas

CloudFront Cacheia, Global Accelerator Acelera o caminho, Transfer Acceleration Transporta upload.

Placement groups

Cluster junta, Spread separa (7 por AZ), Partition divide em racks.

Famílias EC2

C de CPU, R de RAM, M de meio-termo, T de tranquilo com picos, G de gráfico, P de pesado em ML.

Bancos especializados

Grafo = Neptune, Mongo = DocumentDB, Cassandra = Keyspaces, busca de texto = OpenSearch.

Tabela de memorização: não confunda

Quando é...	...e quando é
CloudFront Conteúdo HTTP cacheável, muitos espectadores pedindo o mesmo objeto, origem S3 ou ALB.	**Global Accelerator** API dinâmica global, UDP ou TCP não HTTP, IPs estáticos e failover rápido entre Regiões.
ALB Roteamento por caminho ou host, gRPC, vários microsserviços atrás de um balanceador com SNI.	**NLB** TCP/UDP, Elastic IP fixo por AZ, milhões de requisições por segundo ou protocolo proprietário.

gp3 Precisa de mais IOPS ou throughput sem pagar mais GiB, até 80.000 IOPS.	**io2 Block Express** Mais de 80.000 IOPS, latência consistente abaixo de 1 ms e durabilidade de 99,999% (OLTP pesado).
GSI Novo padrão de consulta por outro atributo em tabela já existente.	**LSI** Ordenar de outro jeito dentro da mesma chave de partição, definido na criação da tabela.
Lazy loading Cachear só o que é lido, tolerando dado um pouco desatualizado até o TTL.	**Write-through** Dado que nunca pode ficar velho no cache, como preço; atualiza a cada gravação.
S3 Transfer Acceleration Upload de clientes distantes para um único bucket, com multipart.	**CloudFront** Entrega global de leitura com cache nos pontos de presença.
Kinesis Data Streams Vários consumidores em tempo real, retenção e replay do fluxo.	**Data Firehose** Entrega gerenciada em lote para S3, Redshift, OpenSearch ou HTTP, sem código de consumo.

Números que caem

Item	Valor
Desempenho de base do gp3	**3.000 IOPS e 125 MiB/s**
IOPS máximas do gp3	**80.000 por volume no EC2; 64.000 no RDS**
gp2	**3 IOPS por GiB; throughput máximo de 250 MiB/s**
Durabilidade io2 Block Express x gp3	**99,999% x 99,8% a 99,9%**
Memória e ephemeral storage máximos da Lambda	**10.240 MB**
Memória da Lambda equivalente a um vCPU	**1.769 MB**
Placement group spread	**Até sete instâncias em execução por AZ**
Registro do Kinesis Data Streams	**Até 1 MB**
Escala do Aurora Serverless v2	**Incrementos de meia ACU**
Velocidades de conexão dedicada do Direct Connect	**1, 10, 100 ou 400 Gbps**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Usar simultaneidade provisionada ou mais /tmp para acelerar uma Lambda limitada por CPU.
Como não cair: Provisionada só elimina cold start e ephemeral storage é disco. CPU vem com a memória: aumente a memória.

⚠ **Alerta de prova:** Aumentar a memória da Lambda para ganhar espaço em disco para arquivos de vários GB.
Como não cair: Memória não amplia /tmp; configure o ephemeral storage até 10.240 MB. Armazenamento compartilhado ou persistente = EFS.

⚠ **Alerta de prova:** Partir para io2 porque o gp3 não provisionaria IOPS acima da base, ou crescer o gp2 para ganhar IOPS.
Como não cair: gp3 provisiona IOPS e throughput independentes do tamanho; gp2 crescido paga GiB sem uso e para em 250 MiB/s. io2 só quando precisa de mais de 80.000 IOPS ou latência abaixo de 1 ms.

⚠ **Alerta de prova:** Trocar o EFS para Max I/O para ganhar throughput.
Como não cair: O modo de desempenho não muda após a criação e Max I/O é legado com latência maior. Picos imprevisíveis = Elastic Throughput.

⚠ **Alerta de prova:** Adicionar réplicas de leitura ou RDS Proxy para resolver gargalo de gravação.
Como não cair: Réplicas aceitam só leitura e o Proxy só reaproveita conexões. Escrita = scale up da instância, io2 no armazenamento, Aurora Serverless v2 ou DynamoDB.

⚠ **Alerta de prova:** Criar LSI na tabela existente para consultar por outro atributo de partição.
Como não cair: LSI só nasce com a tabela e mantém a chave de partição. Use GSI.

⚠ **Alerta de prova:** Habilitar Transfer Acceleration para acelerar downloads de instâncias na mesma Região ou para servir vídeos a usuários globais.
Como não cair: Transfer Acceleration encurta distância pela internet e não faz cache. Mesma Região = byte-range fetches; entrega global = CloudFront.

⚠ **Alerta de prova:** Colocar Global Accelerator na frente de bucket S3, endpoint do RDS ou túneis de VPN.
Como não cair: Endpoints do Global Accelerator são ALB, NLB, instâncias EC2 e Elastic IPs. Leitura distante do RDS = réplica cross-Region.

⚠ **Alerta de prova:** Anexar mais interfaces de rede ou EFA a uma instância antiga para somar banda.
Como não cair: A banda é definida pelo tipo de instância. Migre para Nitro com ENA, de preferência família otimizada para rede.

⚠ **Alerta de prova:** Usar Data Firehose para gravar o stream processado direto no RDS.
Como não cair: RDS não é destino do Firehose. Processamento em tempo real com gravação no RDS = Managed Service for Apache Flink.

⚠ **Alerta de prova:** Ativar aceleração na VPN existente terminada em virtual private gateway.
Como não cair: VPN acelerada só em conexão nova anexada a Transit Gateway. Mais banda de VPN = várias VPNs com BGP e ECMP no Transit Gateway.

⚠ **Alerta de prova:** Usar placement group spread ou partition para baixar a latência entre nós MPI.
Como não cair: Spread e partition isolam falhas. Baixa latência entre nós = placement group cluster em uma AZ.

⚠ **Alerta de prova:** Escalar workers de fila pela CPU média.
Como não cair: A CPU pode ficar baixa com o backlog crescendo. Target tracking em backlog por instância, com alvo derivado do SLA.

⚠ **Alerta de prova:** Montar VPC peering com uma VPC de serviços compartilhados esperando alcançar as outras VPCs através dela.
Como não cair: Peering não é transitivo. Dezenas de VPCs + on-premises em hub-and-spoke = Transit Gateway.

Como cai na prova

📝 **Como cai:** Banco relacional lento; alternativas misturam réplicas, cache, instância maior e troca de serviço.
Como resolver: 1) Gargalo de leitura repetida = cache (ElastiCache, DAX) ou réplicas. 2) Gargalo de gravação = scale up, io2 ou Serverless v2; réplica não ajuda. 3) Analítico pesado em banco transacional = Redshift ou Athena. 4) Acesso por chave em escala massiva = DynamoDB, mantendo joins no relacional.

📝 **Como cai:** Usuários globais reclamam de latência; alternativas oferecem CloudFront, Global Accelerator, Transfer Acceleration e outra Região.
Como resolver: 1) Conteúdo cacheável = CloudFront (Origin Shield para muitos espectadores). 2) Dinâmico, UDP ou IP fixo = Global Accelerator. 3) Upload distante = Transfer Acceleration. 4) Leitura de banco em outro continente = réplica cross-Region ou global tables.

📝 **Como cai:** Escolher armazenamento de arquivos por protocolo e carga.
Como resolver: Linux compartilhado multi-AZ = EFS; Windows com AD = FSx for Windows; NFS + SMB = ONTAP; HPC sobre S3 = FSx for Lustre; on-premises com cache para S3 = S3 File Gateway; bloco iSCSI = Volume Gateway.

📝 **Como cai:** Escolher a métrica e o tipo de escalabilidade para uma frota.
Como resolver: Web atrás de ALB = target tracking em RequestCountPerTarget; fila = backlog por instância; evento com hora marcada = scheduled; boot longo com picos imprevisíveis = warm pool; padrão recorrente = predictive; ECS em EC2 sem espaço para tarefas = capacity provider com managed scaling.

📝 **Como cai:** Ingestão contínua de eventos com vários consumidores e reprocessamento.
Como resolver: 1) Precisa reler o histórico e vários consumidores independentes = Kinesis Data Streams ou MSK. 2) Só entregar no S3 para análise = Firehose + Athena. 3) Processamento em tempo real gerenciado = Managed Service for Apache Flink. 4) Dado que chega uma vez por dia = lote (S3 + Glue), não streaming.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
arquivos compartilhados entre instâncias de várias AZs, Linux	**Amazon EFS.**
SMB com AD e NTFS / mesmos arquivos por NFS e SMB / HPC lendo dados do S3	**FSx for Windows File Server / FSx for NetApp ONTAP / FSx for Lustre.**
servidores on-premises gravando por NFS ou SMB no S3 com cache local	**S3 File Gateway.**
leituras repetitivas no DynamoDB em microssegundos	**DAX.**
placar ou ranking em tempo real	**Sorted set no ElastiCache for Valkey.**
SQL analítico sobre bilhões de linhas / dados frios no S3 na mesma consulta	**Amazon Redshift / Redshift Spectrum.**
Athena lento e caro com CSV	**Parquet comprimido, partições e bucketing reescritos com CTAS.**
permissão por linha ou coluna no data lake	**AWS Lake Formation com data filters.**
regra por país (lei, licença)	**Route 53 geolocation com registro padrão.**
cold start de Java sem pagar ambiente ocioso	**Lambda SnapStart.**
lógica leve em toda requisição na borda (redirect, cabeçalho, chave de cache)	**CloudFront Functions com KeyValueStore.**
processar dentro da fábrica com APIs da AWS, dados não podem sair	**AWS Outposts.**

fraude ou recomendação com relacionamentos de vários saltos	**Amazon Neptune.**
manter drivers Cassandra / drivers MongoDB / PostGIS	**Amazon Keyspaces / Amazon DocumentDB / RDS for PostgreSQL com a extensão.**

🎯 **Se você lembrar de só uma coisa deste capítulo...**

- Réplica de leitura e RDS Proxy não escalam gravação.
- CloudFront faz cache; Global Accelerator e Transfer Acceleration não.
- Na Lambda, CPU vem com a memória; /tmp é configuração separada.
- Workers de fila escalam por backlog por instância, não por CPU.

Essenciais para a prova (checklist)

☐ Sei escolher entre gp3, gp2, io2 Block Express, st1, sc1 e instance store pelo padrão de I/O.

☐ Sei escolher entre EFS, FSx for Windows, FSx for Lustre, FSx for NetApp ONTAP e Storage Gateway pelo protocolo.

☐ Sei acelerar o S3 com CloudFront, prefixos, byte-range fetches, Transfer Acceleration ou Express One Zone.

☐ Sei a diferença entre GSI e LSI e quando usar DAX ou ElastiCache.

☐ Sei otimizar Athena com Parquet, partições e bucketing e quando usar Redshift, Spectrum e Concurrency Scaling.

☐ Sei diferenciar Kinesis Data Streams, Firehose, MSK, Flink e Kinesis Video Streams.

☐ Sei escolher entre ALB, NLB e GWLB e entre CloudFront e Global Accelerator.

☐ Sei escolher o placement group e a família de instância pelo gargalo.

☐ Sei configurar target tracking com a métrica certa e quando usar warm pool ou scheduled scaling.

Questões de revisão do capítulo

R3.1 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma fintech tem uma função Lambda em Python configurada com 512 MB que calcula, para cada proposta recebida, uma simulação de crédito com laços numéricos intensos. As métricas mostram uso de memória de 180 MB, duração média de 9 segundos e nenhuma espera por rede ou banco de dados. O time de produto quer reduzir a duração para menos de 3 segundos sem reescrever o algoritmo. Qual ação atende a esse requisito?

A. Aumentar o armazenamento efêmero /tmp da função para 10.240 MB para que o runtime tenha mais espaço de trabalho durante os cálculos
B. Aumentar a memória configurada da função, por exemplo para 1.769 MB ou mais, já que a CPU alocada pela Lambda cresce na mesma proporção
C. Reduzir a memória da função para 256 MB, já que o uso medido é baixo, e aumentar o timeout para 30 segundos para evitar falhas por tempo esgotado
D. Configurar simultaneidade provisionada na função para manter ambientes de execução já inicializados e eliminar o tempo de cold start das invocações

R3.2 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma transportadora processa toda noite um lote de otimização de rotas em 40 instâncias EC2 que leem e gravam intensamente em um Amazon FSx for Lustre de implantação em uma única AZ (us-east-1a). Se a AZ falhar, o lote pode ser executado de novo no dia seguinte, mas a camada web de rastreamento, que roda em um Auto Scaling group, precisa continuar disponível. A empresa quer o melhor desempenho do lote e evitar cobranças de transferência entre AZs. Qual solução atende a esses requisitos?

A. Lançar os workers do lote em um placement group spread em us-east-1a para que cada instância fique em um rack distinto e tenha mais largura de banda
B. Distribuir os workers do lote entre três AZs para que uma falha de zona não interrompa o processamento, mantendo o sistema de arquivos FSx for Lustre em us-east-1a
C. Colocar os workers do lote e as instâncias da camada web em um mesmo placement group cluster para reduzir a latência entre todos os componentes
D. Executar os workers do lote em sub-redes de us-east-1a, junto do sistema de arquivos, e manter o Auto Scaling group da camada web distribuído em várias AZs

R3.3 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma empresa de jogos online está projetando a arquitetura de backend para um jogo multiplayer em tempo quase real. O tráfego é altamente variável, com picos imprevisíveis durante lançamentos de eventos sazonais. A aplicação é stateless e será implantada em múltiplas zonas de disponibilidade. O arquiteto de soluções precisa garantir alta performance, escalabilidade automática e custo otimizado, além de minimizar a complexidade operacional. Qual abordagem é MAIS adequada?

A. Executar a aplicação no Amazon ECS com capacity provider do EC2 em instâncias reservadas dimensionadas para o pico dos eventos
B. Hospedar a aplicação no Amazon ECS com AWS Fargate em várias AZs, com Service Auto Scaling por métricas e um Application Load Balancer
C. Executar a aplicação em um cluster Amazon EKS com nós gerenciados, mantendo o Cluster Autoscaler e expondo o serviço por um NLB
D. Implantar a aplicação em instâncias Amazon EC2 em Auto Scaling Group com políticas de scaling baseadas em CPU, atrás de um Application Load Balancer, utilizando instâncias sob demanda

R3.4 · Domínio 3: Criação de arquiteturas de alto desempenho

Uma aplicação de análise de dados em tempo quase real recebe picos de até 200.000 mensagens por segundo provenientes de dispositivos IoT e precisa processá-las com baixa latência, além de permitir reprocessamento em caso de falhas. O time de arquitetura deseja uma solução gerenciada, altamente escalável e com capacidade de retenção de dados para reprocessamento. Qual opção de arquitetura é MAIS adequada para esse cenário?

A. Utilizar Amazon SNS para publicar as mensagens e múltiplas filas SQS como assinantes, cada uma processando uma cópia da mensagem em paralelo.
B. Usar Kinesis Data Streams com consumidores KCL escalando por shards
C. Utilizar um Application Load Balancer (ALB) distribuindo requisições HTTP diretamente para um grupo de Auto Scaling de instâncias EC2 que processam e gravam os dados em um banco de dados relacional.
D. Usar SQS com Auto Scaling de instâncias EC2 que consomem e processam as mensagens na hora

Respostas comentadas

R3.1. B) Aumentar a memória configurada da função, por exemplo para 1.769 MB ou mais, já que a CPU alocada pela Lambda cresce na mesma proporção

Por quê: Na Lambda a memória é a principal alavanca de desempenho: a CPU é alocada em proporção à memória configurada (de 128 MB a 10.240 MB), e em 1.769 MB a função tem o equivalente a um vCPU. Uma função CPU-bound que usa pouca RAM fica lenta por falta de CPU; aumentar a memória reduz a duração e, como a cobrança é por GB-segundo, o custo pode até cair. O AWS Lambda Power Tuning e o Compute Optimizer ajudam a escolher o valor.

✗ A) Incorreta. O armazenamento efêmero amplia o espaço em disco de /tmp, não a CPU; a função não está limitada por disco.

✗ C) Incorreta. Com menos memória a função recebe ainda menos CPU e fica mais lenta; aumentar o timeout só evita o erro, não reduz a duração.

✗ D) Incorreta. A simultaneidade provisionada elimina a latência de inicialização, mas a duração de 9 segundos vem do cálculo em si, limitado pela pouca CPU disponível.

> **Dica de prova:** Lambda lenta por CPU: aumente a memória, porque a CPU vem junto. Simultaneidade provisionada resolve cold start, não falta de CPU.

R3.2. D) Executar os workers do lote em sub-redes de us-east-1a, junto do sistema de arquivos, e manter o Auto Scaling group da camada web distribuído em várias AZs

Por quê: A posição dos recursos é uma troca entre desempenho e disponibilidade: colocar a computação na mesma AZ dos dados reduz latência e elimina a cobrança de tráfego entre AZs, mas cria dependência da zona. Isso serve para cargas que podem ser repetidas, como lotes; cargas que precisam de alta disponibilidade continuam distribuídas em várias AZs. Sistemas FSx for Lustre com armazenamento SSD ou HDD ficam em uma única AZ, então clientes em outras zonas pagam transferência entre AZs e têm mais latência.

✗ A) Incorreta. O placement group spread separa poucas instâncias em hardware distinto para reduzir falhas correlacionadas; ele não aumenta a largura de banda e limita o número de instâncias por AZ.

✗ B) Incorreta. Os workers fora de us-east-1a teriam mais latência e pagariam transferência entre AZs, e o processamento pararia de qualquer forma se a AZ do sistema de arquivos falhasse.

✗ C) Incorreta. O placement group cluster fica em uma única AZ; colocar nele a camada web eliminaria a alta disponibilidade exigida para o rastreamento.

> **Dica de prova:** Lote repetível e sensível a latência: computação na mesma AZ dos dados. Camada crítica: várias AZs.

R3.3. B) Hospedar a aplicação no Amazon ECS com AWS Fargate em várias AZs, com Service Auto Scaling por métricas e um Application Load Balancer

Por quê: Com o AWS Fargate, o ECS executa cada tarefa sem que a equipe provisione ou corrija servidores, e o Service Auto Scaling ajusta o número de tarefas por métricas como CPU ou requisições por target. Distribuído em várias AZs atrás de um ALB, o serviço absorve picos imprevisíveis pagando pelos recursos das tarefas em execução. EC2 e EKS também escalam, mas acrescentam gerenciamento de instâncias ou de cluster.

✗ A) Incorreta. Reservar instâncias para o pico paga capacidade ociosa fora dos eventos e ainda deixa a equipe gerenciando instâncias e o escalonamento do cluster.

✗ C) Incorreta. O EKS funciona, mas exige operar Kubernetes, nós e o Cluster Autoscaler, o que contraria o requisito de minimizar a complexidade operacional.

✗ D) Incorreta. Embora o Auto Scaling Group com ALB ofereça escalabilidade e alta disponibilidade, a necessidade de gerenciar instâncias EC2, AMIs, patches e capacidade é maior.

> **Dica de prova:** Quando o enunciado enfatiza picos imprevisíveis, alta escalabilidade e desejo de reduzir a complexidade operacional, pense em serviços serverless ou gerenciados (como Fargate, Lambda, DynamoDB). Quando o foco recai em controle fino de infraestrutura, normalmente a resposta tende a EC2 ou Kubernetes (EKS).

R3.4. B) Usar Kinesis Data Streams com consumidores KCL escalando por shards

Por quê: O Amazon Kinesis Data Streams ingere registros em alta taxa com baixa latência, mantém os dados por um período de retenção configurável e permite que os consumidores releiam o fluxo a partir de um sequence number ou de um horário, o que viabiliza o reprocessamento. A capacidade cresce com o número de shards, e os consumidores KCL distribuem os shards entre os workers. SQS e SNS não guardam o histórico depois da entrega.

✗ A) Incorreta. A combinação SNS + SQS é adequada para fan-out e desacoplamento, mas não é ideal para cenários de streaming em alta taxa com necessidade de reprocessamento temporal e consumo paralelo ordenado.

✗ C) Incorreta. Usar diretamente um ALB com instâncias EC2 para receber todas as requisições não é a melhor escolha para ingestão massiva de eventos de IoT.

✗ D) Incorreta. O SQS escala bem, mas cada mensagem é entregue para processamento e excluída depois; não há como reler os dados a partir de um ponto no tempo nem ter vários consumidores lendo o mesmo fluxo, o que impede o reprocessamento pedido.

> **Dica de prova:** Streaming em alta taxa com retenção e replay: Kinesis Data Streams. Fila de tarefas: SQS. Fan-out de notificações: SNS.

04

CAPÍTULO 4 · 20% DA PROVA

Domínio 4: Criação de arquiteturas com custo otimizado

Aqui eu preciso casar cada perfil de uso com o modelo de compra certo (On-Demand, Savings Plans, Reserved Instances, Spot), escolher a classe de armazenamento do S3 pela frequência de acesso e cortar custos de rede (NAT gateway, transferência entre AZs, saída para a internet, IPv4 público). Vale 20% e quase sempre traz um qualificador como MAIS econômica sem perder disponibilidade ou sem mudar a aplicação. Onde mais erro é comprar compromisso para carga que roda poucas horas, mandar objeto pequeno ou de vida curta para Standard-IA e esquecer que Compute Savings Plans não cobrem RDS. A régua que acerta: base estável 24x7 com compromisso, pico com On-Demand, interrompível com Spot, ocioso desligado.

Neste capítulo

17 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Classes do S3

Standard para acesso frequente; Standard-IA e One Zone-IA cobram por GB recuperado, mínimo de 128 KB por objeto e 30 dias; One Zone-IA guarda em uma AZ só. Glacier Instant Retrieval lê em milissegundos com mínimo de 90 dias; Glacier Flexible Retrieval exige restauração de minutos a horas; Glacier Deep Archive é a mais barata, recuperação em até 48 horas e mínimo de 180 dias.

S3 Intelligent-Tiering x lifecycle

Intelligent-Tiering move objetos entre níveis conforme o acesso e serve a padrão desconhecido ou variável, mas cobra monitoramento por objeto; cada leitura devolve o objeto ao nível frequente e são precisos 90 dias seguidos sem acesso para voltar ao nível Archive Instant Access. Com padrão por idade conhecido, regras de lifecycle saem mais baratas.

Regras de lifecycle além da transição

Expiração automática, AbortIncompleteMultipartUpload para partes de uploads incompletos, NoncurrentVersionExpiration para versões antigas e filtros por prefixo ou tag de objeto. Suspender o versionamento não remove versões já existentes.

Savings Plans

Compute Savings Plans cobrem EC2 de qualquer família e Região, Fargate e Lambda, com desconto menor. EC2 Instance Savings Plans prendem família e Região (até 72% de desconto), permitindo mudar tamanho e sistema operacional. Database Savings Plans cobrem RDS e Aurora mesmo trocando motor, família ou Região. O compromisso por hora é cobrado em todas as horas do prazo.

Reserved Instances e reservas

RI Standard dá grande desconto preso aos atributos (zonal: tipo, tamanho, plataforma e AZ); RI conversível permite trocar atributos por exchange, com desconto menor. Instância reservada do RDS fica presa a Região, motor e tipo e não pode ser cancelada. Capacity Reservation garante capacidade, mas cobra a tarifa On-Demand.

Instâncias Spot

Maior desconto para cargas tolerantes a interrupção, com aviso de 2 minutos (evento no EventBridge ou metadados). Estratégia recomendada price-capacity-optimized com vários tipos e AZs; preço máximo alto não impede interrupção. Trabalho longo exige checkpoint, como estado salvo no S3.

Mixed instances policy

Um Auto Scaling group com base On-Demand (coberta por Savings Plans ou RIs) e excedente em Spot, diversificando famílias e tamanhos. É o padrão para carga com base estável e picos tolerantes a interrupção.

Rightsizing e famílias

AWS Compute Optimizer recomenda tipo e tamanho; Cost Explorer recomenda Savings Plans para a carga estável. CPU alta e memória sobrando = família C; memória alta e CPU sobrando = R; CPU média baixa com picos curtos = T (créditos de CPU); melhor preço-desempenho = Graviton. Ajuste o tamanho antes de reservar.

Desligar o que está ocioso

Ações agendadas no Auto Scaling group com mínimo e desejado em 0 fora do horário; parar instâncias de um ASG por fora faz o grupo substituí-las. Instance Scheduler ou Systems Manager param e iniciam instâncias preservando EBS. Hibernação preserva a memória, mas só pode ser habilitada no lançamento, com volume raiz EBS criptografado.

Custos de NAT e endpoints

Tráfego de sub-rede privada para S3 e DynamoDB pelo gateway endpoint não paga processamento de NAT. Dev com pouco tráfego e sem exigência de HA aceita instância NAT pequena (verificação de origem e destino desligada) ou um único NAT gateway compartilhado; produção usa NAT gateway por AZ ou regional.

Transferência de dados

CloudFront na frente de ALB, EC2 ou S3 reduz o custo de saída para a internet. NLB com cross-zone ligado cobra transferência entre AZs. VPC peering não cobra por GB processado, ao contrário do Transit Gateway. Direct Connect tem taxa de saída reduzida para muitos TB por mês. Todo IPv4 público é cobrado por hora desde fevereiro de 2024, inclusive Elastic IP associado ou ocioso.

Custos do DynamoDB

Leitura eventualmente consistente consome metade da fortemente consistente, e a transacional consome o dobro. Tráfego estável e previsível sai mais barato em modo provisionado com auto scaling e capacidade reservada (só no provisionado da classe Standard); imprevisível ou novo, on-demand. TTL apaga itens vencidos sem consumir escrita.

Custos do Aurora e do RDS

Aurora I/O-Optimized vale quando o I/O representa 25% ou mais do gasto do Aurora. Aurora Serverless v2 com mínimo de 0 ACU pausa sozinho em ociosidade. Clone do Aurora é copy-on-write, rápido e barato para testes. Storage auto scaling do RDS evita superdimensionar, e o disco do RDS não encolhe. Retenção longa para consulta eventual = exportar snapshot para o S3 em Parquet e consultar no Athena.

Ferramentas de gestão de custos

AWS Budgets com budget actions bloqueia ou para recursos automaticamente; alarmes de faturamento e Cost Anomaly Detection só avisam. CUR 2.0 pelo AWS Data Exports em Parquet no S3, consultado no Athena, dá SQL no nível de linha. Cost Explorer dá visão agregada. Organizations com faturamento consolidado compartilha descontos de Savings Plans e RIs entre contas.

API Gateway HTTP API x REST API

HTTP API é mais barata e valida JWT nativamente; REST API custa mais e só se justifica quando precisa de API keys, usage plans, cache, validação de requisição ou mapping templates.

EBS e Route 53 que economizam

gp2 para gp3 com Elastic Volumes reduz custo sem perda de desempenho de base e sem downtime. sc1 é o bloco mais barato para dados sequenciais raramente acessados; st1 é para sequencial frequente. Registro alias do Route 53 aponta para recursos da AWS no domínio raiz, onde CNAME não é permitido.

Licenças e hardware dedicado

Licença por soquete ou núcleo trazida pelo cliente (BYOL) = Dedicated Hosts com AWS License Manager. Dedicated Instances só dão isolamento de hardware, sem visibilidade de soquetes. Para Oracle para PostgreSQL, DMS Schema Conversion converte esquema e código e o AWS DMS migra dados com CDC.

Macetes para memorizar

💡 Modelos de compra

Base = compromisso, Pico = On-Demand, Pode cair = Spot, Parado = desligado.

💡 Mínimos do S3

IA = 128 KB e 30 dias; Instant = 90; Deep = 180 e até 48 horas.

💡 Savings Plans

Compute é Coringa (EC2, Fargate, Lambda); Instance é Fixo na família; Database cuida do banco.

💡 DynamoDB

Eventual = meia leitura, Forte = uma, Transacional = duas.

💡 Rede barata

S3 e DynamoDB pelo portão (gateway endpoint), internet pela CDN, VPCs vizinhas por peering.

💡 Aurora

I/O chegou a um quarto da conta? I/O-Optimized. Ficou ocioso? Zero ACU.

💡 Rightsizing

Primeiro emagrece (rightsizing), depois casa (reserva).

Tabela de memorização: não confunda

Quando é...	...e quando é
Compute Savings Plans Pode trocar de família ou Região, ou mover carga para Fargate e Lambda.	**EC2 Instance Savings Plans** Família e Região fixas, mudando só tamanho ou sistema operacional; maior desconto.
S3 Intelligent-Tiering Padrão de acesso desconhecido ou que varia por objeto, com leitura rápida quando necessário.	**Regras de lifecycle** Padrão por idade conhecido (Standard por 30 dias, IA até 180, depois Deep Archive e expiração).

S3 Glacier Instant Retrieval Acesso raro, cerca de uma vez por trimestre, com leitura em milissegundos.	**S3 Glacier Deep Archive** Retenção de anos com recuperação em horas aceitável, menor custo.
NAT gateway Saída IPv4 para a internet (pacotes, repositórios) de sub-redes privadas.	**Gateway endpoint** Tráfego privado para S3 e DynamoDB, sem custo de processamento de NAT.
VPC peering Poucas VPCs com muito tráfego e sem necessidade de trânsito, sem taxa por GB processado.	**Transit Gateway** Muitas VPCs ou on-premises em hub-and-spoke, aceitando cobrança por anexo e por GB.
DynamoDB on-demand Tráfego imprevisível ou tabela nova.	**DynamoDB provisionado com auto scaling** Curva diária estável e previsível, com capacidade reservada na base.
Fargate com Compute Savings Plan Carga variável ou equipe que não quer operar servidores.	**ECS em EC2 com EC2 Instance Savings Plan** Contêiner 24x7 estável, equipe que aceita gerenciar instâncias, com binpack para consolidar.

Números que caem

Item	Valor
Standard-IA e One Zone-IA	**Mínimo de 128 KB por objeto e 30 dias**
Glacier Instant Retrieval	**Mínimo de 128 KB por objeto e 90 dias**
Glacier Deep Archive	**Mínimo de 180 dias; recuperação em até 48 horas**
Intelligent-Tiering: volta ao Archive Instant Access após uma leitura	**90 dias seguidos sem acesso**
Aviso de interrupção Spot	**2 minutos**
Desconto máximo do EC2 Instance Savings Plans	**Até 72%**
Consumo de leitura do DynamoDB	**Eventual = metade da forte; transacional = dobro da forte**
Quando usar Aurora I/O-Optimized	**I/O em 25% ou mais do gasto do Aurora**
Aurora Serverless v2 com auto-pause	**Capacidade mínima de 0 ACU**
Cobrança de IPv4 público	**Por hora, desde fevereiro de 2024**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Mandar miniaturas de 15 KB que vivem 10 dias para Standard-IA, One Zone-IA ou Glacier Instant Retrieval.
Como não cair: Essas classes cobram no mínimo 128 KB por objeto e 30 dias (90 no Glacier IR). Objeto pequeno ou de vida curta fica no Standard.

⚠ **Alerta de prova:** Escolher Intelligent-Tiering quando o padrão de acesso por idade já é conhecido.
Como não cair: O Intelligent-Tiering só acrescenta taxa de monitoramento; lifecycle por idade é mais barato e ainda expira os dados.

⚠ **Alerta de prova:** Usar Glacier Flexible Retrieval com recuperação Expedited para abrir arquivos em milissegundos.
Como não cair: Expedited leva minutos e exige restaurar antes. Milissegundos com acesso raro = Glacier Instant Retrieval.

⚠ **Alerta de prova:** Guardar a única cópia de registros legais ou originais em One Zone-IA.
Como não cair: One Zone-IA perde os dados se a AZ for perdida. Use só para dado reproduzível; cópia única vai para classe multi-AZ.

⚠ **Alerta de prova:** Comprar Savings Plan ou RI para carga que roda poucas horas por mês ou só no horário comercial.
Como não cair: O compromisso é cobrado em todas as horas do prazo. Uso curto e previsível = On-Demand só na janela (scheduled scaling) ou desligamento.

⚠ **Alerta de prova:** Cobrir Fargate com EC2 Instance Savings Plan ou bancos RDS com Compute Savings Plan.
Como não cair: Fargate e Lambda só com Compute Savings Plans; RDS e Aurora com Database Savings Plans ou instância reservada do RDS.

⚠ **Alerta de prova:** Definir o preço máximo do Spot igual ao On-Demand para evitar interrupções.
Como não cair: A AWS interrompe quando precisa da capacidade, independentemente do preço máximo. Use checkpoint, aviso de 2 minutos e diversificação.

⚠ **Alerta de prova:** Parar a instância existente e habilitar a hibernação.
Como não cair: Hibernação só é habilitada no lançamento; lance uma nova instância com volume raiz EBS criptografado com espaço para a RAM.

⚠ **Alerta de prova:** Aurora Serverless v2 com mínimo de 0,5 ACU para pausar à noite.
Como não cair: Só com mínimo de 0 ACU há auto-pause; acima de zero, a capacidade continua cobrada.

⚠ **Alerta de prova:** Criar RDS com 2 TB e reduzir o armazenamento depois.
Como não cair: O armazenamento alocado do RDS não encolhe. Comece pequeno com storage auto scaling e limite máximo.

⚠ **Alerta de prova:** Trocar IPv4 públicos automáticos por Elastic IPs ou guardar EIPs não associados por serem grátis.
Como não cair: Desde fevereiro de 2024 todo IPv4 público é cobrado por hora, associado ou não. Instâncias privadas atrás do ALB e liberar EIPs ociosos (Public IP Insights do IPAM).

⚠ **Alerta de prova:** Agrupar leituras em TransactGetItems para economizar capacidade no DynamoDB.
Como não cair: Leitura transacional consome o dobro da forte. Para economizar, leitura eventual (metade).

⚠ **Alerta de prova:** Configurar só notificação do Budgets ou Cost Anomaly Detection para bloquear gastos automaticamente.
Como não cair: Avisos não agem. Bloqueio automático sem código = budget actions aplicando política IAM ou parando instâncias.

⚠ **Alerta de prova:** Usar Global Accelerator ou NLB com Elastic IPs para reduzir custo de saída de aplicação HTTP.
Como não cair: Global Accelerator acrescenta taxa por hora e Data Transfer-Premium; NLB não muda a tarifa de saída. HTTP cacheável sem IP fixo = CloudFront.

Como cai na prova

📝 **Como cai:** Carga com perfil de uso descrito (24x7, horário comercial, poucas horas por mês, tolerante a interrupção) e pede o menor custo.
Como resolver: 1) Separe base estável, picos e partes interrompíveis. 2) Base 24x7 por anos = Savings Plans ou RI. 3) Picos curtos = On-Demand. 4) Interrompível com checkpoint = Spot. 5) Uso em janela = scheduled scaling ou desligamento; nunca compromisso para carga ociosa.

📝 **Como cai:** Dados no S3 com faixas de acesso por idade e exigência de tempo de recuperação.
Como resolver: 1) Mapeie cada faixa para uma classe: frequente = Standard; raro com milissegundos = Glacier IR; anos com horas = Deep Archive. 2) Confira os mínimos de tamanho e duração. 3) Padrão conhecido = lifecycle; desconhecido = Intelligent-Tiering. 4) Exigência de resiliência elimina One Zone-IA.

📝 **Como cai:** Conta de rede alta: NAT gateway, transferência entre AZs, Transit Gateway ou IPv4.
Como resolver: 1) Tráfego para S3/DynamoDB = gateway endpoint. 2) Saída HTTP para usuários = CloudFront. 3) Duas VPCs com muito tráfego = peering. 4) NLB com cross-zone = desligar e dimensionar cada AZ. 5) Dev = um NAT compartilhado ou instância NAT.

📝 **Como cai:** Escolher o tipo de Savings Plan ou reserva com requisito de flexibilidade futura.
Como resolver: 1) Vai mudar família, Região ou ir para Fargate/Lambda = Compute Savings Plans. 2) Família e Região fixas = EC2 Instance Savings Plans. 3) Banco que vai trocar de motor = Database Savings Plans. 4) Configuração fixa no RDS = instância reservada do RDS.

📝 **Como cai:** Ambiente de desenvolvimento caro com a mesma arquitetura da produção.
Como resolver: Dev tolerante a falha = uma AZ, Spot, RDS Single-AZ, um NAT gateway e desligamento fora do horário; produção continua multi-AZ. Não aplique a mesma régua de disponibilidade aos dois.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
quem baixa os dados deve pagar a transferência	**S3 Requester Pays.**
base 24x7 estável + picos tolerantes a interrupção	**Savings Plans ou RIs na base, Spot nos picos, mixed instances policy.**
lote tolerante a interrupção no Fargate	**Fargate Spot via capacity provider strategy.**
Spot com menos interrupções e menor preço	**Estratégia price-capacity-optimized com vários tipos de instância.**
custo alto de NAT com tráfego para S3 ou DynamoDB	**Gateway endpoint.**
bucket versionado crescendo sem explicação	**Lifecycle com AbortIncompleteMultipartUpload e NoncurrentVersionExpiration.**
custo alto de Cross-Region Replication com prefixos temporários	**Filtro de prefixo ou tag na regra de replicação; RTC acelera, não economiza.**
I/O representa 25% ou mais da conta do Aurora	**Aurora I/O-Optimized.**
banco de desenvolvimento ocioso à noite e no fim de semana, uso imprevisível	**Aurora Serverless v2 com mínimo de 0 ACU.**
apagar itens antigos do DynamoDB sem gastar escrita	**TTL com atributo de expiração em epoch (segundos).**
análise de custo por recurso com SQL livre	**CUR 2.0 no AWS Data Exports em Parquet + Athena.**

orçamento estourado deve bloquear criação de instâncias sem código	**AWS Budgets com budget actions.**
CPU média baixa com picos curtos	**Instâncias da família T (créditos de CPU).**
licença por soquete ou núcleo trazida pela empresa	**Dedicated Hosts + AWS License Manager.**

🎯 Se você lembrar de só uma coisa deste capítulo...

- Compromisso (Savings Plan ou RI) é cobrado em todas as horas: só para base estável.
- Compute Savings Plans não cobrem RDS; EC2 Instance Savings Plans não cobrem Fargate.
- Standard-IA cobra 128 KB e 30 dias mínimos por objeto.
- Gateway endpoint para S3 e DynamoDB elimina custo de NAT.
- Ajuste o tamanho antes de reservar.

Essenciais para a prova (checklist)

☐ Sei escolher a classe do S3 pela frequência de acesso, tempo de recuperação e mínimos de cobrança.

☐ Sei quando usar lifecycle e quando usar Intelligent-Tiering.

☐ Sei diferenciar Compute, EC2 Instance e Database Savings Plans e RIs Standard e conversíveis.

☐ Sei montar mixed instances policy com base On-Demand e Spot e a estratégia price-capacity-optimized.

☐ Sei reduzir custo de NAT, transferência entre AZs, saída para a internet e IPv4 público.

☐ Sei otimizar custos do DynamoDB (modo de capacidade, leitura eventual, TTL) e do Aurora (I/O-Optimized, 0 ACU, clone).

☐ Sei usar Budgets actions, Cost Explorer, CUR 2.0 com Athena e Compute Optimizer.

☐ Sei desligar recursos ociosos com scheduled actions, Instance Scheduler e hibernação.

Questões de revisão do capítulo

R4.1 · Domínio 4: Criação de arquiteturas com custo otimizado

O time de FinOps de uma varejista com 40 contas no AWS Organizations precisa responder perguntas detalhadas, como o custo diário de cada instância EC2 por tag de centro de custo, quanto de cada Savings Plan foi aplicado a cada conta e quais recursos geraram cobranças de transferência de dados. As consultas mudam toda semana e o time quer usar SQL sobre os dados de cobrança no nível de linha e de recurso, com o MENOR esforço. Qual solução atende a esses requisitos?

A. Usar os relatórios do AWS Cost Explorer agrupados por conta, serviço e tag e exportar os gráficos em CSV a cada semana para as análises
B. Criar uma trilha organizacional do AWS CloudTrail entregue no S3 e consultar no Amazon Athena os eventos de criação de recursos para estimar os custos
C. Configurar orçamentos do AWS Budgets por centro de custo com relatórios diários enviados por e-mail ao time de FinOps para as análises semanais
D. Criar no AWS Data Exports uma exportação padrão do CUR 2.0 com IDs de recurso, entregue em Parquet no S3 da conta de gerenciamento, e consultá-la no Amazon Athena

R4.2 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma empresa de adtech executa centenas de instâncias Amazon EC2 em um Auto Scaling group para processar leilões em tempo real (RTB). A carga é altamente variável minuto a minuto, mas as análises mostram que, em média, 60% da capacidade de computação é utilizada 24x7 ao longo de meses, enquanto os 40% restantes são picos imprevisíveis. A empresa quer reduzir custos de computação sem comprometer significativamente a disponibilidade do serviço. Qual combinação de modelos de compra de instância é MAIS adequada para otimizar custos mantendo um bom nível de disponibilidade?

A. Usar instâncias On-Demand para as cargas, garantindo disponibilidade máxima a custo alto
B. Migrar toda a carga para Amazon EC2 Dedicated Hosts
C. Usar só instâncias Spot no Auto Scaling group, confiando na substituição rápida em interrupções
D. Usar Savings Plans ou RIs para cerca de 60% da base e Spot para o restante

R4.3 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma produtora de vídeo precisa migrar 40 TB de arquivos de um NAS NFS local para o Amazon S3 em até 3 semanas. Ela tem um link de internet de 1 Gbps com mais da metade da capacidade ociosa à noite e nos fins de semana. Uma nova conexão AWS Direct Connect levaria semanas para ser provisionada e geraria custo mensal contínuo. Qual solução cumpre o prazo com o MENOR custo?

A. Implantar um agente do AWS DataSync perto do NAS e agendar uma tarefa de NFS para S3 com limite de banda ajustado aos horários ociosos
B. Contratar uma conexão dedicada do AWS Direct Connect de 10 Gbps e copiar os arquivos com a AWS CLI depois que a porta estiver ativa
C. Solicitar dispositivos AWS Snowball Edge Storage Optimized, copiar os 40 TB localmente e enviá-los de volta à AWS
D. Implantar um Amazon S3 File Gateway e copiar os arquivos do NAS para o compartilhamento do gateway durante o horário comercial

R4.4 · Domínio 4: Criação de arquiteturas com custo otimizado

Uma startup de mídia digital precisa armazenar e distribuir milhares de pequenos arquivos de vídeo gerados diariamente. Esses arquivos são muito acessados nas primeiras 24 horas após o upload e raramente acessados depois disso. A empresa deseja minimizar custos de armazenamento e transferência de dados, mantendo baixa latência para os usuários durante o primeiro dia. Qual solução atende melhor a esses requisitos com o menor custo total?

A. Armazenar os vídeos em um bucket Amazon S3 Standard e distribuí-los pelo Amazon CloudFront
B. Armazenar os vídeos diretamente em instâncias Amazon EC2 com EBS e usar um Application Load Balancer para distribuir o tráfego
C. Usar S3 Intelligent-Tiering como classe dos vídeos, com CloudFront e TTLs ajustados
D. Armazenar os vídeos em Amazon S3 Glacier Flexible Retrieval e usar CloudFront para distribuir os objetos sob demanda

Respostas comentadas

R4.1. D) Criar no AWS Data Exports uma exportação padrão do CUR 2.0 com IDs de recurso, entregue em Parquet no S3 da conta de gerenciamento, e consultá-la no Amazon Athena

Por quê: O AWS Data Exports cria exportações recorrentes dos dados de cobrança no Amazon S3, e o Cost and Usage Report 2.0 é a forma recomendada de receber os dados mais detalhados de custo e uso, com seleção de colunas e filtros por SQL. Entregue em Parquet, o relatório pode ser catalogado e consultado no Amazon Athena para análises ad hoc por recurso, conta, tag, Savings Plan ou tipo de uso, como transferência de dados.

✗ A) Incorreta. O Cost Explorer mostra custos agregados em gráficos e filtros predefinidos, mas não permite SQL livre sobre cada linha de cobrança, e exportar gráficos toda semana é trabalho repetitivo.

✗ B) Incorreta. O CloudTrail registra chamadas de API, não valores cobrados; estimar custos a partir de eventos de criação não traz uso, preços, descontos de Savings Plans nem transferência de dados.

✗ C) Incorreta. O Budgets acompanha gastos contra limites e envia alertas e relatórios resumidos; não fornece dados no nível de linha de cobrança para consultas SQL.

> **Dica de prova:** Análise de custo no nível de linha/recurso com SQL = CUR 2.0 (Data Exports) no S3 + Athena. Cost Explorer é visão agregada; Budgets é alerta.

R4.2. D) Usar Savings Plans ou RIs para cerca de 60% da base e Spot para o restante

Por quê: Para uma carga com base estável e picos variáveis, combinam-se modelos de compra: a parte usada 24x7, cerca de 60%, fica coberta por Savings Plans ou Reserved Instances, com desconto de uso comprometido, e os picos imprevisíveis rodam em instâncias Spot, com o maior desconto. Um Auto Scaling group com mixed instances policy, várias famílias de instância e a estratégia de alocação price-capacity-optimized reduz o risco de interrupção nos picos.

✗ A) Incorreta. On-Demand garante flexibilidade, mas ignora a economia clara de uma base usada 24x7, que é candidata clássica a Savings Plans ou Reserved Instances.

✗ B) Incorreta. Dedicated Hosts atendem a requisitos de licenciamento ou de hardware dedicado e custam mais que instâncias compartilhadas, sem tratar a elasticidade dos 40% de pico.

✗ C) Incorreta. Com tudo em Spot, uma onda de interrupções pode derrubar a base de capacidade de que o leilão depende 24x7; falta uma camada garantida coberta por Savings Plans ou Reserved Instances.

> **Dica de prova:** Base estável: Savings Plans ou RI. Picos elásticos e tolerantes: Spot. Combine tudo em um ASG com mixed instances policy.

R4.3. A) Implantar um agente do AWS DataSync perto do NAS e agendar uma tarefa de NFS para S3 com limite de banda ajustado aos horários ociosos

Por quê: Antes de contratar conectividade nova, calcule se o link atual comporta a migração no prazo: 40 TB a cerca de 500 Mbps levam pouco mais de uma semana de transferência contínua. O AWS DataSync faz migrações pela rede com protocolo otimizado, verificação de integridade, criptografia, agendamento e limite de banda, sem custo fixo de porta.

✗ B) Incorreta. O provisionamento da porta e da cross connect pode consumir o prazo, e a conexão gera custo mensal que a empresa quer evitar para uma migração pontual.

✗ C) Incorreta. O Snowball Edge está fechado a novos clientes, e enviar dispositivos não é necessário quando o link existente comporta a transferência dentro do prazo.

✗ D) Incorreta. Copiar em horário comercial disputa o link com a operação, e o gateway acrescenta um cache local e uma camada de arquivos que a migração pontual não precisa.

> **Dica de prova:** Faça a conta banda x volume x prazo. Se o link atual dá conta, DataSync pela rede é mais barato que Direct Connect novo.

R4.4. C) Usar S3 Intelligent-Tiering como classe dos vídeos, com CloudFront e TTLs ajustados

Por quê: O S3 Intelligent-Tiering move cada objeto para camadas mais baratas conforme ele deixa de ser acessado (Infrequent Access após 30 dias e Archive Instant Access após 90 dias), sem taxa de recuperação e com leitura em milissegundos, o que serve a arquivos que esfriam rápido. O CloudFront absorve o pico das primeiras 24 horas a partir do cache nos pontos de presença, com latência menor e menos requisições e transferência a partir do S3. Objetos menores que 128 KB não são movidos de camada, mas vídeos costumam ser maiores.

✗ A) Incorreta. O S3 Standard com CloudFront entrega bem no primeiro dia, mas os arquivos que quase não são mais acessados continuam pagando a tarifa do Standard, mais cara que a das camadas de acesso infrequente.

✗ B) Incorreta. Guardar vídeos em EBS de instâncias EC2 custa bem mais por GB que o S3 e exige gerenciar capacidade, disponibilidade e escala dos servidores.

✗ D) Incorreta. O Glacier Flexible Retrieval exige restaurar o objeto antes da leitura, o que leva de minutos a horas, incompatível com o acesso intenso logo após o upload.

> **Dica de prova:** Conteúdo que esfria rápido: Intelligent-Tiering (ou lifecycle se o padrão for fixo) + CloudFront para o pico. Glacier Flexible Retrieval não serve para leitura imediata.

DIA DA PROVA

Estratégia de prova

A SAA-C03 é uma prova de cenários: cada enunciado descreve uma empresa, um problema e uma ou duas restrições decisivas, como menor custo, menor esforço operacional, sem alterar código, sem tráfego pela internet ou sem credenciais de longo prazo. Quase sempre duas alternativas funcionam tecnicamente; a certa é a que atende a TODAS as restrições com um serviço gerenciado e nativo. As erradas costumam ser soluções caseiras (Lambda agendada, script, servidor próprio em EC2), serviços vizinhos com outra função ou recursos que não existem do jeito descrito, como gateway endpoint para SQS ou WAF em NLB. São 65 itens em 130 minutos, 15 deles não pontuados e sem identificação, em múltipla escolha e múltipla resposta, com nota de 100 a 1000 e mínimo de 720.

Como responder

✓ Leia primeiro a pergunta final e marque o qualificador: MAIS econômica, MENOR esforço operacional, MAIS resiliente, MENOR latência. Ele decide entre duas opções que funcionam.

✓ Liste as restrições do cenário (sem mudar código, sem internet, sem chaves, preventivo, mesma URL) e elimine qualquer alternativa que viole uma delas, mesmo que funcione.

✓ Quando o enunciado fala em menor esforço, prefira o recurso nativo e gerenciado a Lambda agendada, script ou servidor próprio.

✓ Identifique o gargalo antes das alternativas: leitura, gravação, IOPS, throughput, latência de rede, CPU, memória ou custo de transferência.

✓ Em múltipla resposta (Escolha DUAS), cada alternativa certa costuma resolver uma parte diferente do requisito; confira se o par cobre o cenário inteiro.

✓ Ritmo de cerca de 2 minutos por item: marque os longos para revisão e volte a eles no fim, sem travar o restante da prova.

✓ Distribua a revisão pelos pesos: arquiteturas seguras 30%, resilientes 26%, alto desempenho 24% e custo otimizado 20%.

✓ Desconfie de alternativas com à mão, manual ou script quando existe recurso nativo, e de limites impossíveis (Lambda acima de 15 minutos, delay do SQS de dias).

Erros que derrubam candidatos

✗ Escolher a solução tecnicamente mais poderosa (ativo-ativo, io2, cluster dedicado) quando o enunciado pediu a mais econômica.

✗ Escolher controle detectivo (AWS Config, Access Analyzer, GuardDuty) quando o enunciado pede impedir.

✗ Tratar réplica de leitura como alta disponibilidade ou como escala de gravação.

✗ Confundir serviços de nome parecido: CloudFront x Global Accelerator x Transfer Acceleration, GuardDuty x Inspector x Macie, Textract x Rekognition x Comprehend.

✗ Ignorar uma restrição escondida no meio do texto, como sem alterar o código, a mesma URL ou manter o bucket único.

✗ Comprar compromisso (Savings Plan, RI) para carga que fica ociosa a maior parte do tempo.

✗ Esquecer limites que derrubam alternativas inteiras: Lambda 15 minutos, Express 5 minutos, SQS delay 15 minutos e mensagem de 1 MiB.

Na semana da prova

- ☐ Refaça as provas e, em cada erro, escreva qual restrição do enunciado você ignorou; esse padrão se repete.
- ☐ Revise as tabelas de pares que a prova confunde: SG x NACL, Multi-AZ x réplica, delay x visibility, CloudFront x Global Accelerator, Compute x EC2 Instance Savings Plans.
- ☐ Decore os números que eliminam alternativas: limites de Lambda, SQS e Step Functions, mínimos das classes do S3, base do gp3, portas efêmeras.
- ☐ Releia os gatilhos de cada domínio e treine dizer o serviço antes de ver as alternativas.
- ☐ Na véspera, só revisão leve dos macetes e do checklist; descanse para manter o ritmo de 2 minutos por item.

REVISÃO FINAL

Folha de cola

Leia na véspera e na manhã da prova: os gatilhos e as pegadinhas mais importantes de cada domínio.

1. Domínio 1: Criação de arquiteturas seguras

‣ sem chaves de acesso de longo prazo, credenciais temporárias → **Role do IAM: instance profile no EC2, task role no ECS, EKS Pod Identity ou IRSA, OIDC para pipelines.**

‣ fornecedor terceiro que atende vários clientes assume role → **Trust policy com sts:ExternalId.**

‣ funcionários com o login corporativo em várias contas → **IAM Identity Center federado via SAML 2.0, permission sets por conta.**

‣ usuários finais de aplicativo móvel acessando S3 ou DynamoDB direto → **Cognito user pool + identity pool com role usando cognito-identity.amazonaws.com:sub.**

‣ limitar requisições por IP, SQL injection, XSS → **AWS WAF com regra rate-based e regras gerenciadas no CloudFront, ALB ou API Gateway.**

⚠ SCP nunca concede nada e não afeta a conta de gerenciamento; a permissão vem de política do IAM e a carga controlada por SCP deve ficar em conta-membro.

⚠ A key policy de chave gerenciada pela AWS não pode ser editada. Copie ou recriptografe com chave gerenciada pelo cliente e libere a conta na key policy.

⚠ Sem kms:Decrypt liberado na key policy a leitura falha com acesso negado; são precisas bucket policy E key policy.

2. Domínio 2: Criação de arquiteturas resilientes

‣ mensagens invisíveis por alguns minutos logo após o envio → **Delay queue (DelaySeconds até 15 minutos).**

‣ muitas respostas vazias, custo alto de ReceiveMessage → **Long polling com ReceiveMessageWaitTimeSeconds de 20 segundos.**

‣ arquivo de centenas de MB na mensagem da fila → **SQS Extended Client Library: objeto no S3, ponteiro na mensagem.**

‣ SQS + Lambda processando a mesma mensagem duas vezes → **Visibility timeout de pelo menos 6 vezes o timeout da função.**

‣ Lambda que só repassa a requisição para SQS, EventBridge, Kinesis ou Step Functions → **Integração direta do API Gateway com role do IAM (HTTP API SQS-SendMessage).**

⚠ O visibility timeout só começa quando um consumidor recebe a mensagem. Atraso inicial = delay queue; acima de 15 minutos = EventBridge Scheduler.

⚠ FIFO deduplica envios repetidos em 5 minutos, não reentregas após falha do consumidor. Torne o consumidor idempotente com gravação condicional (attribute_not_exists).

⚠ Cada mensagem é consumida por um só consumidor. Vários consumidores independentes = SNS com uma fila SQS por sistema (SNS FIFO para SQS FIFO se exigir ordem).

3. Domínio 3: Criação de arquiteturas de alto desempenho

‣ arquivos compartilhados entre instâncias de várias AZs, Linux → **Amazon EFS.**

‣ SMB com AD e NTFS / mesmos arquivos por NFS e SMB / HPC lendo dados do S3 → **FSx for Windows File Server / FSx for NetApp ONTAP / FSx for Lustre.**

‣ servidores on-premises gravando por NFS ou SMB no S3 com cache local → **S3 File Gateway.**

‣ leituras repetitivas no DynamoDB em microssegundos → **DAX.**

‣ placar ou ranking em tempo real → **Sorted set no ElastiCache for Valkey.**

⚠ Provisionada só elimina cold start e ephemeral storage é disco. CPU vem com a memória: aumente a memória.

⚠ Memória não amplia /tmp; configure o ephemeral storage até 10.240 MB. Armazenamento compartilhado ou persistente = EFS.

⚠ gp3 provisiona IOPS e throughput independentes do tamanho; gp2 crescido paga GiB sem uso e para em 250 MiB/s. io2 só quando precisa de mais de 80.000 IOPS ou latência abaixo de 1 ms.

4. Domínio 4: Criação de arquiteturas com custo otimizado

‣ quem baixa os dados deve pagar a transferência → **S3 Requester Pays.**

‣ base 24x7 estável + picos tolerantes a interrupção → **Savings Plans ou RIs na base, Spot nos picos, mixed instances policy.**

‣ lote tolerante a interrupção no Fargate → **Fargate Spot via capacity provider strategy.**

‣ Spot com menos interrupções e menor preço → **Estratégia price-capacity-optimized com vários tipos de instância.**

‣ custo alto de NAT com tráfego para S3 ou DynamoDB → **Gateway endpoint.**

⚠ Essas classes cobram no mínimo 128 KB por objeto e 30 dias (90 no Glacier IR). Objeto pequeno ou de vida curta fica no Standard.

⚠ O Intelligent-Tiering só acrescenta taxa de monitoramento; lifecycle por idade é mais barato e ainda expira os dados.

⚠ Expedited leva minutos e exige restaurar antes. Milissegundos com acesso raro = Glacier Instant Retrieval.

GABARITO

Respostas do diagnóstico

D1. B) Criar uma CA privada no AWS Private CA e emitir pelo ACM certificados privados para os ALBs internos, renovados pelo ACM → revise o capítulo 1

Por quê: AWS Private CA cria hierarquias de autoridade certificadora privada (raiz e subordinadas) para emitir certificados confiáveis só dentro da organização. Integrado ao ACM, o certificado privado pode ser associado a ALBs e a outros serviços integrados ao ACM, que gerencia a renovação, sem servidores de CA para operar.

✗ A) Incorreta. Certificados públicos exigem validar um domínio registrado publicamente; um domínio privado como corp.internal não pode ser validado por uma CA pública.

✗ C) Incorreta. Server certificates do IAM são uma opção legada para Regiões sem ACM, sem renovação automática e ainda dependentes de uma CA para emiti-los.

✗ D) Incorreta. Certificados importados no ACM não são renovados pelo ACM, e manter uma CA em OpenSSL numa instância é a operação que a empresa quer evitar.

> **Dica de prova:** Domínio interno com TLS e renovação gerenciada = AWS Private CA + certificado privado do ACM.

D2. B) Habilitar o padrão AWS Foundational Security Best Practices no Security Hub CSPM das contas e configurar a agregação entre Regiões com uma Região inicial → revise o capítulo 1

Por quê: O AWS Security Hub CSPM faz o gerenciamento da postura de segurança: executa verificações automáticas baseadas em padrões, como AWS Foundational Security Best Practices e CIS, consolida achados de outros serviços e calcula pontuações. Com administrador delegado e agregação entre Regiões, a equipe vê e gerencia a postura da organização inteira a partir de uma Região.

✗ A) Incorreta. O GuardDuty detecta ameaças e atividades maliciosas em logs; ele não verifica se as configurações seguem boas práticas, e contar achados não mede a postura.

✗ C) Incorreta. O Inspector procura vulnerabilidades de software e exposição de rede em cargas de trabalho; não avalia configurações como buckets públicos ou criptografia do EBS.

✗ D) Incorreta. Funciona, mas escrever e manter dezenas de regras e funções em 25 contas e quatro Regiões é o esforço que os controles gerenciados do Security Hub CSPM eliminam.

> **Dica de prova:** Postura de configuração contra boas práticas, com pontuação = Security Hub CSPM (padrão FSBP). Ameaças = GuardDuty. Vulnerabilidades de software = Inspector.

D3. B) Habilitar a criptografia por padrão do EBS em sa-east-1 em cada conta e definir a chave KMS gerenciada pelo cliente como chave padrão de criptografia do EBS → revise o capítulo 1

Por quê: Criptografia por padrão do EBS é uma configuração por conta e por Região: depois de ligada, os volumes novos e as cópias de snapshot criados na Região saem criptografados, sem que ninguém precise marcar a opção. A chave padrão é a aws/ebs, mas pode ser trocada por uma chave gerenciada pelo cliente. A configuração não afeta volumes e snapshots existentes, que são criptografados copiando o snapshot com criptografia e criando um novo volume a partir dele.

✗ A) Incorreta. Continua dependendo de cada equipe e não cobre volumes criados por scripts ou pelo console fora desses templates.

✗ C) Incorreta. É reativa e pesada: o volume nasce sem criptografia e só depois é substituído, com risco de interrupção e muito mais automação para manter.

✗ D) Incorreta. A política exige chave gerenciada pelo cliente, não a aws/ebs, e um volume existente não é criptografado pelo modify-volume; é preciso copiar um snapshot com criptografia e criar um volume novo.

> **Dica de prova:** Todo EBS novo criptografado sem depender das equipes = encryption by default na Região, com a chave KMS do cliente como padrão.

D4. B) Usar Multi-Region ativo-passivo com failover automático no Route 53 e dados replicados de forma assíncrona e contínua → revise o capítulo 2

Por quê: Para RTO de minutos e RPO de até 5 minutos diante da perda de uma Região inteira, a estratégia é ativo-passivo multi-Região: a Região secundária mantém a infraestrutura em pé (warm standby) e recebe os dados por replicação assíncrona contínua, como réplicas entre Regiões do RDS ou Aurora Global Database, e o Route 53 com health checks desvia o tráfego. Backups diários dão RPO de até 24 horas, e replicação síncrona entre Regiões não é prática.

✗ A) Incorreta. Manter apenas uma região ativa e fazer cópias cross-Region diárias via AWS Backup implica em reconstruir toda a infraestrutura e restaurar dados manualmente em caso de desastre regional.

✗ C) Incorreta. A arquitetura Multi-Region Active-Active com replicação síncrona entre regiões é, na prática, inviável na maior parte dos cenários devido à latência de rede entre regiões e à ausência de suporte nativo a replicação estritamente...

✗ D) Incorreta. A utilização de instâncias em modo 'standby' em outra região com backups diários do Amazon RDS resulta em RPO potencialmente de muitas horas (até 24h) e em um RTO elevado, pois a restauração a partir de backup é demorada.

> **Dica de prova:** Desastre regional com RTO e RPO de minutos: warm standby em outra Região, replicação assíncrona e failover no Route 53.

D5. B) Gerar a AMI com patches em um pipeline do EC2 Image Builder, publicar uma nova versão do launch template e substituir as instâncias por instance refresh → revise o capítulo 2

Por quê: Infraestrutura imutável significa substituir em vez de modificar: servidores não recebem patches nem mudanças de configuração depois de lançados. Toda alteração gera um novo artefato versionado, como uma AMI criada por um pipeline do EC2 Image Builder, e o Auto Scaling troca as instâncias pela nova versão do launch template. Isso elimina o desvio de configuração, torna os ambientes reproduzíveis e simplifica o rollback.

✗ A) Incorreta. O AWS Config detecta e registra desvios, mas a correção continuaria sendo feita em cada servidor em execução, sem garantir instâncias idênticas.

✗ C) Incorreta. Continua sendo infraestrutura mutável: os patches são aplicados sobre servidores em execução, o histórico de mudanças diverge entre nós e não há uma versão de imagem para onde voltar.

✗ D) Incorreta. Capturar a imagem de um servidor já alterado à mão herda os desvios acumulados, e o processo não é reproduzível nem automatizado.

> **Dica de prova:** Servidores divergentes por patch no lugar = infraestrutura imutável: nova AMI (Image Builder), nova versão do launch template e substituição das instâncias.

D6. B) Adicionar ao listener do ALB uma regra de caminho /receitas/* que encaminhe ao target group do serviço ECS, mantendo a regra padrão no target group do monolito → revise o capítulo 2

Por quê: No padrão strangler fig, uma camada de proxy intercepta as requisições e encaminha as funcionalidades já migradas para os novos serviços, enquanto o monolito continua atendendo o restante, até poder ser desligado. Na AWS, essa camada pode ser o ALB com regras por caminho ou o Amazon API Gateway com rotas para os microsserviços e uma rota padrão para o monolito.

✗ A) Incorreta. O redirecionamento muda a URL que o cliente acessa e depende de os aplicativos seguirem o 301, o que viola o requisito de manter a URL.

✗ C) Incorreta. O DNS não distingue caminhos; os registros ponderados mandariam todas as rotas, e não só as de receitas, para o novo balanceador.

✗ D) Incorreta. A troca de uma vez só é a migração big bang que o padrão strangler fig evita; ela concentra o risco e não permite devolver só uma rota ao monolito.

> **Dica de prova:** Migrar o monolito aos poucos sem mudar a URL = strangler fig: proxy (ALB ou API Gateway) roteando por caminho.

D7. A) Configurar o Amazon CloudFront como CDN na frente do ALB e habilitar cache de respostas estáticas e dinâmicas quando possível → revise o capítulo 3

Por quê: Quando muitas requisições são idênticas e leem dados que mudam pouco, o cache elimina o trabalho repetido. Com o Amazon CloudFront na frente do ALB, as respostas cacheáveis (inclusive conteúdo dinâmico com TTL curto definido na cache policy) são entregues pelas edge locations, perto dos usuários, e o tráfego que chega às instâncias EC2 e ao banco cai bastante. Trocar o load balancer, crescer o RDS ou adicionar instâncias só aumenta a capacidade para repetir o mesmo trabalho.

✗ B) Incorreta. Escalar verticalmente o RDS pode oferecer algum ganho, mas não é a abordagem mais eficiente para lidar com requisições repetidas e dados raramente alterados. Continua existindo latência de rede e carga desnecessária no banco.

✗ C) Incorreta. O Network Load Balancer é adequado para tráfego TCP/UDP de baixa latência e alto throughput, mas não oferece cache de conteúdo nem resolve o problema de múltiplas leituras repetidas ao banco de dados.

✗ D) Incorreta. Apenas aumentar o número de instâncias EC2 trata sintomas de carga, mas não resolve a causa principal: múltiplas requisições idênticas a dados quase estáticos. Isso aumenta custos e não utiliza o benefício de cache em borda.

> **Dica de prova:** Muitas leituras idênticas de dados que mudam pouco: cache. Na borda e perto do usuário, CloudFront; na frente do banco, ElastiCache.

D8. B) Criar um Amazon FSx for NetApp ONTAP com volumes multiprotocolo, snapshots agendados e política de tiering para o capacity pool → revise o capítulo 3

Por quê: O Amazon FSx for NetApp ONTAP é um sistema de arquivos gerenciado com acesso multiprotocolo (NFS, SMB e iSCSI), snapshots e eficiência de armazenamento. Cada sistema tem um nível SSD de alto desempenho e um capacity pool elástico e mais barato; a política de tiering de cada volume move automaticamente os dados pouco usados para o capacity pool.

✗ A) Incorreta. O FSx for Lustre exige o cliente Lustre e é voltado a computação de alto desempenho; ele não serve compartilhamentos SMB nem NFS às estações de trabalho.

✗ C) Incorreta. O Amazon EFS é acessado por NFS; ele não oferece SMB, então as estações Windows não abririam os arquivos pelo protocolo exigido.

✗ D) Incorreta. O FSx for Windows File Server oferece compartilhamentos SMB; ele não serve os mesmos dados por NFS às estações Linux.

> **Dica de prova:** Mesmos arquivos por NFS e SMB + snapshots + tiering automático = FSx for NetApp ONTAP.

D9. B) Converter os dados para o formato Parquet compactado e criar partições nas tabelas do Athena com base em colunas de alto filtro, como data ou região. → revise o capítulo 3

Por quê: No Amazon Athena, o custo e o tempo da consulta acompanham a quantidade de dados lidos. Formatos colunares como Parquet leem só as colunas usadas, compactam por coluna e permitem pular blocos pelas estatísticas, e o particionamento por data ou região descarta pastas inteiras que não atendem ao filtro. A conversão pode ser feita com CTAS no próprio Athena ou com um job do AWS Glue.

✗ A) Incorreta. Migrar tudo para Redshift adiciona complexidade e esforço operacional significativo (provisionamento, dimensionamento, cópias, manutenção), além de não resolver o problema raiz no S3 (formato CSV ineficiente).

✗ C) Incorreta. O S3 Express One Zone reduz a latência de cada requisição, mas o custo do Athena depende dos bytes escaneados: CSV sem partição continua lido por inteiro. Além disso, a classe custa mais por GB e guarda os dados em uma única AZ.

✗ D) Incorreta. Usar ElastiCache para armazenar resultados de consultas não resolve a ineficiência do formato CSV e da ausência de particionamento.

> **Dica de prova:** No Athena, custo e performance estão diretamente ligados à quantidade de dados lidos. Use formatos colunares (Parquet, ORC) e particionamento no S3 sempre que lidar com grandes volumes. Lembre-se: 'Menos bytes lidos = consultas mais rápidas e mais baratas'.

D10. D) Substituir os NAT gateways por uma NAT instance pequena em sub-rede pública, com a verificação de origem e destino desligada → revise o capítulo 4

Por quê: NAT gateways são gerenciados, escalam e são redundantes na zona, e a AWS os recomenda na maioria dos casos. Uma NAT instance exige patches, failover por script e dimensionamento, mas o custo é só o da instância, o que pode valer a pena em ambientes de desenvolvimento com pouco tráfego e sem requisito de alta disponibilidade.

✗ A) Incorreta. O NAT gateway regional simplifica a operação, mas continua sendo um NAT gateway gerenciado, com cobrança por hora e por GB processado, mais caro que uma instância pequena para poucos GB.

✗ B) Incorreta. Reduz o custo em relação a três gateways, mas um NAT gateway ligado o mês inteiro ainda custa mais que uma instância pequena, e há transferência entre zonas.

✗ C) Incorreta. O egress-only internet gateway funciona para tráfego IPv6; ele não faz NAT de IPv4 e não pode ser destino de uma rota IPv4.

> **Dica de prova:** Dev com pouco tráfego, sem HA e equipe que aceita administrar = NAT instance pequena. Produção = NAT gateway.

D11. B) Substituir o Global Accelerator por uma distribuição do Amazon CloudFront com o ALB como origem e políticas de cache para os e-books e os arquivos do site → revise o capítulo 4

Por quê: Global Accelerator melhora o desempenho de aplicações TCP e UDP pela rede da AWS e se justifica para tráfego não HTTP (jogos em UDP, IoT com MQTT, voz) ou HTTP que precise de IPs estáticos ou failover regional rápido; ele cobra uma taxa fixa por hora de cada acelerador e uma taxa Data Transfer-Premium por GB, além da transferência normal. Para conteúdo HTTP cacheável, o CloudFront entrega dos pontos de presença, reduz as idas à origem e não cobra a busca de origens AWS, com menor custo e latência.

✗ A) Incorreta. IPs fixos não são necessários e o NLB não tem rede de borda nem cache, então os leitores distantes perderiam desempenho.

✗ C) Incorreta. O Transfer Acceleration acelera transferências de e para o bucket com cobrança adicional por GB, mas não faz cache dos arquivos para os leitores.

✗ D) Incorreta. Uma segunda Região e mais um endpoint group aumentam o custo de infraestrutura e o Global Accelerator continua cobrando a taxa por hora e a Data Transfer-Premium por GB.

> **Dica de prova:** HTTP cacheável sem exigência de IP fixo = CloudFront. Global Accelerator vale o custo para UDP ou TCP não HTTP, IPs estáticos ou failover multirregião rápido.

D12. B) Criar em cada conta um orçamento do AWS Budgets com budget actions em 100% que apliquem uma política IAM negando ec2:RunInstances e parem as instâncias de dev → revise o capítulo 4

Por quê: O AWS Budgets pode executar ações quando o custo ou o uso passa de um limite: aplicar uma política IAM, aplicar uma SCP (a partir da conta de gerenciamento) ou parar instâncias EC2 e RDS da própria conta. As ações rodam automaticamente ou com aprovação e usam uma role de execução, transformando o orçamento em controle, e não só em alerta.

✗ A) Incorreta. A solução depende de código próprio na Lambda, que o requisito exclui, e o alarme de faturamento só avisa sobre o custo estimado.

✗ C) Incorreta. A notificação avisa, mas a resposta depende de pessoas seguirem o runbook, então o bloqueio não é automático.

✗ D) Incorreta. O Cost Anomaly Detection identifica gastos fora do padrão e notifica, mas não compara com um orçamento fechado nem executa ações.

> **Dica de prova:** Orçamento que precisa agir (bloquear, parar) = AWS Budgets com budget actions. Anomaly Detection e alarmes só avisam.

Agora, treine no formato real

Faça as 6 provas completas do curso SAA-C03: 390 questões inéditas, tempo do exame, explicação de cada alternativa e link da documentação para revisar.

Ver as 6 provas completas e o cupom do mês →

Até o dia da prova, uma revisão por dia no seu feed

Instagram @strecht.simulados

Um post de prova por dia: questão comentada, pegadinhas e o “não confunda” de cada certificação.

Seguir no Instagram →

YouTube Strecht Simulados

Shorts diários com pegadinhas de prova e, aos sábados, um simulado comentado.

Inscrever-se no canal →

Material independente de preparação, sem afiliação ou endosso de Amazon Web Services (AWS). As questões são inéditas e não reproduzem a prova real.