Strecht Simulados · Guia de preparação



Guia de preparação SOA-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.

5 capítulos por domínio	**32** 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 SOA-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: Monitoramento, registro em log, análise, correção e otimização de desempenho	22%	18 conceitos · 14 pegadinhas
2. Domínio 2: Confiabilidade e continuidade dos negócios	22%	18 conceitos · 14 pegadinhas
3. Domínio 3: Implantação, provisionamento e automação	22%	18 conceitos · 14 pegadinhas
4. Domínio 4: Segurança e conformidade	16%	18 conceitos · 14 pegadinhas
5. Domínio 5: Redes e entrega de conteúdo	18%	18 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: Monitoramento, registro em log, análise, correção e otimização de desempenho

Uma empresa tem um runbook do Systems Manager Automation que cria AMIs de instâncias marcadas com Backup=semanal. Ele deve rodar todo domingo às 2h no horário de Brasília, nunca durante o expediente, com no máximo três horas de duração e sem iniciar execuções novas na última meia hora. Qual solução atende ao requisito?

A. Criar uma regra do EventBridge com expressão cron em UTC que inicia o runbook e configurar o rate control do Automation para encerrar após três horas
B. Criar um comando do Run Command com o documento do runbook e um agendamento semanal definido nas opções de execução do próprio comando
C. Criar uma Maintenance Window com agendamento cron no fuso America/Sao_Paulo, duração de 3 horas e cutoff de 30 minutos, com alvos pela tag e uma tarefa do tipo Automation
D. Criar um Change Calendar com eventos aos domingos das 2h às 5h e configurar o runbook para consultar o calendário antes de cada execução nas instâncias

D2 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Três auditores externos, sem conta na AWS, precisam acompanhar por algumas semanas um dashboard do CloudWatch com métricas de disponibilidade. A segurança não aceita criar usuários IAM para eles nem deixar o dashboard acessível a qualquer pessoa com o link. Qual abordagem atende aos requisitos?

A. Criar usuários IAM com a política CloudWatchReadOnlyAccess e acesso ao console, restritos ao dashboard por meio de condição na política
B. Configurar a observabilidade entre contas do CloudWatch, vinculando a conta como origem de métricas para uma conta monitora dos auditores
C. Compartilhar o dashboard com os e-mails dos três auditores, para que cada um crie a própria senha de acesso
D. Compartilhar o dashboard publicamente e enviar o link só aos auditores, revogando o compartilhamento ao final da auditoria

D3 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Um cluster HPC usa um cluster placement group com 8 instâncias c6in em execução. Ao tentar adicionar mais 4 instâncias do mesmo tipo ao grupo, o lançamento falha com InsufficientInstanceCapacity. A equipe pode ter uma breve janela de parada e quer manter a baixa latência entre os nós. O que deve ser feito?

A. Converter o grupo para um spread placement group, que coloca cada instância em hardware distinto e evita falta de capacidade
B. Lançar as 4 novas instâncias em outra zona de disponibilidade usando o mesmo cluster placement group para dividir a carga do cluster
C. Pedir aumento da service quota de vCPUs On-Demand da família C, já que o erro indica limite de instâncias da conta
D. Parar as 8 instâncias do grupo, iniciá-las de novo e repetir o lançamento, o que pode levar o grupo a hardware com capacidade

D4 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma aplicação de relatórios roda em uma instância EC2 em uma única AZ, com dados em um volume Amazon EBS gp3. A empresa precisa se recuperar de uma falha da AZ com RPO de até 1 hora e RTO de algumas horas, com o menor custo e sem reescrever a aplicação. Qual abordagem atende aos requisitos?

A. Manter uma segunda instância em outra AZ com volume próprio e replicar os blocos de forma contínua por um software de replicação
B. Criar um plano do AWS Backup com backup diário do volume e retenção de 7 dias e restaurar o recovery point mais recente em outra AZ
C. Habilitar o EBS Multi-Attach em um volume io2 e anexá-lo também a uma instância em espera em outra AZ, que assume se a primeira AZ falhar
D. Criar uma política do Data Lifecycle Manager com snapshots do volume a cada hora e, na falha, restaurar o snapshot mais recente em outra AZ

D5 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma empresa vai expor uma aplicação web na internet com alta disponibilidade, distribuindo o tráfego entre instâncias Amazon EC2 em duas zonas de disponibilidade. A equipe quer verificações de saúde em nível de aplicação, retirada automática de instâncias com falha e gerenciamento centralizado dos certificados TLS no balanceamento, com o menor esforço operacional. Qual arquitetura atende a esses requisitos?

A. Usar um Application Load Balancer nas sub-redes públicas das duas zonas, com as instâncias no target group e listener HTTPS com certificado do ACM
B. Usar um Network Load Balancer com listener TCP na porta 443 e terminar o TLS nas instâncias, com os certificados importados em cada servidor web
C. Criar uma distribuição do CloudFront com as instâncias como origens em um origin group, usando origin failover para alternar entre as zonas
D. Criar registros de failover no Route 53 com health checks apontando para os endereços IP elásticos das instâncias de cada zona de disponibilidade

D6 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma função Lambda chamada pelo Amazon API Gateway grava em um Amazon RDS for MySQL que suporta cerca de 150 conexões. Em campanhas, a função chega a 1.000 execuções simultâneas, o banco recusa conexões e outras aplicações que usam o mesmo banco caem. A empresa quer limitar a função a 100 execuções simultâneas sem afetar as demais funções da conta. O que deve ser feito?

A. Configurar reserved concurrency de 100 na função, que limita as execuções simultâneas e reserva essa capacidade para ela
B. Aumentar a classe da instância RDS e o max_connections para comportar o pico de 1.000 execuções simultâneas da função
C. Configurar provisioned concurrency de 100 na função para manter no máximo 100 ambientes de execução ao mesmo tempo
D. Reduzir a cota de execuções simultâneas da conta para 100 no Service Quotas para conter as invocações da função

D7 · Domínio 3: Implantação, provisionamento e automação

Uma equipe de plataforma quer que vários times reutilizem o mesmo bloco padronizado de rede e segurança em seus próprios templates do AWS CloudFormation. O bloco precisa ter controle de versão, e cada time deve poder combiná-lo com os demais recursos do seu template. Mudanças futuras no bloco devem ser feitas uma única vez pelo time de plataforma. Qual solução atende com o menor esforço operacional?

A. Distribuir o bloco com CloudFormation StackSets para as contas dos times e permitir que cada time edite a stack instance na própria conta
B. Registrar o bloco como módulo do CloudFormation no registry privado e fazer os times o incluírem nos templates pelo nome do tipo terminado em MODULE
C. Publicar o bloco como produto do AWS Service Catalog e pedir que os times copiem o template do produto para dentro de seus templates antes de cada implantação
D. Manter o template do bloco em um repositório no GitHub conectado por CodeConnections e pedir que cada time faça fork e mantenha a própria cópia

D8 · Domínio 3: Implantação, provisionamento e automação

Instâncias EC2 de um Auto Scaling group precisam receber, no momento do boot, valores de configuração versionados no Parameter Store do AWS Systems Manager, como endpoints de serviços internos. Esses valores mudam com frequência, e a equipe quer evitar criar uma AMI nova a cada mudança, sem atraso entre o lançamento e a configuração. Qual abordagem atende aos requisitos?

A. Usar uma AMI base com um script de user data que lê os parâmetros do Parameter Store na inicialização e aplica a configuração
B. Criar um pipeline do EC2 Image Builder acionado pelo EventBridge a cada mudança de parâmetro, gerando uma AMI nova para o launch template
C. Criar uma associação do State Manager que aplica os parâmetros às instâncias uma vez por dia depois que o Auto Scaling group as lança
D. Gerar uma AMI por ambiente com os valores de configuração embutidos e atualizar o launch template do grupo sempre que um parâmetro mudar

D9 · Domínio 3: Implantação, provisionamento e automação

Para reduzir custos, as instâncias EC2 com a tag Ambiente=dev de uma conta devem ser paradas às 20h de segunda a sexta, no horário de Brasília. A equipe quer usar o mínimo de código próprio e ter histórico de execução por instância para auditoria. As instâncias não fazem parte de Auto Scaling groups. Qual solução atende?

A. Criar uma função Lambda que liste as instâncias pela tag e as pare, acionada por uma entrada de cron em uma instância EC2
B. Criar uma ação agendada do Auto Scaling que reduza a zero a capacidade das instâncias de desenvolvimento às 20h
C. Criar um agendamento do EventBridge Scheduler com cron e fuso America/Sao_Paulo que inicie o runbook AWS-StopEC2Instance pela tag
D. Criar em cada instância um alarme do CloudWatch de CPUUtilization baixa com a ação de EC2 que para a instância de desenvolvimento

D10 · Domínio 4: Segurança e conformidade

O AWS Trusted Advisor de uma conta mostra em vermelho a verificação Exposed Access Keys, para uma chave de um usuário IAM publicada em um repositório público, e a verificação MFA on Root Account. A conta não faz parte de uma organização. Quais ações seguem as recomendações dessas verificações?

A. Marcar a verificação como resolvida no Trusted Advisor após avisar o usuário e exigir MFA para o root por meio de um SCP
B. Remover do repositório público o arquivo com a chave e criar uma tag policy que exija MFA para o usuário root da conta
C. Rotacionar a chave exposta mantendo a antiga ativa por 7 dias para não interromper as aplicações e criar chaves de acesso para o root
D. Excluir a chave exposta, verificar no CloudTrail e nos consoles o uso não autorizado e habilitar MFA no usuário root da conta

D11 · Domínio 4: Segurança e conformidade

Desenvolvedores precisam criar roles IAM para suas funções Lambda sem abrir chamado para a segurança. Porém, nenhuma role criada por eles pode ter mais permissões do que as definidas na política gerenciada DevMaxPermissions, e eles não podem usar a criação de roles para ampliar os próprios privilégios. Como permitir isso?

A. Exigir que os desenvolvedores validem cada política com o IAM Access Analyzer antes de criar a role e que revisem os achados apontados
B. Aplicar DevMaxPermissions como SCP na conta e liberar iam:* aos desenvolvedores, já que a SCP limita as roles que eles venham a criar
C. Permitir iam:CreateRole e iam:AttachRolePolicy aos desenvolvedores e criar uma regra do AWS Config que alerte quando uma role tiver AdministratorAccess
D. Permitir iam:CreateRole com a condição iam:PermissionsBoundary igual ao ARN de DevMaxPermissions e negar a remoção ou troca do boundary

D12 · Domínio 4: Segurança e conformidade

Uma organização no AWS Organizations separa as contas de produção e de não produção em OUs. A equipe de segurança precisa impor em todas as contas controles mínimos, como impedir que o CloudTrail seja interrompido e restringir as regiões permitidas, mesmo para administradores locais, e as equipes de cada conta devem continuar definindo permissões próprias para seus workloads. Qual abordagem atende a esses requisitos de forma centralizada?

A. Anexar SCPs às OUs com os controles obrigatórios e deixar que cada conta defina políticas IAM adicionais para os seus workloads
B. Criar na conta de segurança uma política IAM gerenciada com os controles mínimos e anexá-la como permissions boundary às roles das contas-membro
C. Implantar regras do AWS Config com StackSets em cada conta e corrigir as violações com documentos do Run Command executados pela equipe de segurança
D. Habilitar o Security Hub CSPM com o padrão AWS Foundational Security Best Practices e usar os controles do padrão para bloquear as ações proibidas

01

CAPÍTULO 1 · 22% DA PROVA

Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Aqui eu preciso escolher a ferramenta certa de observabilidade para cada sintoma: métricas padrão x CloudWatch agent, logs, traces (X-Ray e ADOT), Application Signals, Container Insights, Database Insights e Enhanced Monitoring. A prova descreve um problema (latência, memória, 5XX, CPUCreditBalance ou BurstBalance zerado, regra que não dispara) e pede a métrica, o alarme ou a correção com o menor esforço operacional. Também cai muito EventBridge (event pattern, input transformer, retry e DLQ, archive e replay), desenho de alarmes (período x pontos de dados, composto, anomaly detection, mute rules) e remediação com Systems Manager Automation. A otimização de desempenho e custo mora aqui: EBS (gp2, gp3, io2, st1, sc1), EFS, S3, placement groups, Lambda e Savings Plans. Onde mais erro: achar que monitoramento detalhado traz memória, confundir Database Insights com Enhanced Monitoring e esquecer que eventos de serviços globais chegam em us-east-1.

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

Tópicos-chave

CloudWatch agent

As métricas padrão do EC2 vêm do hipervisor (CPU, rede, operações de disco) e não incluem memória nem espaço usado no sistema de arquivos. O agente coleta mem_used_percent e disk_used_percent, logs do sistema operacional e da aplicação e, com o plugin procstat, métricas por processo (cpu_usage, memory_rss, pid_count). Em frota grande, instale pelo Systems Manager (pacote AmazonCloudWatchAgent, mantido por State Manager) e guarde a configuração no Parameter Store.

Desenho de alarmes do CloudWatch

No monitoramento básico do EC2 as métricas chegam a cada 5 minutos e no detalhado a cada 1 minuto (o detalhado não acrescenta memória nem disco), e o período do alarme deve casar com essa frequência. Um alarme entra em ALARM quando M de N pontos de dados violam o limite (5 minutos x 3 pontos = 15 minutos para reagir). Alarmes compostos combinam outros alarmes com AND, OR e NOT e podem notificar SNS, invocar Lambda, criar OpsItem ou incidente. Alarmes de anomaly detection comparam a métrica com um modelo que considera tendência e sazonalidade, e Alarm Mute Rules silenciam ações em janelas definidas por cron.

CloudWatch Metrics Insights

Consulta métricas em sintaxe SQL, como SELECT AVG(CPUUtilization) FROM SCHEMA('AWS/EC2', InstanceId) GROUP BY InstanceId ORDER BY AVG() DESC LIMIT 10. Em um widget, devolve sempre o Top N do momento, sem lista fixa. Alarmes sobre consultas do Metrics Insights acompanham frotas dinâmicas, inclusive de contas de origem na observabilidade entre contas.

Observabilidade entre contas do CloudWatch

Contas de origem compartilham métricas, logs e traces com uma conta de monitoramento por meio de um sink em cada região. Ela centraliza a visualização e a consulta, não o armazenamento: para guardar logs em um lugar só, use trilha organizacional e subscription filters para um destino central (Data Firehose). Para várias regiões em uma única conta, basta um dashboard com a propriedade region por widget.

X-Ray e ADOT (OpenTelemetry)

Traces distribuídos mostram o tempo de cada requisição por segmento e subsegmento, inclusive chamadas downstream e SQL instrumentado. O ALB não envia dados ao X-Ray, só adiciona o cabeçalho X-Amzn-Trace-Id; a instrumentação fica na aplicação. Para código novo, prefira ADOT: em Java, o agente de autoinstrumentação entra com -javaagent sem mudar o código-fonte, e no Collector cada exporter só funciona se aparecer em service.pipelines (awsemf no pipeline de métricas, awsxray no de traces).

Application Signals e Container Insights

Container Insights coleta CPU, memória, rede e armazenamento por cluster, serviço e task (namespace ECS/ContainerInsights), mas não captura stdout da aplicação nem traces. Application Signals mede volume, latência, falhas e erros por serviço e operação, com SLOs. No ECS, usa o CloudWatch agent como sidecar e um contêiner init com o agente do ADOT; no EKS, o add-on Amazon CloudWatch Observability habilita os dois por padrão.

CloudWatch Database Insights x Enhanced Monitoring

O Database Insights (que substituiu o console do Performance Insights a partir de 31 de julho de 2026) mostra a DB Load por SQL, wait events, hosts e usuários, já no modo Standard. O Enhanced Monitoring usa um agente na instância RDS para coletar métricas do sistema operacional e a lista de processos em intervalos de 1 a 60 segundos, gravando no grupo RDSOSMetrics. No RDS não há acesso ao SO para instalar o CloudWatch agent.

EventBridge: filtrar, formatar, repetir e reprocessar

Event patterns filtram por conteúdo com anything-but, numeric, prefix, suffix, exists e wildcard; o input transformer (input path + input template) só reformata o que chega ao destino. Cada target tem retry policy (até 24 horas e 185 tentativas) e pode ter DLQ, que deve ser fila SQS padrão na mesma região. Um archive guarda eventos a partir da sua criação, e o replay devolve um intervalo ao bus para todas as regras ou só para as escolhidas.

Permissões de quem entrega o evento

Para o EventBridge invocar uma Lambda criada por CLI ou IaC, é preciso add-permission na resource-based policy da função com principal events.amazonaws.com. Notificações do S3 para SNS ou SQS dependem da access policy do destino liberando s3.amazonaws.com com aws:SourceArn e aws:SourceAccount. Alarme do CloudWatch que publica em tópico SNS criptografado exige chave gerenciada pelo cliente com key policy liberando kms:GenerateDataKey* e kms:Decrypt a cloudwatch.amazonaws.com.

Systems Manager Automation em escala

Runbooks aceitam alvos por tag, grupo de recursos ou parâmetro, com rate controls: concurrency limita quantos rodam ao mesmo tempo e error threshold define quando parar. A ação aws:approve pausa até os aprovadores liberarem, e aws:executeScript roda Python ou PowerShell. Em multi-conta, a AutomationAdministrationRole fica na conta central e a AutomationExecutionRole em cada conta-alvo; quem informa uma role a outro serviço precisa de iam:PassRole sobre ela.

Maintenance Windows e EC2Rescue

Maintenance Windows agendam ações disruptivas com cron ou rate e fuso horário, duração máxima e cutoff (tempo antes do fim em que não se iniciam tarefas novas), com tarefas de Run Command, Automation, Lambda ou Step Functions. Para instância inacessível por erro de SO (fstab, SSH), o runbook AWSSupport-ExecuteEC2Rescue cria uma AMI de backup e corrige o volume raiz a partir de uma instância auxiliar. Session Manager depende do SSM Agent conectado.

Tipos de volume EBS e seus limites

gp2 tem linha de base de 3 IOPS por GiB (mínimo 100) e passa disso gastando créditos (BurstBalance). gp3 entrega 3.000 IOPS e 125 MiB/s de base, com IOPS e throughput provisionados sem aumentar o tamanho; a troca é feita com Elastic Volumes sem parar a instância. io2 Block Express dá latência abaixo de 1 ms, 99,999% de durabilidade e até 256.000 IOPS em Nitro; st1 chega a 500 MiB/s para cargas sequenciais e sc1 a 250 MiB/s para dados frios.

Limite de EBS e de rede da instância

O volume entrega o menor valor entre o que ele suporta e o que a instância suporta; muitas instâncias até 4xlarge têm desempenho EBS de base com burst por cerca de 30 minutos a cada 24 horas (EBSIOBalance% e EBSByteBalance%). Na rede, bw_in_allowance_exceeded e bw_out_allowance_exceeded indicam limite agregado do tipo de instância. Um fluxo TCP único fica em 5 Gbps; o ENA Express (SRD) aumenta esse limite, mas precisa estar ativo nas duas pontas.

Instâncias burstable e Lambda

Em T3 no modo standard, CPUCreditBalance zerado limita a CPU à linha de base; o modo unlimited cobra o excedente e compensa só enquanto a média de 24 horas fica abaixo do ponto de equilíbrio, acima disso uma família M sai mais barata. Na Lambda, a CPU é proporcional à memória (128 MB a 10.240 MB; cerca de 1.769 MB equivalem a um vCPU), e mais memória pode reduzir o custo total. O Compute Optimizer só considera memória do EC2 quando o agente publica mem_used_percent no namespace CWAgent com a dimensão InstanceId.

Placement groups

Cluster junta instâncias em uma única AZ para baixa latência (HPC com EFA), de preferência lançadas em uma solicitação e do mesmo tipo; InsufficientInstanceCapacity se resolve parando e iniciando todas e relançando. Partition divide em até 7 partições por AZ com racks próprios e partição visível nos metadados (HDFS, HBase, Cassandra). Spread põe cada instância em hardware distinto, com no máximo 7 instâncias em execução por AZ.

Armazenamento de arquivos certo para cada cenário

EFS é NFS para Linux; com BurstCreditBalance zerado e carga imprevisível, use throughput Elastic, e para vazão alta e constante, Provisioned; no lifecycle, Transition into Standard em On first access traz de volta o arquivo lido. FSx for Windows File Server atende SMB com Active Directory e shadow copies; FSx for Lustre atende HPC Linux com data repository association para o S3; FSx for NetApp ONTAP dá NFS e SMB sobre os mesmos dados. Amazon S3 Files monta um bucket como file system completo em EC2, Lambda, ECS e EKS sem copiar dados.

Desempenho e custo do S3 e de computação

O S3 suporta pelo menos 3.500 escritas e 5.500 leituras por segundo por prefixo; 503 Slow Down pede mais prefixos e retry com backoff exponencial. Multipart upload retoma partes que falham, e a regra AbortIncompleteMultipartUpload remove partes órfãs; Transfer Acceleration usa o endpoint s3-accelerate e exige bucket sem pontos no nome. Intelligent-Tiering serve para acesso desconhecido, sem taxa de recuperação, e Compute Savings Plans cobre EC2 de qualquer família ou região, Fargate e Lambda.

Monitoramento do Amazon Bedrock

O Bedrock publica sozinho métricas no namespace AWS/Bedrock, como Invocations, InvocationLatency, InvocationThrottles, InputTokenCount e OutputTokenCount, prontas para alarmes. O conteúdo de prompts e respostas só é guardado com o model invocation logging, que vem desligado e envia ao CloudWatch Logs e ao S3. O CloudTrail registra a chamada de API, não o corpo do prompt.

Macetes para memorizar

O que o hipervisor não vê

Dentro do SO = agente: memória, disco, processo e log da aplicação só chegam com o CloudWatch agent.

Banco no CloudWatch

Database Insights = SQL; Enhanced Monitoring = SO.

Janela do alarme

Período x pontos = tempo para disparar. 5 min x 3 = 15 min; 1 min com 2 de 3 = 2 a 3 min.

EventBridge

Filtra no pattern, formata no transformer, guarda no archive, repete com retry, sobra na DLQ padrão da mesma região.

Serviço global

Global mora na Virgínia: IAM, Trusted Advisor e outros serviços globais entregam eventos em us-east-1.

Placement groups

Cluster junta, Partition separa por rack, Spread isola uma por uma; os dois últimos param no 7 por AZ.

EBS

gp3 é o padrão barato, io2 é o submilissegundo, st1 é sequencial quente e sc1 é sequencial frio.

Tabela de memorização: não confunda

Quando é...	...e quando é
CloudWatch Database Insights Quero saber qual instrução SQL, wait event, host ou usuário gera a carga do banco (DB Load).	**Enhanced Monitoring** Quero métricas do sistema operacional da instância RDS e a lista de processos com granularidade de até 1 segundo.
Monitoramento detalhado do EC2 Preciso das métricas padrão do hipervisor em 1 minuto em vez de 5.	**CloudWatch agent** Preciso de memória, espaço em disco, métricas por processo (procstat) ou logs de dentro do SO.
Container Insights Quero CPU, memória e rede por cluster, serviço e task para alarmes e scaling.	**Application Signals** Quero latência, erros, falhas e SLOs por serviço e operação, correlacionados com traces.
Alarme de limite estático A métrica tem um teto fixo conhecido, como CPU acima de 80% por 5 minutos.	**Alarme de anomaly detection** A carga varia ao longo do dia e um valor fixo geraria falsos alarmes ou perderia desvios.
gp3 com IOPS provisionados Preciso de mais IOPS sem depender de burst e com o menor custo (latência de um dígito de milissegundo).	**io2 Block Express** Banco crítico pede latência abaixo de 1 ms consistente e durabilidade de 99,999%.

st1 (Throughput Optimized HDD) Carga grande e sequencial acessada com frequência que precisa de até 500 MiB/s.	**sc1 (Cold HDD)** Dados grandes e pouco acessados, com o menor custo e até 250 MiB/s.
Instance store NVMe Cache ou dados temporários recriáveis que pedem IOPS extremos com baixo custo.	**EBS** Dados que precisam sobreviver a parada, hibernação ou encerramento da instância.

Números que caem

Item	Valor
Métricas do EC2: monitoramento básico x detalhado	**5 minutos x 1 minuto**
Retenção de métricas no CloudWatch	**até 15 meses**
Métrica de alta resolução com StorageResolution 1	**dados de 1 segundo ficam disponíveis por 3 horas**
Retry policy de target do EventBridge	**até 24 horas e 185 tentativas**
Compartilhamento de dashboard por e-mail	**até 5 e-mails**
Enhanced Monitoring do RDS	**1, 5, 10, 15, 30 ou 60 segundos**
gp3 de base	**3.000 IOPS e 125 MiB/s**
gp2 de base	**3 IOPS por GiB (mínimo 100)**
io2 Block Express	**latência abaixo de 1 ms, 99,999% de durabilidade, até 256.000 IOPS**
st1 x sc1 por volume	**até 500 MiB/s x até 250 MiB/s**
S3 por prefixo particionado	**pelo menos 3.500 escritas e 5.500 leituras por segundo**
Memória da Lambda	**128 MB a 10.240 MB; cerca de 1.769 MB equivalem a 1 vCPU**
Spread e partition placement group	**máximo de 7 instâncias por AZ (spread) e 7 partições por AZ (partition)**
Fluxo TCP único entre instâncias	**5 Gbps sem ENA Express**
S3 Intelligent-Tiering	**Infrequent após 30 dias sem acesso, Archive Instant após 90; objetos menores que 128 KB não mudam de camada**
Burst de desempenho EBS de instâncias até 4xlarge	**cerca de 30 minutos a cada 24 horas**
Cost allocation tags	**até 24 horas para aparecer e até 24 horas para ativar**
Compute Savings Plans	**compromisso de 1 ou 3 anos, desconto de até 66%**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Ativar o monitoramento detalhado para ver a memória da instância.
Como não cair: O detalhado só muda a frequência para 1 minuto; o hipervisor não enxerga a memória. Memória e disco exigem o CloudWatch agent.

⚠ **Alerta de prova:** Achar que o Compute Optimizer coleta memória sozinho ou que o enhanced infrastructure metrics resolve isso.
Como não cair: Ele só lê memória publicada pelo agente no namespace CWAgent com a dimensão InstanceId; o enhanced infrastructure metrics só amplia o período analisado.

⚠ **Alerta de prova:** Criar alarme com período de 1 minuto sobre CPUUtilization com monitoramento básico.
Como não cair: No básico a métrica chega a cada 5 minutos, então haveria pontos faltando. Case o período com a frequência da métrica.

⚠ **Alerta de prova:** Usar o tratamento de dados ausentes para não notificar durante manutenção.
Como não cair: A métrica continua chegando na manutenção; quem silencia ações em janelas recorrentes são as Alarm Mute Rules.

⚠ **Alerta de prova:** Dar permissão na execution role da Lambda para o EventBridge conseguir invocá-la.
Como não cair: A execution role é o que a função faz; quem invoca precisa estar na resource-based policy da função (add-permission para events.amazonaws.com).

⚠ **Alerta de prova:** Criar a regra do EventBridge para CreateAccessKey do IAM ou para o Trusted Advisor na região onde a empresa opera.
Como não cair: Eventos de serviços globais chegam ao EventBridge em us-east-1; a regra precisa estar lá.

⚠ **Alerta de prova:** Usar fila FIFO ou fila de outra região como DLQ de um target do EventBridge.
Como não cair: A DLQ deve ser fila SQS padrão na mesma região, com resource policy liberando sqs:SendMessage ao events.amazonaws.com.

⚠ **Alerta de prova:** Usar o input transformer para não invocar a função com eventos irrelevantes.
Como não cair: O input transformer só muda o formato entregue; filtrar é papel do event pattern (anything-but, numeric, prefix).

⚠ **Alerta de prova:** Criar um archive agora para reprocessar os eventos das últimas horas.
Como não cair: O archive só guarda eventos recebidos depois de criado; ele precisa existir antes do problema para o replay funcionar.

⚠ **Alerta de prova:** Aumentar os IOPS do volume quando ele entrega menos que o provisionado.
Como não cair: Se EBSIOBalance% zerou, o limite é o desempenho EBS da instância; troque para um tipo de instância com mais desempenho EBS de base.

⚠ **Alerta de prova:** Aumentar o tamanho do gp2 para chegar ao IOPS necessário.
Como não cair: gp2 dá 3 IOPS por GiB e depende de créditos; gp3 com IOPS provisionados resolve sem crescer o disco e sai mais barato.

⚠ **Alerta de prova:** Colocar 8 instâncias em um spread placement group na mesma AZ ou criar 10 partições em uma AZ.
Como não cair: Spread aceita no máximo 7 instâncias em execução por AZ e partition no máximo 7 partições por AZ.

⚠ **Alerta de prova:** Usar reserved concurrency ou provisioned concurrency em $LATEST para eliminar cold start.
Como não cair: Reserved só reserva e limita; provisioned concurrency pré-inicializa ambientes e exige versão publicada ou alias invocado pela API.

⚠ **Alerta de prova:** Exportar logs para o S3 para reduzir o custo do CloudWatch Logs.
Como não cair: A ingestão já foi cobrada; reduza na origem com filtros do agente e use a classe Infrequent Access para logs raramente consultados.

Como cai na prova

📝 **Como cai:** Sintoma de desempenho com várias métricas no enunciado (CPU sobrando, IOPS no teto, saldo de créditos zerado).
Como resolver: Identifique qual recurso está no limite pela métrica citada: wait events de I/O com IOPS no teto pede mais IOPS, não mais CPU; EBSIOBalance% zerado aponta a instância; BurstBalance aponta o gp2; CPUCreditBalance aponta a T3.

📝 **Como cai:** Regra ou notificação que não dispara apesar de o evento acontecer.
Como resolver: Verifique na ordem: o padrão casa (MatchedEvents subindo)? A região é a certa (serviço global em us-east-1)? O destino autoriza quem entrega (resource policy da Lambda, access policy do SNS ou SQS, key policy da chave KMS)?

📝 **Como cai:** Pede monitoramento ou correção com o menor esforço operacional.
Como resolver: Elimine Lambdas e scripts que recriam algo nativo; prefira recursos prontos como Application Insights, add-on do CloudWatch Observability, Service Quotas, input transformer ou runbook gerenciado.

📝 **Como cai:** Alarme com números (milissegundos, porcentagem, períodos).
Como resolver: Converta a unidade da métrica (TargetResponseTime é em segundos: 300 ms = 0,3), confira a frequência da métrica e calcule período x pontos para ver se a janela atende ao pedido.

📝 **Como cai:** Remediação automática a partir de um alarme ou achado.
Como resolver: Monte o caminho alarme ou evento, regra do EventBridge e Systems Manager Automation (ou Lambda); se pedir aprovação humana, aws:approve; se pedir janela fixa, Maintenance Window.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
memória, espaço em disco ou um processo específico da instância	**CloudWatch agent (mem_used_percent, disk_used_percent, procstat) com configuração no Parameter Store**
Top 10 dinâmico de instâncias no dashboard	**Widget com consulta do Metrics Insights usando ORDER BY e LIMIT**
qual consulta SQL pesa no banco, wait events, DB Load	**CloudWatch Database Insights**
processos do SO de uma instância RDS em segundos	**Enhanced Monitoring (1 a 60 segundos)**
BurstBalance chegando a 0% em gp2 pequeno	**Migrar para gp3 com Elastic Volumes e provisionar os IOPS**
CPUCreditBalance zerado com uso alto e constante	**Instância de desempenho fixo (família M); unlimited só para picos ocasionais**
503 Slow Down no S3	**Distribuir gravações em mais prefixos e manter retry com backoff exponencial**
Lambda + RDS com too many connections e CPU folgada	**RDS Proxy com credenciais no Secrets Manager**
reprocessar eventos já entregues depois de corrigir um bug	**Replay de um archive do EventBridge que já existia**
carga com padrão que muda ao longo do dia	**Alarme de anomaly detection, com remediação por EventBridge (Alarm State Change) para SSM Automation**

agir só quando 5XX e latência piorarem juntos	**Alarme composto com AND e ação de OpsItem ou Lambda**
avisar antes de atingir uma quota de serviço	**Alarme em AWS/Usage com SERVICE_QUOTA() e pedido de aumento no Service Quotas**
auditores sem conta AWS precisam ver um dashboard	**Compartilhamento do dashboard por e-mail (até 5), com senha individual**
latência por requisição e por dependência em ECS, EKS ou EC2	**Traces com X-Ray via ADOT ou Application Signals**

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

- Memória e disco do EC2 só existem no CloudWatch com o agente; detalhado é só frequência.
- Eventos de serviços globais (IAM, Trusted Advisor) chegam ao EventBridge em us-east-1.
- Quem invoca ou publica precisa de permissão no destino (resource policy da Lambda, access policy do SNS ou SQS, key policy da chave gerenciada pelo cliente); a execution role não resolve isso.
- Latência percebida pelo cliente se mede em percentis (p95, p99) de TargetResponseTime, não em média nem em CPU.
- Tag só aparece no Cost Explorer depois de ativada como cost allocation tag na conta de gerenciamento.

Essenciais para a prova (checklist)

☐ Sei quais métricas exigem o CloudWatch agent e como distribuí-lo em frota pelo Systems Manager e Parameter Store.

☐ Sei calcular a janela de um alarme (período x pontos de dados) e casar o período com a frequência da métrica.

☐ Sei quando usar alarme composto, anomaly detection, Alarm Mute Rules e alarmes sobre consultas do Metrics Insights.

☐ Sei diferenciar Container Insights, Application Signals, X-Ray com ADOT, Database Insights e Enhanced Monitoring.

☐ Sei configurar EventBridge com event pattern, input transformer, retry policy, DLQ padrão e archive com replay.

☐ Sei diagnosticar gargalos de EBS pelas métricas BurstBalance, EBSIOBalance% e IOPS no teto e escolher gp3, io2, st1 ou sc1.

☐ Sei escolher entre placement groups cluster, partition e spread e seus limites por AZ.

☐ Sei escolher EFS, FSx for Windows, FSx for Lustre, FSx for NetApp ONTAP ou S3 Files pelo protocolo e pela carga.

- ☐ Sei montar uma remediação automática com EventBridge e Systems Manager Automation, inclusive multi-conta e com aprovação.
- ☐ Sei quais métricas o Bedrock publica e que o conteúdo das chamadas exige model invocation logging.

Questões de revisão do capítulo

R1.1 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Uma empresa opera uma aplicação com Auto Scaling group, Application Load Balancer e Amazon RDS Multi-AZ. A fatura do CloudWatch Logs cresceu com logs de debug pouco úteis, e a equipe precisa identificar gargalos por camada, com alarmes acionáveis e correlação entre métricas, logs e traces. Qual abordagem atende com o menor esforço operacional contínuo?

A. Enviar os logs por subscription filter ao Amazon Data Firehose e ao S3 para consulta no Athena, criando dashboards e alarmes com as métricas padrão de cada camada
B. Habilitar VPC Flow Logs com tráfego ACCEPT e REJECT nas VPCs, criar filtros de métrica para os padrões relevantes e configurar alarmes do CloudWatch sobre essas métricas
C. Habilitar detailed monitoring nas instâncias e Enhanced Monitoring no RDS, montar dashboards com as métricas de cada camada e exportar os logs para o S3 com lifecycle de 7 dias
D. Ativar o Application Insights, filtrar os logs na origem com filtros do CloudWatch agent, usar a classe Infrequent Access nos logs pouco consultados e habilitar o X-Ray

R1.2 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Uma API crítica publica no Amazon CloudWatch, a cada minuto, uma métrica personalizada com a taxa de erros. O time de operações quer ser notificado quando a taxa ficar acima de 5% por 5 minutos seguidos, mas não quer receber alertas durante as janelas de manutenção, que ocorrem todo domingo das 2h às 4h. Qual configuração atende aos requisitos com o menor esforço?

A. Criar um alarme com período de 1 minuto e 5 de 5 pontos de dados acima de 5% e configurar o tratamento de dados ausentes como notBreaching durante as janelas
B. Criar um alarme com período de 1 minuto e 5 de 5 pontos de dados acima de 5%, notificando o SNS, e uma alarm mute rule com expressão cron para as janelas
C. Criar um alarme com período de 1 minuto e 1 de 1 ponto de dados acima de 5% e uma assinatura do SNS com política de filtro que descarta mensagens de domingo
D. Criar um alarme com período de 5 minutos e 5 de 5 pontos de dados acima de 5%, notificando o SNS, e uma alarm mute rule com expressão cron para as janelas

R1.3 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Uma aplicação crítica roda em um Auto Scaling group atrás de um Application Load Balancer. Usuários relatam falhas intermitentes, mas a CPU das instâncias está normal. O time de operações quer ser alertado automaticamente quando houver picos de erros HTTP 5xx, gerados pelos targets ou pelo próprio balanceador, ou quando a latência dos targets subir. Qual configuração atende a esse requisito?

A. Criar um alarme do CloudWatch em RequestCount do ALB com limiar fixo baseado no pico histórico, notificando um tópico do SNS quando o limiar for excedido
B. Criar alarmes do CloudWatch em HTTPCode_Target_5XX_Count, HTTPCode_ELB_5XX_Count e TargetResponseTime do ALB, notificando um tópico do SNS
C. Habilitar access logs do ALB em um bucket S3 e criar uma tabela no Athena para consultar os campos elb_status_code e target_processing_time
D. Criar alarmes do CloudWatch em UnHealthyHostCount de cada target group, notificando um tópico do SNS quando algum target ficar não íntegro

R1.4 · Domínio 1: Monitoramento, registro em log, análise, correção e otimização de desempenho

Usuários relatam lentidão intermitente em uma aplicação que roda em instâncias Amazon EC2 Linux gerenciadas pelo Systems Manager. O time de operações quer verificar no CloudWatch se o problema vem de CPU, de memória ou de espaço em disco, sem acessar as instâncias por SSH e usando ferramentas da AWS. Qual abordagem atende a esses requisitos?

A. Habilitar o AWS Compute Optimizer, que coleta a memória e o disco das instâncias para recomendar um tipo de instância mais adequado
B. Instalar o agente do CloudWatch pelo Systems Manager e coletar mem_used_percent e disk_used_percent junto às métricas de CPU do EC2
C. Criar um dashboard com as métricas padrão do namespace AWS/EC2, como CPUUtilization, EBSReadOps e MemoryUtilization das instâncias
D. Habilitar o monitoramento detalhado do EC2, que passa a publicar as métricas de memória e de uso de disco das instâncias a cada minuto

Respostas comentadas

R1.1. D) Ativar o Application Insights, filtrar os logs na origem com filtros do CloudWatch agent, usar a classe Infrequent Access nos logs pouco consultados e habilitar o X-Ray

Por quê: Os filtros include e exclude do CloudWatch agent impedem que linhas sem valor saiam do host, e a classe de log Infrequent Access cobra menos pela ingestão de logs consultados raramente (sem suporte a filtros de métrica). O CloudWatch Application Insights configura métricas, logs e alarmes dinâmicos para EC2, ELB e RDS e cria dashboards com anomalias correlacionadas; o X-Ray completa com traces das requisições.

✗ A) Incorreta. A subscription filter não reduz o volume ingerido no CloudWatch Logs e acrescenta outro pipeline, sem correlação automática entre métricas, logs e traces.

✗ B) Incorreta. Flow Logs adicionam ainda mais ingestão de logs e mostram tráfego de rede, não gargalos da aplicação ou do banco.

✗ C) Incorreta. Mais métricas detalhadas aumentam o custo e não trazem traces; exportar para o S3 não reduz a ingestão no CloudWatch Logs, que já foi cobrada.

> **Dica de prova:** Reduzir custo de logs: filtrar na origem (filtros do agente) e usar a classe Infrequent Access para logs raramente consultados. Exportar para o S3 depois não desfaz a ingestão.

R1.2. B) Criar um alarme com período de 1 minuto e 5 de 5 pontos de dados acima de 5%, notificando o SNS, e uma alarm mute rule com expressão cron para as janelas

Por quê: O alarme avalia N períodos e entra em ALARM quando M deles violam o limite; com período de 60 segundos e 5 de 5 pontos, a condição equivale a 5 minutos acima de 5%. As Alarm Mute Rules definem janelas, por expressão cron ou horário, em que as ações dos alarmes-alvo ficam silenciadas, enquanto o CloudWatch continua avaliando o estado; se o alarme ainda estiver em ALARM quando a janela terminar, a ação é executada.

✗ A) Incorreta. A métrica continua sendo publicada durante a manutenção, então não há dados ausentes; o tratamento de dados ausentes não suprime notificações.

✗ C) Incorreta. Com 1 de 1 ponto, um único minuto acima de 5% já notifica, e as políticas de filtro do SNS avaliam atributos da mensagem, não o dia ou o horário.

✗ D) Incorreta. Com período de 5 minutos, 5 de 5 pontos exigem 25 minutos acima do limite, o que atrasa demais a notificação.

> **Dica de prova:** Silenciar alertas em manutenção recorrente: Alarm Mute Rules (ou supressão de ações em alarme composto). Tratamento de dados ausentes não é supressão.

R1.3. B) Criar alarmes do CloudWatch em HTTPCode_Target_5XX_Count, HTTPCode_ELB_5XX_Count e TargetResponseTime do ALB, notificando um tópico do SNS

Por quê: O ALB publica no CloudWatch métricas separadas para códigos 5xx gerados pelos targets (HTTPCode_Target_5XX_Count) e pelo próprio balanceador (HTTPCode_ELB_5XX_Count), além de TargetResponseTime. Alarmes nessas métricas detectam a degradação percebida pelo cliente mesmo com CPU normal e notificam por SNS.

✗ A) Incorreta. RequestCount mede volume de requisições, não erros nem latência; a aplicação pode falhar com tráfego normal e o alarme não dispararia.

✗ C) Incorreta. Access logs servem para análise detalhada posterior; consultas no Athena não geram alertas automáticos quando os erros acontecem.

✗ D) Incorreta. Targets podem passar no health check e ainda retornar 5xx em algumas rotas, e erros gerados pelo próprio balanceador não aparecem nessa métrica.

> **Dica de prova:** 5xx do target x 5xx do ELB são métricas diferentes: monitore as duas. Latência = TargetResponseTime. Alerta automático = alarme do CloudWatch + SNS; access logs = análise posterior.

R1.4. B) Instalar o agente do CloudWatch pelo Systems Manager e coletar mem_used_percent e disk_used_percent junto às métricas de CPU do EC2

Por quê: As métricas padrão do EC2 vêm do hipervisor e incluem CPU, rede e operações de disco, mas não memória nem espaço usado no sistema de arquivos. O agente do CloudWatch coleta métricas do sistema operacional, como mem_used_percent e disk_used_percent, e pode ser distribuído e configurado pelo Systems Manager, sem acesso SSH.

✗ A) Incorreta. O Compute Optimizer gera recomendações de dimensionamento e só considera memória quando o agente do CloudWatch a publica; ele não coleta essas métricas.

✗ C) Incorreta. O namespace AWS/EC2 não tem métrica de memória, porque o hipervisor não enxerga o uso de memória dentro do sistema operacional.

✗ D) Incorreta. O monitoramento detalhado só muda a frequência das métricas padrão para 1 minuto; memória e uso de disco não fazem parte delas.

> **Dica de prova:** Memória e espaço em disco no CloudWatch = agente do CloudWatch. Monitoramento detalhado muda só a frequência (1 minuto).

02

CAPÍTULO 2 · 22% DA PROVA

Domínio 2: Confiabilidade e continuidade dos negócios

Neste domínio eu preciso casar requisitos de RTO, RPO e custo com a solução certa: Multi-AZ, réplicas, backups, DR entre regiões e escalabilidade automática. A prova cobra o comportamento exato de cada serviço: o que restaura em recurso novo, o que é síncrono ou assíncrono, o que escala leitura e o que escala escrita. Auto Scaling aparece em muitos detalhes (scheduled x predictive, warmup x grace period, lifecycle hooks, health check do ELB), assim como AWS Backup, versionamento e replicação do S3. Onde mais erro: achar que a standby do Multi-AZ lê, que o Aurora Auto Scaling mexe no writer e que o PITR volta a própria tabela ou instância. Também preciso ordenar as quatro estratégias de DR por custo e tempo sem hesitar.

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

Tópicos-chave

Estratégias de DR

Da mais barata para a mais cara: backup and restore (RPO e RTO em horas, recriação por IaC), pilot light (dados replicados sempre e aplicação desligada, RTO em dezenas de minutos), warm standby (versão reduzida e funcional rodando, RTO em minutos) e multi-site active/active (RTO quase zero, gravação em várias regiões). RPO de minutos exige replicação contínua, não cópia periódica.

RDS Multi-AZ x réplica de leitura

Multi-AZ de instância mantém uma standby síncrona em outra zona, com failover automático para o mesmo endpoint; a standby não atende leituras. Réplicas de leitura usam replicação assíncrona, escalam leituras e têm o atraso medido por ReplicaLag; a réplica entre regiões pode ser promovida a instância independente no desastre. Multi-AZ fica dentro da região e não protege contra desastre regional.

Backups e restauração do RDS

Backups automáticos têm retenção de 0 a 35 dias e sustentam o point-in-time restore com logs de transação enviados a cada cinco minutos. Toda restauração cria uma nova instância com novo endpoint; para guarda longa, use snapshots manuais, que não expiram. A replicação de backups automáticos entre regiões permite PITR na região de DR sem réplica rodando, e o backup contínuo do AWS Backup chega aos 5 minutos mais recentes.

Aurora: capacidade e recuperação

Aurora Serverless v2 escala em ACUs (cerca de 2 GiB cada, máximo de 256, mínimo pode ser 0 com pausa) com faixa definida no cluster. Aurora Auto Scaling ajusta só o número de réplicas (0 a 15) e só remove as que criou; custom endpoints (até cinco) dirigem cargas a réplicas escolhidas. Backtrack volta o próprio cluster Aurora MySQL em até 72 horas se tiver sido habilitado na criação ou na restauração do cluster, cloning usa copy-on-write e Global Database replica com latência típica abaixo de um segundo.

DynamoDB: capacidade

On-demand é para tráfego novo, imprevisível ou com picos súbitos e acomoda até o dobro do pico anterior; provisionado com auto scaling serve para tráfego previsível e gradual, com target utilization de 20% a 90%. Cada GSI tem capacidade própria, e um GSI com WCU insuficiente faz throttling na tabela base. Throttling com capacidade sobrando indica partição quente: a correção é uma chave de partição de alta cardinalidade.

DynamoDB: proteção e alcance global

O PITR restaura em segundos de granularidade dentro de 1 a 35 dias, sempre em uma nova tabela, sem levar auto scaling, Streams, TTL, alarmes e tags. Para guarda longa, backups sob demanda ou AWS Backup. Global tables permitem gravação local em várias regiões, mas replicam também os erros; DAX acelera só leituras eventualmente consistentes em microssegundos.

ElastiCache: escala e disponibilidade

Com cluster mode desabilitado há um shard: réplicas e reader endpoint escalam leitura, e escrita vai só ao primário. Escrita ou memória além de um nó pedem cluster mode habilitado com mais shards (resharding online) ou nó maior; Evictions altas indicam falta de memória. Multi-AZ com réplicas em outras AZs garante failover contra perda de zona, e o ElastiCache Serverless dispensa escolher nós, cobrando por dados e ECPUs.

ElastiCache: backup, cache e global

Um backup de Valkey ou Redis OSS sempre é restaurado em um novo cache ou cluster, com novo endpoint; não há PITR. Global Datastore cria secundários somente leitura em outras regiões, promovíveis em DR. Na estratégia de cache, lazy loading guarda só o que é lido, write-through evita dado velho e TTL expira o que não é usado.

Políticas de escalabilidade do Auto Scaling

Target tracking mantém uma métrica no alvo; step scaling dá ajustes proporcionais por faixa sem esperar cooldown; simple scaling faz um ajuste e espera. Scheduled scaling usa cron e fuso para picos em horário conhecido, e predictive scaling antecipa padrões do histórico (precisa de dados). Para workers de SQS, escale por backlog por instância com alvo igual à latência aceitável dividida pelo tempo de processamento.

Ciclo de vida das instâncias no ASG

O health check grace period evita terminar instâncias novas antes do bootstrap; o default instance warmup tira instâncias novas das métricas agregadas até estabilizarem; warm pool mantém instâncias pré-inicializadas. Lifecycle hooks pausam em Pending:Wait ou Terminating:Wait (padrão de uma hora, ajustável pelo heartbeat timeout) até complete-lifecycle-action. Por padrão o ASG usa só status checks do EC2; ative os health checks do ELB para trocar instâncias que a aplicação marcou como unhealthy.

Health checks e drenagem no ALB

O matcher padrão é 200; o health check não segue redirecionamento e uma resposta 302 gera Target.ResponseCodeMismatch. Timeout no health check aponta security group: entrada nas instâncias a partir do SG do ALB e saída do SG do ALB para a porta. O deregistration delay (padrão de 300 segundos) deve superar a requisição mais longa, e com todos os alvos unhealthy o ALB faz fail-open.

Route 53 para failover

Roteamento de failover responde com o primário enquanto o health check dele estiver saudável. Calculated health checks combinam filhos com limiar de quantos precisam estar saudáveis; para IPs privados, o health check monitora um alarme do CloudWatch, porque os health checkers ficam fora da VPC e usam as faixas ROUTE53_HEALTHCHECKS do ip-ranges.json. No failover regional, reduza o TTL antes, promova o banco e só então vire o DNS.

Application Recovery Controller e testes de DR

Zonal shift tira o tráfego de uma AZ prejudicada para as demais da região, de forma temporária (expiração de até três dias, extensível), em ALB, NLB, ASG ou EKS habilitados. Routing controls fazem failover regional decidido por pessoas, com data plane em cinco regiões. Para provar RTO e RPO, combine experimentos do AWS FIS com runbooks do SSM Automation agendados.

AWS Backup

Planos têm regras (frequência, janela, retenção, cópias) e resource assignments, e a seleção por tag pega recursos novos sozinha. Regras de cópia levam recovery points a outras regiões ou contas, e o Vault Lock em modo compliance impede alterar ou apagar até pelo root. Backup contínuo dá PITR de até 35 dias, cold storage exige no mínimo 90 dias, Windows VSS dá consistência de aplicação, o Audit Manager gera relatórios de conformidade e backup policies do Organizations padronizam planos imutáveis nas contas.

Versionamento do S3

Sobrescritas criam versões e um DELETE sem version ID cria um delete marker; apagar o delete marker desfaz a exclusão. Depois de habilitado, o versionamento só pode ser suspenso, e suspender não apaga versões existentes. MFA Delete é habilitado pelo root, e NoncurrentVersionExpiration com NewerNoncurrentVersions controla o custo das versões antigas.

Replicação do S3

Exige versionamento nos buckets de origem e de destino, e a replicação de delete markers fica desabilitada por padrão, protegendo o destino. CRR com S3 Replication Time Control replica novos objetos em 15 minutos com SLA; objetos já existentes vão com Batch Replication, sem RTC. Multi-Region Access Point com replicação e failover controls dá um endpoint global ativo-passivo.

Snapshots de EBS e outros recursos

O Data Lifecycle Manager cria e expira snapshots de EBS e AMIs (intervalos de 1 a 24 horas ou cron); snapshots são regionais e restauram em qualquer AZ. Multi-Attach só anexa io1 ou io2 a instâncias da mesma AZ e não protege contra perda da zona. Snapshot RDS criptografado só vai a outra conta se recriptografado com chave gerenciada pelo cliente; aws/rds nunca sai da conta.

Resiliência de contêineres e filas

No EKS, HPA escala pods e Karpenter, Cluster Autoscaler ou EKS Auto Mode escalam nós; nós em várias AZs com topologySpreadConstraints por zona sobrevivem à perda de AZ. No ECS sobre EC2, capacity provider com managed scaling cria instâncias para tarefas em PROVISIONING. No SQS, o visibility timeout deve superar o processamento, FIFO deduplica e a DLQ isola mensagens problemáticas.

Macetes para memorizar

Estratégias de DR

B-P-W-A: Backup (horas), Pilot (dezenas de minutos), Warm (minutos), Active (quase zero). Quanto mais rápido, mais caro.

Multi-AZ x réplica

Multi-AZ é Seguro e Síncrono; Réplica é para Relatório e Região.

Restaurações

Restore nasce novo: RDS, DynamoDB e ElastiCache sempre criam recurso e endpoint novos. Só o Backtrack volta no lugar.

Concurrency da Lambda

Reserved Reserva e Restringe; Provisioned Pré-aquece.

Tempos do Auto Scaling

Grace = saúde; Warmup = métrica; Cooldown = espaçar ajustes; Hook = pausar para tarefa.

ElastiCache

Leitura pede réplica; escrita ou memória pedem shard.

Versionamento do S3

Versionamento não desliga, só suspende; delete marker apagado traz o objeto de volta.

Tabela de memorização: não confunda

Quando é...	...e quando é
RDS Multi-AZ Alta disponibilidade na região com cópia síncrona e failover automático no mesmo endpoint.	**Réplica de leitura** Escalar leituras (relatórios) ou ter DR entre regiões, aceitando replicação assíncrona e promoção manual.
Pilot light RPO de minutos com dados replicados, mas RTO de dezenas de minutos ou horas permite ligar a aplicação no failover.	**Warm standby** RTO de minutos com custo controlado: uma versão reduzida da produção já roda na outra região.
Zonal shift Falha parcial em uma AZ que o health check não pega; desviar tráfego para as outras AZs da região.	**Routing controls** Failover entre regiões decidido por pessoas, pelo data plane mais resiliente.
Scheduled scaling Pico em horário fixo e conhecido, inclusive em grupo recém-criado sem histórico.	**Predictive scaling** Padrão diário ou semanal recorrente que pode ser aprendido do histórico de métricas.
Health check grace period Instâncias novas estão sendo terminadas em loop antes de concluir o bootstrap.	**Default instance warmup** Instâncias novas puxam a métrica agregada e causam scale-out exagerado.
DynamoDB on-demand Tráfego desconhecido ou com picos repentinos.	**Provisionado com auto scaling** Tráfego previsível que varia de forma gradual e precisa de custo menor.
Réplicas no ElastiCache (cluster mode desabilitado) Muita leitura, sem particionar chaves; usar o reader endpoint.	**Shards (cluster mode habilitado)** Escrita saturando o primário ou dados maiores que a memória de um nó.

Números que caem

Item	Valor
Aurora Serverless v2	**cada ACU com cerca de 2 GiB; máximo de 256 ACUs; mínimo pode ser 0 com pausa**
Aurora Auto Scaling	**de 0 a 15 réplicas**
Aurora custom endpoints	**até cinco por cluster provisionado**
Aurora Backtrack	**janela máxima de 72 horas**
Retenção de backups automáticos do RDS	**0 a 35 dias (0 desativa)**
Envio de logs de transação do RDS ao S3	**a cada cinco minutos**
PITR do DynamoDB	**período de 1 a 35 dias**
Target utilization do auto scaling do DynamoDB	**20% a 90%**
Partição física do DynamoDB	**da ordem de 3.000 RCU e 1.000 WCU**
DynamoDB on-demand	**acomoda até o dobro do pico anterior**
Lifecycle hook do ASG	**tempo padrão de uma hora, ajustável pelo heartbeat timeout**
Deregistration delay padrão	**300 segundos**
Health check grace period no console	**300 segundos**
Zonal shift do ARC	**expiração de até três dias, extensível**
S3 Replication Time Control	**15 minutos com SLA**
AWS Backup para S3	**snapshots com retenção de até 99 anos; contínuo com PITR de até 35 dias**
Cold storage do AWS Backup	**mínimo de 90 dias**
NewerNoncurrentVersions	**de 1 a 100**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Habilitar o Aurora Auto Scaling para aumentar a classe do writer nos picos.
Como não cair: Ele só adiciona ou remove réplicas Aurora; para CPU e memória do writer acompanharem a carga, use Aurora Serverless v2 com faixa de ACUs.

⚠ **Alerta de prova:** Habilitar storage autoscaling para ganhar CPU e memória.
Como não cair: Storage autoscaling só aumenta armazenamento (no RDS, com maximum storage threshold); não muda CPU nem memória.

⚠ **Alerta de prova:** Usar EBS Multi-Attach para sobreviver à falha de uma AZ.
Como não cair: Multi-Attach só funciona entre instâncias da mesma AZ; snapshots regionais (DLM ou AWS Backup) permitem recriar em outra zona.

⚠ **Alerta de prova:** Apontar relatórios para a standby do Multi-AZ.
Como não cair: A standby de uma implantação Multi-AZ de instância não aceita leituras; leitura escalada é réplica de leitura.

⚠ **Alerta de prova:** Restaurar o PITR sobre a tabela, instância ou cluster existente.
Como não cair: RDS, DynamoDB e ElastiCache restauram sempre em recurso novo com novo endpoint; só o Backtrack do Aurora MySQL volta o próprio cluster.

⚠ **Alerta de prova:** Usar global tables, réplicas ou clone como backup contra exclusão acidental.
Como não cair: Réplicas recebem as mesmas gravações, inclusive as erradas, e o clone reflete o estado atual; para voltar no tempo, PITR ou backup.

⚠ **Alerta de prova:** Configurar target utilization de 95% no auto scaling do DynamoDB.
Como não cair: O valor aceito vai de 20% a 90%, e alvo alto deixa menos folga nos picos.

⚠ **Alerta de prova:** Usar provisioned concurrency para limitar conexões que a Lambda abre no banco.
Como não cair: Provisioned concurrency pré-aquece e não limita; reserved concurrency define o teto da função sem mexer na cota da conta.

⚠ **Alerta de prova:** Ativar failover automático no ElastiCache sem Multi-AZ achando que resiste à perda da zona.
Como não cair: Sem Multi-AZ não há garantia de réplicas em AZs diferentes; habilite Multi-AZ com réplicas em outras zonas.

⚠ **Alerta de prova:** Aumentar o grace period para corrigir scale-out exagerado causado por instâncias novas.
Como não cair: Grace period é para health check; quem tira instâncias novas das métricas é o default instance warmup.

⚠ **Alerta de prova:** Aumentar timeout ou tentativas do health check quando a página responde 302.
Como não cair: O health check não segue redirecionamento; aponte para um path que responda 200 ou inclua 302 no matcher.

⚠ **Alerta de prova:** Confiar na ação recover do EC2 para sobreviver à falha da AZ.
Como não cair: A instância recuperada fica na mesma AZ; use um ASG com mínimo, máximo e desejado 1 em várias AZs.

⚠ **Alerta de prova:** Suspender o versionamento para apagar as versões antigas e reduzir custo.
Como não cair: Suspender não apaga nada; use lifecycle com NoncurrentVersionExpiration e NewerNoncurrentVersions.

⚠ **Alerta de prova:** Configurar no plano do AWS Backup a transição para cold storage e a expiração do recovery point menos de 90 dias depois dela.
Como não cair: Em cold storage o mínimo é de 90 dias: a expiração deve ocorrer pelo menos 90 dias após a transição (por exemplo, transição após 30 dias e retenção de 7 anos).

Como cai na prova

📝 **Como cai:** Cenário com RTO, RPO e restrição de custo para DR entre regiões.
Como resolver: Classifique o RPO (horas permite backup periódico; minutos exige replicação contínua) e o RTO (horas, dezenas de minutos, minutos ou zero) e escolha a estratégia mais barata que atenda aos dois; descarte Multi-AZ, que não sai da região.

📝 **Como cai:** Erro humano apagou ou corrompeu dados e é preciso voltar a um ponto no tempo.
Como resolver: Identifique o serviço e o mecanismo (PITR, Backtrack, versionamento, backup); lembre que a restauração cria recurso novo e que a aplicação precisa ser repontada; descarte réplicas e global tables, que já têm o erro.

📝 **Como cai:** Auto Scaling se comportando mal (lento, exagerado, terminando instâncias).
Como resolver: Mapeie o sintoma ao ajuste: lento para pico conhecido pede scheduled; pico recorrente com bootstrap lento pede predictive e warm pool; terminação em loop pede grace period; scale-out exagerado pede default instance warmup; reação lenta do alarme pede período menor e M de N.

📝 **Como cai:** Banco ou cache saturado e várias opções de escala.
Como resolver: Separe leitura de escrita e memória: leitura vai para réplicas e endpoints de leitura; escrita e memória pedem escala vertical, shards ou Serverless; picos imprevisíveis pedem on-demand ou Serverless.

📝 **Como cai:** Alvos unhealthy, timeouts ou failover de DNS que não acontece.
Como resolver: Confira security groups (entrada e saída), matcher e path do health check, faixas dos health checkers do Route 53 no firewall e se o registro primário tem health check associado.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
RTO e RPO em horas e menor custo	**Backup and restore com cópias para a região de DR e recriação por CloudFormation**
RPO de minutos e RTO de horas com aplicação desligada	**Pilot light com réplica assíncrona do banco**
RTO próximo de zero e gravação nas duas regiões	**Multi-site active/active com DynamoDB global tables e Route 53 com health checks**
backups imutáveis que nem o root apaga, em outra região ou conta	**AWS Backup com Vault Lock em modo compliance e regra de cópia**
FreeStorageSpace zerando no RDS	**Storage autoscaling com maximum storage threshold**
desfazer DELETE recente no mesmo cluster Aurora MySQL em minutos	**Backtrack**
cópia rápida e barata de produção Aurora para testes	**Aurora cloning (copy-on-write)**
throttling com capacidade sobrando na tabela	**Partição quente ou GSI sem WCU suficiente (WriteThrottleEvents do índice)**
Evictions altas no ElastiCache	**Falta de memória: nó maior ou mais shards; réplicas não ajudam**
fazer algo antes de a instância do ASG ser encerrada	**Lifecycle hook de terminação com heartbeat timeout e complete-lifecycle-action**
requisições cortadas durante o scale-in	**Deregistration delay maior que a requisição mais longa**
health check do Route 53 para recurso com IP privado	**Health check baseado em alarme do CloudWatch**
replicação S3 com prazo garantido por SLA	**CRR com S3 Replication Time Control (15 minutos)**
usuário recuperando versão anterior de arquivo no FSx for Windows	**Shadow copies e Versões anteriores no Windows Explorer**

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

- Multi-AZ protege a zona, não a região; para desastre regional, réplica entre regiões, Global Database, global tables ou cópia de backups.
- Restauração de RDS, DynamoDB e ElastiCache sempre cria recurso novo: a aplicação precisa ser repontada e configurações como auto scaling e alarmes refeitas.
- A ordem do failover regional é TTL baixo antes, promover o banco e só então virar o DNS.
- AWS Backup para S3 exige versionamento; replicação S3 exige versionamento nas duas pontas.
- O ASG só troca instâncias que o ALB marcou como unhealthy se os health checks do ELB estiverem ativos no grupo.

Essenciais para a prova (checklist)

☐ Sei ordenar backup and restore, pilot light, warm standby e active/active por custo, RTO e RPO.

☐ Sei diferenciar RDS Multi-AZ, réplica de leitura, réplica entre regiões e replicação de backups automáticos.

☐ Sei quando usar Aurora Serverless v2, Aurora Auto Scaling, custom endpoints, Backtrack, cloning e Global Database.

☐ Sei escolher entre DynamoDB on-demand e provisionado e diagnosticar partição quente e GSI com pouca WCU.

☐ Sei escalar ElastiCache para leitura, escrita ou memória e configurar Multi-AZ e Global Datastore.

☐ Sei escolher a política de Auto Scaling e o ajuste certo entre grace period, warmup, warm pool e lifecycle hook.

☐ Sei configurar health checks do ALB e do Route 53, inclusive calculated e baseados em alarme.

☐ Sei montar um plano do AWS Backup com seleção por tag, cópia entre regiões ou contas, Vault Lock e cold storage.

☐ Sei recuperar objetos no S3 com versionamento e configurar replicação com RTC e Multi-Region Access Point.

☐ Sei quando usar zonal shift, routing controls e testes com AWS FIS e SSM Automation.

Questões de revisão do capítulo

R2.1 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma instância RDS for MySQL de 500 GiB recebe dados de ingestão em ritmo irregular. Duas vezes neste ano a métrica FreeStorageSpace chegou a zero de madrugada, a instância entrou em storage-full e a aplicação parou até alguém aumentar o armazenamento. A equipe quer que o espaço cresça sozinho quando necessário, com um teto de custo e o menor esforço operacional. O que deve ser feito?

A. Criar uma réplica de leitura com armazenamento maior e promovê-la quando FreeStorageSpace ficar abaixo de 10%
B. Criar um alarme em FreeStorageSpace que acione uma Lambda para chamar ModifyDBInstance com 20% a mais de armazenamento alocado
C. Migrar o armazenamento da instância para Provisioned IOPS (io2), que expande o volume à medida que os dados crescem
D. Habilitar o storage autoscaling na instância e definir o maximum storage threshold como limite do crescimento

R2.2 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma aplicação web roda em uma região primária e tem um ambiente de espera em outra região, pronto para receber tráfego. A empresa quer que, se a aplicação primária ficar indisponível, o DNS passe a direcionar os usuários para a região secundária de forma automática. Qual configuração atende a esse requisito?

A. Um Application Load Balancer na região primária com targets das duas regiões e verificação de integridade no target group
B. Roteamento de failover no Route 53, com health check no registro primário e o registro secundário na outra região
C. Roteamento por latência no Route 53 entre as duas regiões, com TTL de 24 horas nos registros de cada região
D. Roteamento ponderado no Route 53 com peso 100 para a primária e 0 para a secundária, sem health checks associados

R2.3 · Domínio 2: Confiabilidade e continuidade dos negócios

Um cluster do Amazon ElastiCache (Redis OSS) com modo cluster desativado tem um nó primário e uma réplica na mesma zona de disponibilidade. Em uma falha recente dessa AZ, a aplicação ficou sem cache até uma intervenção manual. Como aumentar a disponibilidade com o menor esforço?

A. Colocar réplicas em outras AZs e ativar Multi-AZ com failover automático no grupo de replicação
B. Criar snapshots automáticos diários do cluster e restaurá-los em um novo cluster em outra AZ quando houver falha
C. Aumentar o tipo de nó do primário para suportar mais conexões durante os picos de acesso
D. Ativar o failover automático sem Multi-AZ, mantendo a réplica na mesma AZ do nó primário

R2.4 · Domínio 2: Confiabilidade e continuidade dos negócios

Uma empresa cria backups de volumes Amazon EBS e bancos Amazon RDS com o AWS Backup. A segurança exige que esses backups não possam ser excluídos por nenhum administrador, nem pelo usuário root, durante a retenção, e que exista uma cópia em outra região para recuperação de desastres. Qual configuração atende a esses requisitos?

A. Criar uma regra de cópia no plano do AWS Backup para um cofre em outra região, com Vault Lock em modo compliance nos cofres
B. Usar o Data Lifecycle Manager com cópia entre regiões dos snapshots do EBS e manter os backups automáticos do RDS por 35 dias
C. Manter o plano do AWS Backup na mesma região com retenção de 35 dias e uma política IAM que nega a exclusão aos operadores
D. Exportar os snapshots para um bucket do Amazon S3 com replicação entre regiões, versionamento e MFA Delete habilitados

Respostas comentadas

R2.1. D) Habilitar o storage autoscaling na instância e definir o maximum storage threshold como limite do crescimento

Por quê: O storage autoscaling do Amazon RDS monitora o espaço livre e amplia o armazenamento quando ele fica baixo por alguns minutos, respeitando um intervalo entre modificações e o maximum storage threshold, que funciona como teto de custo. O ajuste acontece sem downtime e evita o estado storage-full.

✗ A) Incorreta. Promover réplica troca o endpoint, interrompe a replicação e exige intervenção; não é um mecanismo de crescimento automático de armazenamento.

✗ B) Incorreta. Funciona, mas cria código e alarmes para manter, enquanto o RDS já oferece o crescimento automático de forma nativa.

✗ C) Incorreta. O tipo io2 define desempenho de I/O; ele não aumenta o tamanho alocado sem uma modificação.

> **Dica de prova:** FreeStorageSpace zerando no RDS: storage autoscaling com maximum storage threshold.

R2.2. B) Roteamento de failover no Route 53, com health check no registro primário e o registro secundário na outra região

Por quê: O roteamento de failover do Route 53 responde com o registro primário enquanto o health check associado estiver saudável e passa ao secundário quando ele falha, sem intervenção. Pesos sem health check não mudam o destino, um TTL longo atrasa a mudança e um ALB, por ser regional, não protege contra a perda da região.

✗ A) Incorreta. Um ALB é um recurso regional; mesmo que receba targets remotos, ele continua sendo o ponto de entrada na região primária e fica indisponível junto com ela.

✗ C) Incorreta. O roteamento por latência escolhe a região mais rápida, não a saudável, e o TTL longo atrasa qualquer troca nos resolvedores.

✗ D) Incorreta. Sem health check, o Route 53 não sabe que a primária falhou e o registro com peso 0 continua sem receber tráfego.

> **Dica de prova:** Ativo-passivo entre regiões por DNS: roteamento de failover do Route 53 com health check no registro primário.

R2.3. A) Colocar réplicas em outras AZs e ativar Multi-AZ com failover automático no grupo de replicação

Por quê: Com Multi-AZ, o ElastiCache distribui as réplicas em AZs diferentes e habilita o failover automático: se o primário ou sua AZ falha, a réplica com menor atraso de replicação é promovida e o DNS do endpoint primário passa a apontar para ela. Sem réplica em outra AZ, a promoção não resolve uma falha de AZ.

✗ B) Incorreta. Restaurar um snapshot cria um cluster novo, leva tempo e perde os dados gravados desde o último backup.

✗ C) Incorreta. Mais capacidade no nó não protege contra a perda da zona de disponibilidade.

✗ D) Incorreta. Failover automático sem Multi-AZ não garante réplicas em AZs diferentes; a promoção ocorre, mas não protege contra a falha da AZ inteira.

> **Dica de prova:** ElastiCache resistente a falha de AZ: réplicas em outras AZs + Multi-AZ com failover automático. Failover sem Multi-AZ não garante AZs diferentes.

R2.4. A) Criar uma regra de cópia no plano do AWS Backup para um cofre em outra região, com Vault Lock em modo compliance nos cofres

Por quê: Regras de cópia de um plano do AWS Backup copiam os pontos de recuperação para cofres em outras regiões ou contas. Com o Vault Lock em modo compliance, depois que o período de carência termina, a configuração do cofre não pode ser alterada nem excluída pelo cliente, pelo dono da conta ou pela AWS enquanto houver pontos de recuperação, e os backups ficam protegidos até o fim da retenção.

✗ B) Incorreta. Snapshots do Data Lifecycle Manager e backups do RDS podem ser apagados por administradores com permissão, então não há imutabilidade.

✗ C) Incorreta. Não há cópia em outra região, e um administrador pode alterar a política IAM e excluir os backups.

✗ D) Incorreta. Snapshots do EBS não podem ser exportados para um bucket do cliente como backup restaurável, e a exportação do RDS para o S3 gera arquivos de dados, não um snapshot.

> **Dica de prova:** Backups imutáveis (nem o root apaga) + cópia em outra região = AWS Backup com Vault Lock em modo compliance e regra de cópia entre regiões.

03

CAPÍTULO 3 · 22% DA PROVA

Domínio 3: Implantação, provisionamento e automação

Aqui eu preciso dominar o CloudFormation em detalhe (change sets, políticas de recurso, sinais, capabilities, estados de falha, StackSets), além de CDK e Terraform no fluxo de pipeline. A prova também cobra Systems Manager como plataforma de automação: Run Command, State Manager, Patch Manager, Distributor, Quick Setup, DHMC, hybrid activation, Change Calendar e OpsCenter. Implantação de aplicação aparece com ECS rolling e blue/green, CodeDeploy para EC2 e Lambda, ALB com pesos, instance refresh, EC2 Image Builder e ECR. Onde mais erro: trocar DeletionPolicy por UpdateReplacePolicy, achar que termination protection bloqueia updates e esquecer que regras agendadas do EventBridge são em UTC. Os serviços novos do guia (Kiro e AWS DevOps Agent) também aparecem.

Neste capítulo

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

Tópicos-chave

Change set, stack policy e termination protection

Change sets mostram o impacto de uma atualização, inclusive substituições, antes de executá-la. A stack policy define quais ações de update são permitidas em quais recursos e só vale para updates feitos pelo CloudFormation. A termination protection impede excluir a stack, mas não bloqueia atualizações.

DeletionPolicy x UpdateReplacePolicy

DeletionPolicy (Delete, Retain, Snapshot ou RetainExceptOnCreate) vale quando o recurso sai da stack ou a stack é excluída; para instâncias RDS o padrão é Snapshot, que guarda os dados mas remove o banco. UpdateReplacePolicy vale quando uma propriedade exige substituição durante um update. Para manter o banco vivo ao apagar a stack, atualize-a primeiro com DeletionPolicy: Retain.

CreationPolicy, cfn-signal e UpdatePolicy

CreationPolicy com ResourceSignal faz o CloudFormation esperar Count sinais dentro do Timeout; o cfn-signal no fim do user data envia o resultado, e sem sinal a stack faz rollback. DependsOn só ordena a criação. UpdatePolicy com AutoScalingRollingUpdate troca instâncias do ASG em lotes (MaxBatchSize, MinInstancesInService, WaitOnResourceSignals); AutoScalingReplacingUpdate cria um grupo novo.

Capabilities e service role

Recursos IAM sem nome exigem CAPABILITY_IAM; com nomes personalizados (RoleName, UserName), CAPABILITY_NAMED_IAM; macros e transforms como AWS::Serverless exigem CAPABILITY_AUTO_EXPAND. Sem elas o erro é InsufficientCapabilitiesException. Com service role, o CloudFormation usa só as permissões da role, e quem a associa precisa de iam:PassRole.

Rollback e estados de falha

Por padrão a falha reverte tudo; Preserve successfully provisioned resources serve para depurar. Rollback triggers revertem a operação se um alarme entrar em ALARM durante ela ou no tempo de monitoramento. UPDATE_ROLLBACK_FAILED se resolve com continue update rollback (corrigindo a causa ou com resources to skip), e DELETE_FAILED por bucket com objetos se resolve reexcluindo com RetainResources.

Recursos existentes, nomes e exportações

Resource import traz um recurso existente para a stack com change set do tipo IMPORT, DeletionPolicy e o identificador. Nome físico fixo que já existe gera erro already exists; deixe o CloudFormation gerar o nome ou derive de AWS::StackName. Uma exportação importada por outra stack não pode ser alterada nem a stack apagada: use list-imports, migre as consumidoras e depois exclua.

Extensões do template

Custom resources (Custom:: com ServiceToken de Lambda ou SNS) tratam Create, Update e Delete e respondem com PUT na ResponseURL pré-assinada do S3; Lambda em VPC precisa de NAT ou gateway endpoint do S3, senão a stack espera até o timeout. Modules registrados no registry privado entram como tipo terminado em ::MODULE. Referências dinâmicas resolve:secretsmanager e parâmetros do tipo AWS::SSM::Parameter::Value para a AMI pública evitam valores fixos.

StackSets

Implantam o mesmo template em várias contas e regiões, com parameter overrides por stack instance. No modelo self-managed, AWSCloudFormationStackSetAdministrationRole fica na conta administradora e AWSCloudFormationStackSetExecutionRole em cada conta-alvo; no service-managed, usam o Organizations com trusted access, implantação automática em OUs e até cinco administradores delegados. Nas opções de operação, Maximum concurrent accounts define o lote e Failure tolerance a parada; no modo Strict, a concorrência real fica em tolerância mais um.

Validação e pipeline de IaC

cfn-guard avalia templates contra regras de política escritas em DSL e barra o deploy no pipeline; cfn-lint valida o template; validate-template só confere sintaxe; drift detection compara recursos implantados com o template depois. O CodePipeline encadeia CodeBuild, staging, aprovação manual e produção, e tem as ações CloudFormationStackSet e CloudFormationStackInstances. Git sync sincroniza uma stack com um branch, sem estágios nem aprovação.

AWS CDK

O bootstrap cria a stack CDKToolkit (bucket de assets, repositório ECR, roles de deploy e parâmetro SSM de versão) por ambiente, ou seja, por conta e região. cdk diff faz o papel do change set e pode falhar o pipeline com --fail. Para ambiente efêmero por branch, inclua o nome do branch no ID da stack e rode cdk destroy ao fechar o pull request.

Terraform na AWS

O state remoto fica no backend s3 com versionamento, criptografia e use_lockfile (bloqueio nativo desde o Terraform 1.10; o lock com DynamoDB está sendo descontinuado), com backends separados por ambiente. Recursos criados fora entram com terraform import depois de declarados; state rm tira do controle sem destruir; o state nunca se edita à mão. Drift aparece no plan, e várias contas usam o bloco assume_role no provider.

Systems Manager: execução e estado

Run Command executa uma vez, com max-concurrency (padrão 50) e max-errors (padrão 0), e saída longa vai para o S3. State Manager mantém e reaplica um estado desejado por associações com agendamento e conformidade. Distributor publica pacotes de terceiros versionados instalados por AWS-ConfigureAWSPackage; Inventory com resource data sync para um bucket central permite consultar no Athena.

Systems Manager: patches e gerenciamento de nós

Patch baselines aprovam patches (com atraso de aprovação), patch groups ligam instâncias pela tag Patch Group ou PatchGroup e a maintenance window agenda o AWS-RunPatchBaseline. Managed node precisa de SSM Agent, AmazonSSMManagedInstanceCore e rede (NAT ou endpoints ssm, ssmmessages e ec2messages). A DHMC dispensa instance profile, mas é ativada por conta e região e exige IMDSv2; servidores fora da AWS entram por hybrid activation e aparecem com prefixo mi-.

Systems Manager: governança operacional

Quick Setup Host Management aplica a configuração básica à organização inteira com poucos cliques. Change Calendar define períodos abertos ou fechados, e o runbook o respeita com aws:assertAwsResourceProperty consultando GetCalendarState. OpsCenter centraliza problemas como OpsItems, criados por alarmes ou regras do EventBridge, com deduplicação e runbooks; Parameter Store usa hierarquia por caminho, GetParametersByPath e labels de versão.

Implantação de contêineres e Lambda

No rolling update do ECS, minimumHealthyPercent define o piso de tarefas saudáveis e maximumPercent o teto (100 e 200 mantêm a capacidade inteira na troca); o deployment circuit breaker com rollback marca a implantação como falha quando as tarefas novas não estabilizam e volta à última implantação COMPLETED. O blue/green do CodeDeploy para ECS cria um task set de substituição no mesmo serviço, usa dois target groups e listener de teste e desloca tráfego em canary, linear ou all-at-once. Para Lambda, o CodeDeploy ajusta o peso do alias (canary em dois passos, linear em passos iguais) e reverte por alarmes do deployment group.

Implantação em EC2

No CodeDeploy in-place, OneAtATime é o padrão e o hook ValidateService testa a nova versão depois de ApplicationStart. Instance refresh troca instâncias do ASG respeitando minimum healthy percentage, checkpoints e skip matching. Regras do ALB podem encaminhar para target groups com pesos de 0 a 999, permitindo canário e rollback sem DNS.

EC2 Image Builder, AMIs e ECR

Image recipe gera AMI e container recipe gera imagem no ECR; distribution settings copiam a AMI para regiões e contas, criam versão de launch template e podem gravar o ID em um parâmetro do Parameter Store. Deprecar uma AMI a esconde sem impedir lançamentos por ID, e o Recycle Bin recupera AMIs desregistradas se a regra existir antes. No ECR: tag immutability (IMMUTABLE_WITH_EXCLUSION para exceções como latest), lifecycle policy por tagStatus untagged, enhanced scanning com Inspector e replicação configurada na origem (até 25 destinos, só imagens novas).

Automação por eventos e novos serviços

EventBridge Scheduler aceita cron, rate e one-time com fuso IANA e horário de verão, e pode iniciar runbooks; regras agendadas do EventBridge são sempre em UTC. Eventos aws.health scheduledChange antecipam manutenções; S3 para EventBridge permite filtrar por tamanho e enviar a Step Functions. Kiro organiza trabalho em specs, steering e hooks (gatilhos File Save que rodam cfn-lint e cfn-guard), e o AWS DevOps Agent investiga incidentes em um Agent Space com integração a ferramentas como ServiceNow.

Macetes para memorizar

Políticas de recurso

Deletion = quando sai; Replace = quando é trocado; Stack policy = quando é atualizado; Termination = a stack inteira.

Capabilities

IAM sem nome = IAM; IAM com nome = NAMED_IAM; macro ou transform = AUTO_EXPAND.

Esperar ou ordenar

Signal espera, DependsOn só enfileira.

StackSets

Self-managed = duas roles na mão; Service-managed = Organizations faz por você.

Systems Manager

Run Command roda uma vez; State Manager mantém; Automation orquestra; Maintenance Window diz quando; Change Calendar diz quando não.

ECS em duas roles

Task role é da aplicação; execution role é do agente (imagem, logs e segredos).

Kiro

Spec planeja, steering orienta, hook executa sozinho no evento.

Tabela de memorização: não confunda

Quando é...	...e quando é
DeletionPolicy O recurso vai sair da stack ou a stack vai ser excluída e quero reter ou gerar snapshot.	**UpdateReplacePolicy** Uma atualização vai substituir o recurso (por exemplo, mudar a chave de partição) e quero manter o antigo.
Stack policy Proteger um recurso crítico contra updates do próprio CloudFormation.	**Termination protection** Impedir que a stack inteira seja excluída.
CreationPolicy com cfn-signal Esperar o bootstrap da instância terminar antes de marcar o recurso como criado.	**DependsOn** Só ordenar a criação de um recurso depois de outro chegar a CREATE_COMPLETE.
StackSets self-managed Contas sem AWS Organizations; criar AdministrationRole e ExecutionRole manualmente.	**StackSets service-managed** Organização com OUs, implantação automática em contas novas e administrador delegado.
State Manager Manter um estado desejado reaplicado periodicamente em instâncias novas e existentes.	**Run Command** Executar um comando pontual em muitas instâncias com controle de concorrência e erros.

Rolling update do ECS com circuit breaker Trocar tarefas em lotes no mesmo target group com rollback automático de falhas de implantação.	**Blue/green do CodeDeploy** Ambiente de substituição separado, listener de teste e deslocamento canary, linear ou de uma vez com rollback por alarme.
Canary no CodeDeploy Mandar uma porcentagem por um intervalo e depois o restante (por exemplo, LambdaCanary10Percent5Minutes).	**Linear no CodeDeploy** Avançar em incrementos iguais a cada intervalo até 100%.

Números que caem

Item	Valor
Run Command: max-concurrency e max-errors padrão	**50 e 0**
Administradores delegados de StackSets	**até cinco contas-membro**
Modo Strict de StackSets	**concorrência real limitada à tolerância a falhas mais um**
Lambda assíncrona	**retry de 0 a 2 tentativas; idade máxima do evento até 6 horas**
Visibility timeout para SQS com Lambda	**pelo menos seis vezes o timeout da função**
maxReceiveCount recomendado com Lambda	**pelo menos 5**
Retenção do DynamoDB Streams	**24 horas**
Timeout padrão de custom resource sem resposta	**3.600 segundos**
Pesos de target groups em regra do ALB	**0 a 999**
Replicação do ECR	**até 25 destinos**
ECS rolling update seguro	**minimumHealthyPercent 100 e maximumPercent 200**
DHMC	**SSM Agent 3.2.582.0 ou superior e IMDSv2**
Quick Setup Host Management	**atualiza o SSM Agent a cada duas semanas, inventário a cada 30 minutos e varredura diária de patches**
Bloqueio nativo de state no S3	**use_lockfile, desde o Terraform 1.10**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Usar validate-template para barrar portas abertas ou bucket público no template.
Como não cair: validate-template só confere sintaxe; regras de segurança se validam com cfn-guard no pipeline.

⚠ **Alerta de prova:** Usar drift detection para impedir um template inseguro.
Como não cair: Drift detection compara o que já foi implantado com o template; é pós-implantação.

⚠ **Alerta de prova:** Contar com a termination protection para bloquear updates ou preservar um recurso.
Como não cair: Ela só impede excluir a stack; updates se controlam com stack policy e retenção com DeletionPolicy ou UpdateReplacePolicy.

⚠ **Alerta de prova:** Usar DeletionPolicy: Retain para manter a tabela que será substituída no update.
Como não cair: Substituição em update é coberta pela UpdateReplacePolicy.

⚠ **Alerta de prova:** Configurar StackSets com 10 contas simultâneas e tolerância 2 no modo Strict.
Como não cair: No Strict a concorrência real fica em tolerância mais um (3 contas); use Soft failure tolerance.

⚠ **Alerta de prova:** Criar regra agendada do EventBridge para rodar às 7h de Nova York com horário de verão.
Como não cair: Regras agendadas são avaliadas em UTC; o EventBridge Scheduler aceita fuso IANA e ajusta o horário de verão.

⚠ **Alerta de prova:** Escrever cron com * no dia do mês e no dia da semana.
Como não cair: Um dos dois deve ser ?, como em cron(0 7 ? * MON-FRI *).

⚠ **Alerta de prova:** Informar CAPABILITY_AUTO_EXPAND para um template que cria role com nome fixo.
Como não cair: Role nomeada exige CAPABILITY_NAMED_IAM; AUTO_EXPAND é para macros e transforms.

⚠ **Alerta de prova:** Dar à task role do ECS a permissão para buscar segredos do campo secrets.
Como não cair: Quem busca segredos antes de iniciar os contêineres é a task execution role, com GetSecretValue e kms:Decrypt.

⚠ **Alerta de prova:** Usar DependsOn para esperar o user data terminar.
Como não cair: A instância chega a CREATE_COMPLETE ao iniciar; esperar o bootstrap exige CreationPolicy com cfn-signal.

⚠ **Alerta de prova:** Associar instâncias ao patch baseline com a tag Environment.
Como não cair: Patch groups usam a tag com chave Patch Group ou PatchGroup.

⚠ **Alerta de prova:** Ativar a DHMC em uma região e esperar que valha para as outras.
Como não cair: A DHMC é configurada por conta e por região; ative em cada região usada.

⚠ **Alerta de prova:** Configurar destino on-failure da Lambda assíncrona para uma fila FIFO ou mais de 2 tentativas.
Como não cair: Destinos assíncronos não aceitam FIFO; retry vai de 0 a 2 e idade máxima até 6 horas.

⚠ **Alerta de prova:** Usar data source ou editar o state para colocar um recurso existente sob o Terraform.
Como não cair: Data source só lê; declare o recurso e use terraform import, conferindo um plan sem recriação.

Como cai na prova

📝 **Como cai:** Uma stack falhou ou está presa em um estado e há quatro ações possíveis.
Como resolver: Leia o estado exato: UPDATE_ROLLBACK_FAILED pede continue update rollback; DELETE_FAILED com bucket pede RetainResources; erro already exists pede nome gerado; exportação em uso pede list-imports e migração. Descarte update direto e exclusão da stack.

📝 **Como cai:** Proteger dados ou recursos durante updates e exclusões.
Como resolver: Defina o momento do risco (update, substituição, exclusão do recurso ou da stack) e escolha a política correspondente; lembre que a política só vale depois de aplicada à stack por atualização.

📝 **Como cai:** Mesma infraestrutura em muitas contas e regiões com governança.
Como resolver: Verifique se há Organizations (service-managed com auto deployment) ou não (self-managed com duas roles); para promoção controlada, pipeline com CloudFormationStackSet, aprovação e CloudFormationStackInstances.

📝 **Como cai:** Implantação segura com rollback automático.
Como resolver: Identifique a plataforma: ECS rolling com circuit breaker ou blue/green do CodeDeploy; Lambda com CodeDeploy no alias e alarmes; EC2 com OneAtATime e ValidateService; ASG com instance refresh ou UpdatePolicy; stack com rollback triggers.

📝 **Como cai:** Automação operacional na frota com o menor esforço.
Como resolver: Prefira o recurso nativo do Systems Manager (Quick Setup, State Manager, Patch Manager, Distributor, Run Command com rate control) em vez de Lambda, cron em instância ou scripts por conta.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
trazer bucket ou tabela existente para uma stack sem recriar	**Resource import (change set IMPORT + DeletionPolicy + identificador)**
InsufficientCapabilitiesException	**CAPABILITY_IAM, CAPABILITY_NAMED_IAM ou CAPABILITY_AUTO_EXPAND conforme o template**
UPDATE_ROLLBACK_FAILED	**Continue update rollback, corrigindo a causa ou com resources to skip**
stack presa esperando custom resource com Lambda na VPC	**A função não alcança a ResponseURL do S3: NAT ou gateway endpoint do S3**
reverter a stack se um alarme da aplicação disparar após o deploy	**Rollback triggers com tempo de monitoramento**
mesma base para contas novas e existentes de uma OU	**StackSets service-managed com implantação automática**
autoatendimento de produtos aprovados sem dar permissões amplas	**Portfólio do Service Catalog com launch constraint compartilhado com a OU**
bloco reutilizável e versionado dentro dos templates dos times	**CloudFormation modules no registry privado**
parâmetro /cdk-bootstrap/.../version não encontrado	**Rodar cdk bootstrap naquela conta e região**
congelamento de mudanças para runbooks	**Change Calendar + aws:assertAwsResourceProperty exigindo OPEN**
instalar e versionar agente de terceiros em toda a frota	**Distributor + associação do State Manager**
managed node não aparece no Systems Manager	**SSM Agent, AmazonSSMManagedInstanceCore e rede (NAT ou endpoints ssm, ssmmessages, ec2messages)**
GitHub Actions implantando na AWS sem chaves	**Provedor OIDC token.actions.githubusercontent.com e role com condição em sub**
mensagem processada em dobro com SQS e Lambda	**Visibility timeout de pelo menos 6x o timeout da função + ReportBatchItemFailures**

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

- Termination protection só impede excluir a stack; não bloqueia updates nem preserva recursos.
- Regras agendadas do EventBridge são em UTC; horário local com horário de verão é EventBridge Scheduler.
- CDK bootstrap, DHMC, criptografia padrão e eventos do AWS Health são por região: configure em cada uma.
- Com service role, o CloudFormation ignora as permissões do usuário; AccessDenied em recurso se corrige na role.
- Pipelines e ferramentas externas usam roles com credenciais temporárias (OIDC, assume_role), nunca chaves de longo prazo.

Essenciais para a prova (checklist)

☐ Sei usar change sets, stack policy, termination protection, DeletionPolicy e UpdateReplacePolicy no momento certo.

☐ Sei configurar CreationPolicy com cfn-signal e UpdatePolicy de Auto Scaling group.

☐ Sei qual capability cada template exige e o papel da service role com iam:PassRole.

☐ Sei recuperar stacks em UPDATE_ROLLBACK_FAILED e DELETE_FAILED e lidar com exportações em uso.

☐ Sei diferenciar StackSets self-managed e service-managed e configurar concorrência e tolerância a falhas.

☐ Sei montar pipeline de IaC com cfn-lint, cfn-guard, staging, aprovação manual e produção.

☐ Sei gerenciar state, import, drift e múltiplas contas no Terraform e bootstrap e diff no CDK.

☐ Sei escolher entre Run Command, State Manager, Automation, Patch Manager, Distributor e Quick Setup.

☐ Sei configurar implantações ECS rolling, CodeDeploy blue/green, canary e linear e CodeDeploy in-place com hooks.

☐ Sei usar EC2 Image Builder, deprecação de AMI, Recycle Bin e os recursos de imutabilidade, lifecycle, scanning e replicação do ECR.

Questões de revisão do capítulo

R3.1 · Domínio 3: Implantação, provisionamento e automação

O time de plataforma mantém, em uma conta central, produtos aprovados de VPC e de banco RDS definidos em CloudFormation. Equipes de 40 contas de uma OU devem provisioná-los por autoatendimento, sem receber permissões amplas sobre EC2, VPC ou RDS, e contas novas da OU devem ter acesso aos produtos automaticamente. Qual solução atende?

A. Publicar os templates em um bucket S3 compartilhado e conceder aos desenvolvedores permissão para criar stacks em cada uma das contas
B. Compartilhar os templates com a OU pelo AWS RAM e permitir que as equipes criem stacks a partir dos recursos compartilhados
C. Compartilhar o portfólio do Service Catalog com a OU pelo AWS Organizations, com launch constraint e principal names compartilhados
D. Implantar os produtos com StackSets de permissões gerenciadas pelo serviço e implantação automática em contas novas da OU

R3.2 · Domínio 3: Implantação, provisionamento e automação

Uma equipe padroniza o provisionamento em várias contas com AWS CloudFormation StackSets. Os templates ficam em um repositório Git conectado ao AWS CodePipeline, e cada mudança deve ser barrada antes da implantação se violar regras de segurança da empresa, como security groups abertos para 0.0.0.0/0 ou buckets S3 sem bloqueio de acesso público. Qual abordagem atende ao requisito?

A. Ativar a detecção de drift nas instâncias de stack do StackSet após cada implantação e notificar a equipe pelo Amazon SNS
B. Aplicar uma stack policy e a proteção contra encerramento nas stacks criadas pelo StackSet em cada conta de destino
C. Executar o comando aws cloudformation validate-template em uma etapa do CodeBuild antes da etapa que atualiza o StackSet
D. Incluir no pipeline uma etapa do CodeBuild que executa o cfn-guard com as regras de segurança da empresa nos templates

R3.3 · Domínio 3: Implantação, provisionamento e automação

Uma empresa provisiona produção e homologação com AWS CloudFormation pelo pipeline de CI/CD. Em alguns deploys o time altera uma única função AWS Lambda da stack e quer, antes de aplicar, ver exatamente quais recursos serão modificados ou substituídos, além de impedir que o banco de dados da stack seja atualizado por engano. Qual abordagem atende a esses requisitos?

A. Gerar um change set para revisar as mudanças antes de executá-lo e definir uma stack policy que nega Update no recurso do banco
B. Criar um change set do tipo IMPORT com --resources-to-import, informando a função Lambda que deve ser atualizada na stack
C. Executar update-stack com --disable-rollback, preservando os recursos já atualizados se a atualização falhar no meio do processo
D. Habilitar a termination protection da stack, para que os recursos que não fazem parte da mudança fiquem bloqueados durante o update

R3.4 · Domínio 3: Implantação, provisionamento e automação

Uma rede varejista congela mudanças em produção durante a Black Friday. Vários runbooks personalizados do Systems Manager Automation são iniciados por agendamentos e por regras do EventBridge, e a equipe precisa que eles não alterem recursos nesse período, sem desligar cada agendamento e sem depender de aprovação humana. Qual solução atende?

A. Criar uma regra agendada do EventBridge que acione uma Lambda para desligar as regras de origem no início do congelamento e religar no fim
B. Adicionar aos runbooks uma etapa aws:approve com aprovadores da gerência, que liberam a execução fora do período de congelamento
C. Criar um evento de congelamento no Change Calendar e incluir nos runbooks uma etapa aws:assertAwsResourceProperty exigindo OPEN
D. Criar uma maintenance window cobrindo o período de congelamento e registrar os runbooks como tarefas dessa janela

Respostas comentadas

R3.1. C) Compartilhar o portfólio do Service Catalog com a OU pelo AWS Organizations, com launch constraint e principal names compartilhados

Por quê: O AWS Service Catalog organiza produtos aprovados em portfólios. Com a integração ao AWS Organizations, a conta de gerenciamento ou um administrador delegado compartilha o portfólio com a organização ou com OUs, alcançando contas novas. Launch constraints definem a role usada no lançamento, e o compartilhamento de principal names concede acesso aos usuários das contas destinatárias.

✗ A) Incorreta. Criar stacks direto exige dar aos desenvolvedores permissões amplas sobre os serviços envolvidos, o que o requisito proíbe.

✗ B) Incorreta. O AWS RAM compartilha recursos como sub-redes e Transit Gateways; templates do CloudFormation não são compartilhados por ele.

✗ D) Incorreta. StackSets empurram as mesmas stacks para as contas; não oferecem autoatendimento sob demanda escolhido pelas equipes.

> **Dica de prova:** Autoatendimento de produtos aprovados em várias contas sem dar permissões amplas = portfólio do Service Catalog + launch constraint.

R3.2. D) Incluir no pipeline uma etapa do CodeBuild que executa o cfn-guard com as regras de segurança da empresa nos templates

Por quê: O AWS CloudFormation Guard (cfn-guard) avalia templates JSON ou YAML contra regras de política escritas em DSL, o que permite validação preventiva (shift-left) de requisitos de segurança e conformidade. Rodando em uma etapa do CodeBuild, a validação com falha interrompe o pipeline antes que o StackSet implante a mudança nas contas.

✗ A) Incorreta. Drift detection compara os recursos implantados com o template depois da implantação; não barra um template inseguro nem avalia regras de segurança.

✗ B) Incorreta. Stack policy protege recursos contra atualizações e a proteção contra encerramento impede excluir a stack; nenhuma das duas avalia o conteúdo do template.

✗ C) Incorreta. validate-template verifica a sintaxe do template (JSON ou YAML válido), não avalia regras de segurança como portas abertas ou bloqueio de acesso público.

> **Dica de prova:** Barrar template inseguro antes do deploy = cfn-guard no pipeline. validate-template só checa sintaxe; drift é pós-implantação.

R3.3. A) Gerar um change set para revisar as mudanças antes de executá-lo e definir uma stack policy que nega Update no recurso do banco

Por quê: Change sets permitem pré-visualizar o impacto de uma atualização, inclusive substituições, e o CloudFormation só aplica as mudanças quando o change set é executado. A stack policy é um documento JSON que define quais ações de update podem ser feitas em quais recursos; depois de definida, protege todos os recursos por padrão, e é preciso permitir de forma explícita os que podem mudar.

✗ B) Incorreta. Change sets de importação trazem recursos existentes para a gestão da stack; não servem para atualizar de forma seletiva recursos já gerenciados.

✗ C) Incorreta. Desabilitar o rollback não mostra as mudanças com antecedência nem protege recursos; só mantém o estado parcial em caso de falha.

✗ D) Incorreta. A termination protection impede a exclusão da stack; ela não bloqueia atualizações de recursos durante um update.

> **Dica de prova:** Ver antes de aplicar = change set. Proteger recurso crítico durante update = stack policy. Impedir exclusão da stack = termination protection.

R3.4. C) Criar um evento de congelamento no Change Calendar e incluir nos runbooks uma etapa aws:assertAwsResourceProperty exigindo OPEN

Por quê: O Change Calendar define períodos em que ações são permitidas ou bloqueadas: calendários DEFAULT_OPEN ficam CLOSED durante eventos, e DEFAULT_CLOSED ficam OPEN durante eventos. Runbooks do Automation passam a respeitar o calendário com uma etapa aws:assertAwsResourceProperty que consulta GetCalendarState; maintenance windows e associações do State Manager também podem ser vinculadas ao calendário.

✗ A) Incorreta. Desligar e religar regras com Lambda é exatamente a manutenção de agendamentos que a equipe quer evitar e não cobre execuções iniciadas por outros caminhos.

✗ B) Incorreta. A aprovação manual contraria o requisito de não depender de pessoas e ainda deixaria execuções pendentes acumuladas durante o congelamento.

✗ D) Incorreta. Uma maintenance window define quando tarefas devem rodar; registrar os runbooks nela faria com que rodassem justamente no período de congelamento.

> **Dica de prova:** Congelamento de mudanças no Systems Manager: Change Calendar + verificação de GetCalendarState no runbook.

04

CAPÍTULO 4 · 16% DA PROVA

Domínio 4: Segurança e conformidade

Neste domínio eu preciso separar controle preventivo (SCP, permissions boundary, Block Public Access, bucket policy com Deny) de controle detectivo (AWS Config, Security Hub, GuardDuty, Macie, Inspector, Trusted Advisor) e saber a ferramenta de cada detecção. A prova cobra muito KMS (chave gerenciada pela AWS x pelo cliente, key policy, rotação, multi-Region, acesso entre contas), Secrets Manager x Parameter Store, ACM e TLS. Também cai IAM avançado: Identity Center, ExternalId, ABAC, MFA na CLI, Access Analyzer e policy simulator. Onde mais erro: achar que SCP concede permissão ou afeta a conta de gerenciamento, tentar editar a key policy de aws/s3 ou aws/rds e confundir os analyzers do Access Analyzer. Os serviços novos (AWS Security Agent, conformance packs organizacionais, Region deny do Control Tower) também aparecem.

Neste capítulo

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

Tópicos-chave

Service control policies (SCP)

SCPs definem o máximo de permissões das contas-membro e não concedem nada: a permissão efetiva é a interseção com as políticas IAM. Valem até para administradores locais, são herdadas da raiz pelas OUs e não afetam usuários e roles da conta de gerenciamento. Exceções se fazem com condição em aws:PrincipalArn no Deny.

SCP de residência por região

Chamadas a serviços globais chegam com aws:RequestedRegion igual a us-east-1. A SCP usa Deny com NotAction listando iam, cloudfront, route53, support, organizations e outros globais usados, e StringNotEquals em aws:RequestedRegion. No Control Tower, o Region deny aplicado à OU restringe só aquela OU; o do landing zone vale para tudo e não pode negar a região de origem.

Permissions boundary

Define o máximo que uma identidade pode ter; as permissões efetivas são a interseção entre políticas e boundary. Para delegar a criação de roles sem escalada, permita iam:CreateRole só com a condição iam:PermissionsBoundary igual ao ARN do boundary e negue a remoção ou troca dele. Aplicado por principal, não substitui uma SCP para toda a organização.

IAM Identity Center e federação

Permission sets são modelos de permissões; atribuir um grupo do IdP (sincronizado por SCIM) a uma conta com um permission set cria a role nela. Instâncias de conta do Identity Center não suportam permission sets. Para uma conta isolada com IdP SAML 2.0, crie um SAML identity provider no IAM e roles com sts:AssumeRoleWithSAML.

Credenciais temporárias e condições

Workloads usam roles: instance profile no EC2, OIDC para pipelines externos e assume_role entre contas. sts:ExternalId na trust policy, com valor gerado pelo terceiro, evita o confused deputy. MFA na CLI exige aws:MultiFactorAuthPresent e credenciais obtidas com sts get-session-token informando o código; ABAC compara aws:ResourceTag com aws:PrincipalTag.

IAM Access Analyzer e diagnóstico

O analyzer de acesso externo encontra recursos compartilhados fora da zona de confiança (conta ou organização); o de acesso não utilizado aponta roles, chaves e permissões sem uso com tracking period de 1 a 365 dias. A geração de política usa até 90 dias de CloudTrail para chegar ao menor privilégio. O policy simulator explica por que uma ação é negada, e sts decode-authorization-message decodifica mensagens de falha (exige sts:DecodeAuthorizationMessage).

Chaves KMS gerenciadas pela AWS x pelo cliente

Chaves gerenciadas pela AWS (aws/s3, aws/rds, aws/ebs, aws/ssm, aws/secretsmanager, aws/sns) não permitem editar a key policy, por isso não servem para compartilhar com outra conta nem para separar quem administra de quem lê. Chave gerenciada pelo cliente dá key policy, grants, rotação configurável e auditoria no CloudTrail. Snapshots, AMIs e segredos criptografados entre contas exigem recriptografar com uma chave do cliente liberada à outra conta.

Rotação e alcance das chaves KMS

Chaves simétricas do cliente com material gerado pelo KMS aceitam rotação automática com período configurável (padrão de 365 dias) e rotação sob demanda, mantendo key ID e ARN e sem recriptografar dados. Rotação manual troca key ID e ARN. Chaves são regionais; chaves multi-Region compartilham ID e key material entre regiões para descriptografar no DR sem chamar a região primária.

Bucket policy e acesso ao S3

Deny explícito vence qualquer Allow, inclusive de administradores com s3:*. Forçar HTTPS é Deny com aws:SecureTransport false; restringir à rede da VPC é aws:SourceVpce (aws:SourceIp não vale para IP privado vindo de endpoint); liberar a organização é aws:PrincipalOrgID. Acesso entre contas direto precisa de Allow na bucket policy e na política IAM do principal; buckets não se compartilham pelo AWS RAM.

Block Public Access, Object Lock e SSE-KMS

Block Public Access no nível da conta vale para buckets atuais e futuros e prevalece sobre políticas e ACLs. Object Lock em modo Compliance impede sobrescrever ou apagar até pelo root; em Governance, quem tem s3:BypassGovernanceRetention contorna. Com SSE-KMS, cada objeto chama o KMS; o S3 Bucket Key reduz chamadas e custo para objetos novos.

Secrets Manager e Parameter Store

Rotação do Secrets Manager usa Lambda: usuário único tem pequeno risco de negação, usuários alternados evita isso com um segredo de superusuário. Replica secrets levam o segredo a outras regiões com a rotação só no primário; o RDS pode gerenciar a senha mestre no Secrets Manager com rotação padrão de sete dias. Parameter Store SecureString usa aws/ssm ou chave do cliente, mas não tem rotação nativa nem replicação entre regiões.

ACM e TLS

Validação por DNS renova sozinha enquanto o CNAME estiver na zona; por e-mail, cada renovação pede aprovação, e o método não muda no mesmo certificado. Certificados importados não renovam: monitore DaysToExpiry ou o evento Approaching Expiration. CloudFront exige certificado do ACM em us-east-1 para o viewer; no ALB, a versão mínima de TLS vem da security policy do listener (ELBSecurityPolicy-2016-08, padrão por CLI e IaC, aceita TLS 1.0).

Criptografia de dados em uso nos serviços

Para forçar TLS no RDS, use require_secure_transport no MySQL e MariaDB ou rds.force_ssl no PostgreSQL e SQL Server, em parameter group personalizado. A criptografia padrão do EBS é por conta e região; para bloquear volume sem criptografia em várias contas, SCP negando ec2:CreateVolume com ec2:Encrypted false. Cópia de snapshot para outra região usa chave da região de destino.

CloudTrail para auditoria

Organization trail, criado pela conta de gerenciamento ou administrador delegado, entrega eventos de todas as contas em um bucket central e não pode ser alterado pelas contas-membro; complete com SCP negando StopLogging, DeleteTrail e UpdateTrail. A validação de integridade gera arquivos de digest com SHA-256. O Event history guarda 90 dias; para eventos mais antigos, consulte os logs da trilha no S3 com o Athena.

AWS Config e conformidade

Regras gerenciadas como s3-bucket-public-read-prohibited, s3-account-level-public-access-blocks-periodic, access-keys-rotated, encrypted-volumes e required-tags detectam desvios, e a remediação usa documentos do SSM Automation. Eventos Config Rules Compliance Change no EventBridge permitem notificar NON_COMPLIANT. Conformance packs organizacionais pelo administrador delegado aplicam as mesmas regras sem que as contas alterem, e o agregador consolida a visão.

Serviços de detecção

GuardDuty detecta ameaças em CloudTrail, VPC Flow Logs e DNS; Malware Protection for S3 marca objetos com a tag GuardDutyMalwareScanStatus. Inspector acha vulnerabilidades em EC2, imagens de contêiner e Lambda (com fixAvailable). Macie acha dados sensíveis no S3, com custom data identifiers (regex, keywords, ignore words) e descoberta automatizada por amostragem; Security Hub CSPM centraliza achados e suas automation rules, criadas só na conta administradora, só alteram campos dos achados.

Trusted Advisor e AWS Security Agent

Trusted Advisor aponta Exposed Access Keys (exclua a chave e investigue), S3 Bucket Permissions, Specific Ports Unrestricted (vermelho para 20, 21, 22, 1433, 3306, 3389 e 5432) e root sem MFA. A organizational view é habilitada na conta de gerenciamento com plano de suporte elegível, e seus eventos chegam ao EventBridge em us-east-1. O AWS Security Agent faz revisões de design e código, modelagem de ameaças e pentest sob demanda em um Agent Space, após verificar o domínio por registro TXT.

Resposta automática a incidentes

Achados de GuardDuty, Inspector, Macie e Security Hub e chamadas de API via CloudTrail chegam ao EventBridge, cujas regras iniciam Lambda ou runbooks do SSM Automation. Para instância comprometida, troque os security groups por um grupo de isolamento sem regras, preservando evidências em vez de encerrar. Achado CRITICAL do Inspector com correção disponível pode acionar AWS-RunPatchBaseline.

Macetes para memorizar

SCP

SCP é teto, não presente: limita, não concede, e não alcança a conta de gerenciamento.

Preventivo x detectivo

Prevenir = SCP, boundary, Block Public Access e Deny. Detectar = Config, Security Hub, GuardDuty, Macie, Inspector, Trusted Advisor.

Chave da AWS

Chave aws/ não sai de casa: não compartilha, não edita key policy, não escolhe período de rotação.

Detecção por serviço

GuardDuty = ameaça; Inspector = vulnerabilidade; Macie = dado sensível; Config = configuração; Access Analyzer = acesso.

Object Lock

Compliance não cede a ninguém, nem ao root; Governance cede a quem tem s3:BypassGovernanceRetention.

Regiões fixas

CloudFront pega certificado na Virgínia; Trusted Advisor e IAM avisam na Virgínia.

Rotação de segredos

Usuário único pisca; usuários alternados não piscam.

Tabela de memorização: não confunda

Quando é...	...e quando é
SCP Guardrail central em contas ou OUs que nem o administrador local derruba.	**Permissions boundary** Limitar o máximo de permissões de identidades específicas, como roles criadas por desenvolvedores.
Object Lock Compliance Imutabilidade que nenhum usuário, nem o root, consegue encurtar ou remover.	**Object Lock Governance** Proteção que usuários com s3:BypassGovernanceRetention podem contornar.
Chave gerenciada pela AWS Criptografia padrão sem necessidade de controlar a key policy nem compartilhar.	**Chave gerenciada pelo cliente** Controlar quem usa a chave, compartilhar com outra conta, configurar rotação ou negar administradores.
GuardDuty Comportamento malicioso em andamento: credencial anômala, mineração, comunicação com servidor malicioso, malware em uploads do S3.	**Inspector** Vulnerabilidades de software e exposição de rede em EC2, imagens de contêiner e Lambda.
Access Analyzer de acesso externo Encontrar recursos compartilhados com contas fora da organização.	**Access Analyzer de acesso não utilizado** Encontrar roles, chaves e permissões sem uso dentro da organização.
Validação por DNS no ACM Renovação gerenciada sem ação humana, com CNAME mantido na zona.	**Validação por e-mail no ACM** Quando não há como criar o registro DNS, aceitando aprovar cada renovação.
Secrets Manager Segredo com rotação automática, réplica em outra região ou acesso entre contas por resource policy.	**Parameter Store SecureString** Configuração ou segredo sem rotação nativa, criptografado com aws/ssm ou chave do cliente.

Números que caem

Item	Valor
Período padrão de rotação automática de chave KMS do cliente	**365 dias (configurável)**
Rotação padrão da senha mestre gerenciada pelo RDS no Secrets Manager	**a cada sete dias**
Event history do CloudTrail	**últimos 90 dias de eventos de gerenciamento**

Geração de política do IAM Access Analyzer	**atividade de até 90 dias**
Tracking period do analyzer de acesso não utilizado	**1 a 365 dias**
Portas sensíveis que deixam o Trusted Advisor vermelho	**20, 21, 22, 1433, 3306, 3389 e 5432**
Validação de integridade do CloudTrail	**digest com hash SHA-256 e assinatura SHA-256 com RSA**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Anexar uma SCP com Allow e esperar que o usuário passe a ter acesso.
Como não cair: SCP é teto, não concessão; sem política IAM concedendo, o AccessDenied continua.

⚠ **Alerta de prova:** Anexar a SCP diretamente à conta de gerenciamento para restringi-la.
Como não cair: SCPs nunca afetam usuários e roles da conta de gerenciamento.

⚠ **Alerta de prova:** Editar a key policy de aws/rds, aws/ebs ou aws/secretsmanager para liberar outra conta.
Como não cair: Key policy de chave gerenciada pela AWS não é editável; copie ou recriptografe com chave gerenciada pelo cliente.

⚠ **Alerta de prova:** Usar regra do AWS Config ou Security Hub para impedir uma ação.
Como não cair: São controles detectivos: avisam depois. Impedir é SCP, Block Public Access ou Deny em política.

⚠ **Alerta de prova:** Usar aws:SourceIp para restringir acesso que vem por VPC endpoint.
Como não cair: Requisições pelo endpoint chegam com IP privado; use aws:SourceVpce ou aws:SourceVpc.

⚠ **Alerta de prova:** Exigir HTTPS no bucket com um Allow condicionado a aws:SecureTransport true.
Como não cair: Allow não bloqueia quem já tem permissão por IAM; o padrão é Deny com aws:SecureTransport false.

⚠ **Alerta de prova:** Usar Object Lock em modo Governance quando nem o root pode apagar.
Como não cair: Governance pode ser contornado; imutabilidade total é modo Compliance.

⚠ **Alerta de prova:** Rotacionar a chave manualmente para trocar o key material sem quebrar as aplicações.
Como não cair: Rotação manual muda key ID e ARN; a automática mantém os dois e guarda o key material antigo para decifrar.

⚠ **Alerta de prova:** Habilitar a criptografia padrão do EBS na conta de gerenciamento para valer em todas as contas.
Como não cair: É configuração por conta e por região; para todas as contas, SCP com ec2:Encrypted.

⚠ **Alerta de prova:** Usar Allow list do Macie para detectar um padrão interno de matrícula.
Como não cair: Allow list suprime achados; detectar formato próprio é custom data identifier com regex, keywords e ignore words.

⚠ **Alerta de prova:** Usar automation rules do Security Hub para corrigir o recurso.
Como não cair: Elas só mudam campos dos achados (severidade, workflow, notas); correção é EventBridge com SSM Automation ou regra do Config com remediação.

⚠ **Alerta de prova:** Usar chaves de acesso de longo prazo do usuário na CLI com política exigindo MFA.
Como não cair: Chaves de longo prazo não carregam aws:MultiFactorAuthPresent; obtenha credenciais temporárias com sts get-session-token e o código MFA.

⚠ **Alerta de prova:** Procurar no Event history quem apagou um recurso há sete meses.
Como não cair: O Event history guarda 90 dias; consulte os logs da trilha no S3 com o Athena.

⚠ **Alerta de prova:** Colocar certificado do ACM da região do ALB na distribuição do CloudFront.
Como não cair: O certificado do viewer no CloudFront precisa estar em us-east-1; o do ALB protege só a conexão até a origem.

Como cai na prova

📝 **Como cai:** Requisito de governança multi-conta (bloquear, padronizar ou centralizar).
Como resolver: Pergunte se é para impedir (SCP, Region deny, tag policies com enforced_for), detectar (Config com conformance pack organizacional e agregador) ou centralizar logs (organization trail com SCP protegendo). Descarte soluções conta a conta e Lambdas próprias.

📝 **Como cai:** Acesso entre contas a dado criptografado (snapshot, AMI, segredo, objeto).
Como resolver: Exija duas peças: permissão no recurso (resource policy, compartilhamento do snapshot ou AMI, bucket policy) e key policy de chave gerenciada pelo cliente liberando a outra conta. Descarte qualquer opção com chave aws/.

📝 **Como cai:** Um achado de segurança precisa virar ação automática.
Como resolver: Identifique o detector certo, leve o achado ao EventBridge e acione SSM Automation ou Lambda; para isolar instância, troque security groups em vez de encerrar.

📝 **Como cai:** Restringir um bucket a uma role, a uma rede ou exigir TLS.
Como resolver: Use Deny explícito na bucket policy com a condição certa (aws:PrincipalArn, aws:SourceVpce, aws:SecureTransport); Allow condicional e política de endpoint sozinhos não bloqueiam quem já tem s3:*.

📝 **Como cai:** Credencial vazada ou acesso humano inseguro.
Como resolver: Corrija a causa raiz: exclua a chave exposta, troque chaves de longo prazo por roles (instance profile, OIDC), feche portas abertas e use Session Manager, habilite MFA no root.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
proibir algo em todas as contas sem que admins locais contornem	**SCP na raiz ou na OU, com exceção por aws:PrincipalArn**
desenvolvedores criando roles sem escalar privilégio	**Permissions boundary obrigatório via iam:PermissionsBoundary**
terceiro que assume roles em várias contas de clientes	**sts:ExternalId na trust policy**
acesso por classificação sem editar política a cada recurso novo	**ABAC com aws:ResourceTag igual a aws:PrincipalTag**
menor privilégio a partir do uso real	**Geração de política do IAM Access Analyzer com base no CloudTrail**
por que este usuário não consegue fazer esta ação	**IAM policy simulator**
Encoded authorization failure message	**aws sts decode-authorization-message**
descriptografar na região de DR o que foi criptografado na primária	**Chave multi-Region do KMS**
SSE-KMS caro ou com ThrottlingException do KMS	**S3 Bucket Key**

Lambda lendo segredo a cada invocação	**AWS Parameters and Secrets Lambda Extension com cache local**
dados pessoais dentro de objetos do S3	**Amazon Macie**
malware em arquivos enviados ao bucket	**GuardDuty Malware Protection for S3 com tag e bucket policy**
pentest sob demanda com achados validados e PR de correção	**AWS Security Agent**
padronizar chave, grafia e valores de tags na organização	**Tag policies com enforced_for**

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

- SCP não concede permissão e não afeta a conta de gerenciamento.
- Chave gerenciada pela AWS não tem key policy editável: nada de compartilhar com outra conta.
- Deny explícito vence tudo; é a forma de bloquear até administradores com permissões amplas.
- Certificado do CloudFront fica em us-east-1, e eventos de serviços globais chegam em us-east-1.

Essenciais para a prova (checklist)

☐ Sei escrever uma SCP de residência por região com NotAction para serviços globais e exceção por aws:PrincipalArn.

☐ Sei delegar a criação de roles com permissions boundary obrigatório.

☐ Sei configurar acesso multi-conta com permission sets do IAM Identity Center e federação SAML direta.

☐ Sei usar ExternalId, MFA na CLI, ABAC e credenciais temporárias por role e OIDC.

☐ Sei escolher entre os analyzers do IAM Access Analyzer, o policy simulator e o decode-authorization-message.

☐ Sei quando é obrigatório usar chave KMS gerenciada pelo cliente e como funcionam rotação e chaves multi-Region.

☐ Sei montar bucket policies com Deny para TLS, VPC endpoint, role e organização, e usar Block Public Access e Object Lock.

☐ Sei configurar rotação e réplicas no Secrets Manager e quando o Parameter Store basta.

☐ Sei proteger e centralizar o CloudTrail e investigar eventos antigos com o Athena.

☐ Sei qual serviço detecta ameaça, vulnerabilidade, dado sensível, configuração e malware, e como transformar achados em correção automática.

Questões de revisão do capítulo

R4.1 · Domínio 4: Segurança e conformidade

Uma organização tem 30 contas e usa o IAM Identity Center com usuários e grupos sincronizados do IdP corporativo por SCIM. Os DBAs precisam de acesso administrativo a bancos nas contas de produção e acesso somente leitura nas demais, e quem entrar no grupo de DBAs no IdP deve receber o acesso sem ação extra na AWS. Qual solução atende ao requisito com o menor esforço?

A. Criar os permission sets DBAdmin e ReadOnly e atribuir o grupo de DBAs às contas de produção com DBAdmin e às demais contas com ReadOnly
B. Criar os permission sets DBAdmin e ReadOnly e atribuí-los a cada usuário DBA nas contas, refazendo a atribuição quando alguém entrar na equipe
C. Anexar as políticas de administração de bancos diretamente ao grupo de DBAs no IAM Identity Center para que valham em cada conta da organização
D. Implantar com StackSets roles IAM em cada conta com trust policy para o IdP via SAML e mapear o atributo de grupo do IdP para cada role

R4.2 · Domínio 4: Segurança e conformidade

Uma empresa financeira precisa guardar por 7 anos os registros de todas as chamadas de API de suas contas, em todas as regiões. Os registros não podem ser excluídos nem sobrescritos por ninguém, inclusive o usuário root, e auditores precisam comprovar que os arquivos não foram alterados após a entrega. Como o engenheiro deve configurar a solução?

A. Criar uma trilha multirregião do CloudTrail entregando em um bucket com Object Lock em modo Governance de 7 anos e versionamento habilitado
B. Criar uma trilha multirregião do CloudTrail com validação de integridade dos logs, entregando em um bucket com Object Lock em modo Compliance de 7 anos
C. Criar uma trilha multirregião do CloudTrail entregando no CloudWatch Logs, com retenção de 7 anos no log group e proteção de dados habilitada
D. Criar uma trilha multirregião do CloudTrail entregando em um bucket com versionamento, MFA Delete e replicação entre regiões para outro bucket

R4.3 · Domínio 4: Segurança e conformidade

Uma norma interna passou a exigir que o material criptográfico das chaves KMS usadas por buckets S3 e volumes EBS seja trocado a cada 180 dias. Várias aplicações referenciam a chave pelo ARN, e a equipe não quer alterar código nem recriptografar os dados existentes. A chave é simétrica, gerenciada pelo cliente e com material gerado pelo KMS. O que deve ser feito?

A. Habilitar a rotação automática da chave com período de 180 dias, mantendo o mesmo key ID e o mesmo ARN
B. Habilitar a rotação automática da chave, que também recriptografa as data keys e os objetos existentes com o novo material
C. Migrar os recursos para a chave gerenciada pela AWS aws/s3, cujo período de rotação pode ser configurado em 180 dias
D. Criar uma nova chave KMS a cada 180 dias e apontar o alias usado pelas aplicações para ela, desativando a chave anterior

R4.4 · Domínio 4: Segurança e conformidade

Um bucket Amazon S3 guarda logs sensíveis de auditoria. Somente uma aplicação em instâncias EC2 de uma VPC específica, que usa a role AppAuditRole, pode ler os objetos, e as requisições devem passar pelo gateway endpoint do S3 dessa VPC. Usuários IAM da conta com permissões amplas, como s3:*, também não podem ler os logs. Qual solução atende com privilégio mínimo?

A. Adicionar à bucket policy instruções Deny de s3:GetObject para principais diferentes da AppAuditRole (aws:PrincipalArn) e para acessos fora do endpoint (aws:SourceVpce)
B. Adicionar à bucket policy um Allow de s3:GetObject para a AppAuditRole com a condição aws:SourceIp igual ao bloco CIDR privado da VPC da aplicação
C. Anexar ao instance profile uma role com permissão s3:GetObject no bucket, mantendo a bucket policy atual sem condições adicionais de rede ou de identidade
D. Restringir a política do gateway endpoint para permitir s3:GetObject no bucket à AppAuditRole e manter as permissões atuais dos usuários IAM da conta

Respostas comentadas

R4.1. A) Criar os permission sets DBAdmin e ReadOnly e atribuir o grupo de DBAs às contas de produção com DBAdmin e às demais contas com ReadOnly

Por quê: No IAM Identity Center, um permission set é um modelo de permissões (políticas gerenciadas, inline ou permissions boundary). Ao atribuir um usuário ou grupo a uma conta com um permission set, o serviço cria a role correspondente na conta. Atribuir grupos sincronizados por SCIM faz as mudanças de equipe no IdP refletirem no acesso sem trabalho extra.

✗ B) Incorreta. Atribuir por usuário exige ação na AWS a cada mudança na equipe, contrariando o requisito de acesso pelo grupo do IdP.

✗ C) Incorreta. No Identity Center, o acesso a contas é dado por atribuições de permission sets por conta; não existe política anexada ao grupo que valha para as contas.

✗ D) Incorreta. Funciona, mas exige manter roles, trust policies e mapeamento de atributos em cada conta, que o Identity Center já faz de forma centralizada.

> **Dica de prova:** Acesso multi-conta por função: permission sets atribuídos a grupos, conta por conta, no IAM Identity Center.

R4.2. B) Criar uma trilha multirregião do CloudTrail com validação de integridade dos logs, entregando em um bucket com Object Lock em modo Compliance de 7 anos

Por quê: O S3 Object Lock em modo Compliance aplica WORM: a versão do objeto não pode ser sobrescrita ou excluída por nenhum usuário, nem o root, e a retenção não pode ser encurtada. A validação de integridade do CloudTrail entrega arquivos de digest com hash SHA-256 e assinatura SHA-256 com RSA, que permitem verificar se um log foi alterado ou excluído. Uma trilha multirregião cobre as chamadas de todas as regiões.

✗ A) Incorreta. No modo Governance, usuários com a permissão s3:BypassGovernanceRetention podem remover a proteção, e falta a validação de integridade pedida pelos auditores.

✗ C) Incorreta. A retenção do log group pode ser alterada por quem tem permissão, e o CloudWatch Logs não oferece WORM nem arquivos de digest para comprovar a integridade.

✗ D) Incorreta. Versionamento, MFA Delete e replicação aumentam a proteção e a durabilidade, mas não garantem imutabilidade nem fornecem prova criptográfica de integridade.

> **Dica de prova:** Imutável até para o root = Object Lock em modo Compliance (Governance pode ser contornado). Provar que o log não mudou = validação de integridade do CloudTrail (arquivos de digest).

R4.3. A) Habilitar a rotação automática da chave com período de 180 dias, mantendo o mesmo key ID e o mesmo ARN

Por quê: Chaves KMS simétricas gerenciadas pelo cliente, com material gerado pelo KMS, aceitam rotação automática com período configurável (o padrão é 365 dias) e rotação sob demanda. A rotação muda só o material atual; a chave continua sendo o mesmo recurso, e o KMS guarda o material antigo para decifrar dados existentes. Chaves assimétricas, HMAC e de custom key stores exigem rotação manual.

✗ B) Incorreta. A rotação não recriptografa dados nem data keys; o KMS só passa a usar o material novo nas novas cifragens e escolhe sozinho o material certo para decifrar.

✗ C) Incorreta. Chaves gerenciadas pela AWS são rotacionadas pela AWS a cada ano, sem opção de configurar o período.

✗ D) Incorreta. A rotação manual muda o key ID e o ARN, quebrando as aplicações que referenciam o ARN, e desativar a chave anterior impede decifrar dados antigos.

> **Dica de prova:** Trocar material sem mudar key ID nem recriptografar: rotação automática da chave gerenciada pelo cliente (período configurável).

R4.4. A) Adicionar à bucket policy instruções Deny de s3:GetObject para principais diferentes da AppAuditRole (aws:PrincipalArn) e para acessos fora do endpoint (aws:SourceVpce)

Por quê: Para restringir um bucket a uma identidade e a um caminho de rede, a bucket policy usa Deny explícito, que vence qualquer Allow. Uma instrução nega s3:GetObject quando aws:PrincipalArn é diferente da role da aplicação e outra nega quando aws:SourceVpce é diferente do ID do endpoint. A política do endpoint pode complementar, limitando o que se acessa por ele.

✗ B) Incorreta. Um Allow não impede o acesso de quem já tem s3:* por política IAM, e aws:SourceIp não funciona com o IP privado de requisições que passam por um VPC endpoint.

✗ C) Incorreta. A permissão na role libera a aplicação, mas não bloqueia os usuários IAM com s3:* nem exige que o acesso passe pelo endpoint da VPC.

✗ D) Incorreta. A política do endpoint só controla o tráfego que passa por ele; usuários com s3:* continuam lendo os objetos pela internet ou por outra rede.

> **Dica de prova:** Bloquear até administradores da conta = Deny explícito na bucket policy. Origem pela VPC = aws:SourceVpce (ou aws:SourceVpc); aws:SourceIp não vale para IPs privados vindos de endpoint.

05

CAPÍTULO 5 · 18% DA PROVA

Domínio 5: Redes e entrega de conteúdo

Aqui eu preciso montar e diagnosticar redes: rotas, NAT gateway, internet gateway, security groups x NACLs, endpoints, peering, Transit Gateway, VPN e Direct Connect. A prova também cobra DNS a fundo (Route 53 Resolver inbound e outbound, private hosted zones, políticas de roteamento, alias) e entrega de conteúdo com CloudFront, WAF, Shield e Global Accelerator. Muitas questões trazem um sintoma (conexão expira, sem internet, 502 no CloudFront, cache hit baixo) e pedem a causa exata. As ferramentas de diagnóstico e o monitoramento de rede do CloudWatch (Network Flow Monitor, Network Synthetic Monitor, Internet Monitor) são cobrados pelo nome. Onde mais erro: inverter inbound e outbound endpoint, esquecer que NACL é stateless e confundir cache policy com origin request policy.

Neste capítulo

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

Tópicos-chave

Route 53 Resolver híbrido

Para a VPC resolver nomes do datacenter, use outbound endpoint com forwarding rule por domínio associada à VPC; para o datacenter resolver nomes da VPC ou de private hosted zones, use inbound endpoint com IPs em duas AZs e encaminhador condicional no DNS local. O Resolver da VPC fica no endereço base mais dois e só atende consultas de dentro dela. Regras de encaminhamento podem ser compartilhadas pelo AWS RAM com a organização.

Private hosted zones e split-view

Exigem enableDnsSupport e enableDnsHostnames ativos na VPC e podem ser associadas a várias VPCs, inclusive de outras contas. Em split-view, zonas pública e privada têm o mesmo nome e o Resolver prefere a privada; nome que não existe nela retorna NXDOMAIN sem consultar a pública. Repita na privada os nomes públicos que a VPC também precisa.

Políticas de roteamento do Route 53

Latência com alias e Evaluate Target Health dá desempenho e failover entre regiões; weighted divide por pesos de 0 a 255 para migração gradual; multivalue devolve até oito registros saudáveis com health check em cada; failover é ativo-passivo. Geolocation precisa de registro Default para IPs não mapeados. Alias A/AAAA funciona no apex, onde CNAME não é permitido, e acompanha as mudanças de IP do ALB.

Logs e firewall de DNS

Query logging de hosted zone pública grava no CloudWatch Logs com log group em us-east-1 e resource policy liberando route53.amazonaws.com; o Resolver query logging registra consultas feitas a partir das VPCs. O Route 53 Resolver DNS Firewall bloqueia resolução de domínios maliciosos com AWS Managed Domain Lists, começando em Alert e passando a Block.

NAT gateway

O NAT gateway público fica em sub-rede pública com Elastic IP; as sub-redes privadas apontam 0.0.0.0/0 para ele e a sub-rede dele aponta 0.0.0.0/0 para o internet gateway. É zonal: alta disponibilidade é um por AZ com tabela de rotas por AZ; um único reduz custo. Não tem security group, suporta até 55.000 conexões simultâneas por IPv4 para cada destino único, ou seja, IP, porta e protocolo (ErrorPortAllocation pede IPv4 secundários) e o NAT privado não alcança a internet.

Saída para a internet e IPs públicos

Instância em sub-rede pública ainda precisa de IPv4 público ou Elastic IP para usar o internet gateway. Para IPv6 só de saída, use egress-only internet gateway com rota ::/0. A AWS cobra por hora todo IPv4 público, inclusive Elastic IP não associado; o Public IP insights do IPAM localiza os endereços.

Security groups x network ACLs

Security groups são stateful e podem usar outro security group como origem, ideal para camadas com IPs dinâmicos. NACLs são stateless: liberou a entrada, libere a saída para as portas efêmeras do cliente (1024-65535 cobre NAT gateway e ELB). Regras de NACL são avaliadas da menor para a maior numeração, com a negação padrão por último.

VPC endpoints

Gateway endpoints existem só para S3 e DynamoDB, entram como rota para a prefix list e não têm custo adicional, tirando o tráfego do NAT gateway. Interface endpoints (PrivateLink) atendem os demais serviços, cobrados por hora e por GB. Pull do ECR sem internet exige ecr.api, ecr.dkr e gateway do S3 (e logs para o awslogs); API Gateway privado é REST API com interface endpoint execute-api.

PrivateLink e NLB na frente do ALB

Endpoint services do PrivateLink expõem um serviço específico, não a rede, e funcionam com CIDRs sobrepostos; são sustentados por NLB, que pode ter target group do tipo alb para preservar regras de camada 7. NLB com um Elastic IP por AZ e target group do tipo alb dá IP fixo na frente de um ALB; o contrário (ALB na frente de NLB) não é suportado.

Peering, Transit Gateway e VPC sharing

VPC peering liga duas VPCs, não é transitivo e não aceita CIDRs sobrepostos, nem com bloco secundário. Transit Gateway é hub com route tables próprias: precisa de rota na VPC de origem, rota na route table do TGW (propagação ou estática) e caminho de volta; é compartilhado pelo AWS RAM e cada conta cria o attachment. VPC sharing compartilha sub-redes pelo RAM, e os participantes não alteram rotas nem NACLs.

Site-to-Site VPN

Roteamento dinâmico com BGP retira as rotas do túnel que caiu e leva o tráfego ao outro; rotas estáticas não somem. A métrica TunnelState vale 1 para UP e 0 para fora; na dimensão da conexão, alarme abaixo de 1 avisa a perda de um túnel. VPN acelerada só existe com transit gateway e ainda passa pela internet.

Direct Connect

VIF privada liga a um virtual private gateway (direto ou via Direct Connect gateway); transit VIF liga a um Direct Connect gateway associado a transit gateways, para muitas VPCs; VIF pública acessa endpoints públicos da AWS. Resiliência pede conexões em locais distintos, com VPN de backup quando há um link só, e a AWS prefere rotas do Direct Connect às da VPN para o mesmo prefixo. Direct Connect não criptografa: use MACsec (dedicadas de 10, 100 e 400 Gbps) ou VPN IPsec sobre o Direct Connect.

Network Firewall, WAF e Shield

AWS Network Firewall filtra tráfego de rede; rule groups stateful de domain list usam SNI ou Host como allowlist, e a inspeção central de saída usa transit gateway e VPC de inspeção. AWS WAF protege HTTP em CloudFront, ALB e API Gateway (não em NLB): regras rate-based com scope-down, grupos gerenciados testados em Count e logs da web ACL. Shield Advanced dá acesso ao Shield Response Team, proteção de custos e proactive engagement, que exige health checks associados.

CloudFront: cache

A cache policy define o que entra na cache key; a origin request policy só define o que vai à origem. Incluir parâmetros que não mudam a resposta derruba o cache hit; omitir um cookie que personaliza a resposta entrega a página de um cliente a outro. O max-age da origem vale entre o Minimum e o Maximum TTL (Default TTL só sem cabeçalho), s-maxage vale para o CloudFront, e Minimum TTL acima de zero passa por cima de no-cache.

CloudFront: origem, erros e logs

OAC com bucket policy concedendo s3:GetObject a cloudfront.amazonaws.com com AWS:SourceArn da distribuição mantém o bucket privado (OAI é legado). 502 indica falha de TLS ou DNS com a origem (nome que não bate com o certificado); 504 indica origem lenta ou bloqueada. Viewer protocol policy Redirect HTTP to HTTPS leva o visitante ao HTTPS, nomes de arquivo versionados evitam invalidações, e real-time logs no Kinesis Data Streams trazem x-edge-result-type em segundos.

Load balancers e Global Accelerator

ALB decide na camada 7 (caminho e host) e adiciona X-Forwarded-For com o IP do cliente; NLB é camada 4 e aceita Elastic IP; Gateway Load Balancer é para appliances. Access logs do ALB trazem URL, status e tempos por requisição, e connection logs trazem dados do handshake TLS. Global Accelerator dá dois IPs estáticos anycast com entrada pela borda, aceita ALB interno como endpoint e não faz cache.

Diagnóstico e monitoramento de rede

VPC Reachability Analyzer analisa a configuração sem enviar pacotes e aponta o componente bloqueador; Flow Logs mostram ACCEPT ou REJECT do tráfego real, e o formato não muda depois de criado (pkt-srcaddr e pkt-dstaddr só em formato personalizado). Network Flow Monitor usa agentes para medir TCP real entre workloads; Network Synthetic Monitor usa probes ICMP ou TCP sem agentes até o on-premises; Internet Monitor mostra o impacto por cidade e ASN; canaries do Synthetics testam fluxos de usuário.

Custo de transferência

Tráfego entre AZs da mesma região é cobrado; dentro da mesma AZ por IP privado não é, o que favorece aproximar camadas que trocam muitos dados quando a resiliência permite. A transferência de origens na AWS para o CloudFront não é cobrada, então cache na borda reduz o custo de saída. Gateway endpoints eliminam a cobrança de processamento do NAT para S3 e DynamoDB.

Macetes para memorizar

Resolver

OUTbound leva a pergunta para fora; INbound traz a pergunta para dentro.

Saída para a internet

Privada aponta para o NAT, o NAT aponta para o IGW, IPv6 sai pelo egress-only.

Stateful x stateless

Security group lembra a conversa; NACL tem amnésia e precisa liberar a volta nas portas efêmeras.

Endpoints

Gateway é grátis e só para S3 e DynamoDB (S e D); o resto é interface e paga.

CloudFront

Cache policy decide a chave; origin request policy só leva a encomenda.

Erros do CloudFront

502 = aperto de mão TLS falhou; 504 = a origem não respondeu a tempo.

Monitoramento de rede

Flow Monitor = tráfego real com agente; Synthetic Monitor = sonda sem agente até o on-premises; Internet Monitor = usuário final por cidade e ASN.

Tabela de memorização: não confunda

Quando é...	...e quando é
Outbound endpoint do Resolver A VPC precisa resolver nomes do datacenter (com forwarding rule).	**Inbound endpoint do Resolver** O datacenter precisa resolver nomes da VPC ou de private hosted zones (com encaminhador condicional).
Gateway endpoint Acesso privado a S3 ou DynamoDB sem custo adicional, por rota.	**Interface endpoint** Acesso privado a qualquer outro serviço (CloudWatch, ECR, execute-api), pago por hora e por GB.
Security group Filtrar por instância de forma stateful, referenciando outros security groups.	**Network ACL** Filtrar por sub-rede de forma stateless, lembrando da regra de retorno para portas efêmeras.
VPC peering Poucas VPCs, topologia simples e CIDRs sem sobreposição.	**Transit Gateway** Muitas VPCs e redes on-premises com roteamento centralizado em hub.
CloudFront Conteúdo HTTP com cache na borda, WAF e redução do custo de saída.	**Global Accelerator** IPs estáticos anycast para allowlist e entrada pela borda, sem cache.
VIF privada Direct Connect até uma VPC pelo virtual private gateway (direto ou via Direct Connect gateway).	**Transit VIF** Direct Connect para dezenas de VPCs via Direct Connect gateway associado a transit gateway.
Network Flow Monitor Desempenho TCP do tráfego real entre workloads na AWS, com agentes.	**Network Synthetic Monitor** Latência e perda de pacotes entre a VPC e o on-premises com probes, sem agentes.

Números que caem

Item	Valor
Conexões simultâneas do NAT gateway por destino único	**até 55.000 por IPv4**
Multivalue answer do Route 53	**até oito registros saudáveis**
Pesos do weighted routing	**0 a 255**

Portas efêmeras para regra de retorno em NACL	**1024-65535 (Linux 32768-61000; Windows Server 2008 ou superior 49152-65535)**
Sub-rede /28	**16 endereços, 5 reservados pela AWS, 11 livres**
Bloco CIDR IPv4 adicional da VPC	**entre /16 e /28, sem sobreposição**
Resolver da VPC	**endereço base do CIDR mais dois**
MACsec no Direct Connect	**conexões dedicadas de 10, 100 e 400 Gbps**
Janelas das regras rate-based do WAF	**60, 120, 300 ou 600 segundos; limite mínimo de 10**
Malha completa de peerings	**n x (n-1) / 2 conexões**
Global Accelerator	**dois IPs estáticos anycast**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Criar inbound endpoint para a VPC resolver nomes do datacenter.
Como não cair: Inbound é o caminho inverso; consultas da VPC para fora usam outbound endpoint com forwarding rule.

⚠ **Alerta de prova:** Apontar o DHCP options set para o DNS do datacenter.
Como não cair: Toda a resolução passaria a depender do link e as VPCs perderiam nomes da AWS e de private hosted zones.

⚠ **Alerta de prova:** Adicionar security group ao NAT gateway ou mexer em DNS quando a saída não funciona.
Como não cair: NAT gateway não tem security group; confira a rota da sub-rede do NAT para o internet gateway.

⚠ **Alerta de prova:** Usar NAT gateway privado ou egress-only internet gateway para saída IPv4 à internet.
Como não cair: NAT privado não alcança a internet e egress-only é só IPv6; IPv4 privado sai por NAT gateway público.

⚠ **Alerta de prova:** Liberar a resposta da NACL de entrada achando que basta a regra de 443.
Como não cair: NACL é stateless: a resposta sai para a porta efêmera do cliente e precisa de regra de saída 1024-65535.

⚠ **Alerta de prova:** Criar peering transitivo por uma VPC intermediária ou com CIDR sobreposto usando bloco secundário.
Como não cair: Peering não é transitivo e não se cria com qualquer bloco sobreposto; para muitas VPCs, Transit Gateway.

⚠ **Alerta de prova:** Associar AWS WAF a um Network Load Balancer.
Como não cair: WAF protege HTTP em CloudFront, ALB e API Gateway; para o NLB, não há associação.

⚠ **Alerta de prova:** Usar regra stateless do Network Firewall ou security group para allowlist de domínios.
Como não cair: Filtragem por domínio é rule group stateful de domain list; security group só aceita CIDR, prefix list ou outro SG.

⚠ **Alerta de prova:** Enviar o cookie de sessão pela origin request policy para separar páginas personalizadas.
Como não cair: Só a cache policy muda a cache key; valor que personaliza a resposta precisa estar na cache key.

⚠ **Alerta de prova:** Recriar o certificado do ALB em us-east-1 para resolver o 502 do CloudFront.
Como não cair: 502 vem de TLS ou DNS com a origem; ajuste o domínio da origem para casar com o certificado do listener. A exigência de us-east-1 é para o certificado do viewer.

⚠ **Alerta de prova:** Alarme em TunnelState igual a 0 na dimensão da conexão para saber da queda de um túnel.
Como não cair: Nessa dimensão o valor só chega a 0 com os dois túneis fora; alarme abaixo de 1.

⚠ **Alerta de prova:** Usar VPN acelerada ou ECMP para ter latência previsível sem internet.
Como não cair: As duas continuam usando a internet pública; latência consistente e privada é Direct Connect.

⚠ **Alerta de prova:** Usar transit VIF ligada diretamente a um virtual private gateway.
Como não cair: Transit VIF vai a um Direct Connect gateway associado a transit gateways; VPC via virtual private gateway usa VIF privada.

⚠ **Alerta de prova:** Alterar o formato do flow log existente para ver o IP real por trás do NAT.
Como não cair: Flow log não muda depois de criado; crie outro com formato personalizado incluindo pkt-srcaddr e pkt-dstaddr.

Como cai na prova

📝 **Como cai:** Instância ou serviço sem conectividade e quatro mudanças possíveis.
Como resolver: Siga o caminho do pacote: IP público ou EIP, rota da sub-rede, rota da sub-rede do NAT para o IGW, security group (stateful), NACL (stateless, portas efêmeras) e rota de volta; descarte opções que mexem em DNS quando o problema é rota.

📝 **Como cai:** Resolução de nomes entre VPC, outras contas e datacenter.
Como resolver: Defina a direção da consulta (VPC para fora ou fora para dentro), escolha outbound ou inbound endpoint, confira os atributos de DNS da VPC e compartilhe regras ou associe zonas a várias VPCs em vez de duplicar servidores.

📝 **Como cai:** Conectividade híbrida com requisitos de latência, criptografia e resiliência.
Como resolver: Latência previsível e privada aponta Direct Connect; muitas VPCs pedem transit VIF com Direct Connect gateway e transit gateway; criptografia pede MACsec ou VPN sobre o Direct Connect; resiliência pede locais distintos ou VPN de backup com BGP.

📝 **Como cai:** CloudFront com cache baixo, conteúdo errado ou TTL inesperado.
Como resolver: Revise a cache key na cache policy (só o que muda a resposta), compare max-age com Minimum e Maximum TTL e prefira nomes de arquivo versionados a invalidações frequentes.

📝 **Como cai:** Escolher a ferramenta para investigar a rede.
Como resolver: Sem tráfego e querendo a regra culpada: Reachability Analyzer; tráfego real aceito ou rejeitado: Flow Logs; desempenho TCP: Network Flow Monitor; híbrido sem agente: Network Synthetic Monitor; usuários na internet: Internet Monitor; detalhe HTTP: access logs do ALB com Athena.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
private hosted zone não resolve nas instâncias	**enableDnsSupport e enableDnsHostnames da VPC**
mesmo nome com respostas diferentes dentro e fora da VPC	**Split-view: zonas pública e privada com o mesmo nome**
apex do domínio apontando para ALB ou CloudFront	**Registro alias A/AAAA**
migração gradual por porcentagem no DNS	**Weighted routing com TTL baixo**
ErrorPortAllocation no NAT gateway	**Associar IPv4 secundários ao NAT gateway**
custo alto de NAT gateway com tráfego para S3 ou DynamoDB	**Gateway endpoints nas tabelas de rotas privadas**

expor um serviço a muitas contas com CIDRs sobrepostos	**PrivateLink com NLB e target group do tipo alb**
IPs fixos para allowlist de parceiros	**Global Accelerator ou NLB com Elastic IP por AZ**
qual regra bloqueia o caminho entre A e B sem gerar tráfego	**VPC Reachability Analyzer**
conexões aceitas ou rejeitadas por security group ou NACL	**VPC Flow Logs (campo action)**
IP real do cliente atrás do ALB	**Cabeçalho X-Forwarded-For**
limitar requisições por IP em /login	**Regra rate-based do WAF com scope-down statement**
SRT ligando durante ataque DDoS	**Shield Advanced com health checks e proactive engagement**
conteúdo do S3 só pelo CloudFront	**Origin access control (OAC) e bucket policy com AWS:SourceArn**

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

- A sub-rede do NAT gateway precisa de rota 0.0.0.0/0 para o internet gateway, e NAT gateway não tem security group.
- NACL é stateless: toda liberação de entrada precisa da saída para as portas efêmeras.
- Peering não é transitivo nem aceita CIDR sobreposto; Transit Gateway precisa de rota nos dois sentidos e nas duas camadas (VPC e TGW).
- Cache key se define na cache policy; a origin request policy não muda o cache.
- Query logging de zona pública e certificado de viewer do CloudFront ficam em us-east-1.

Essenciais para a prova (checklist)

☐ Sei configurar outbound e inbound endpoints do Resolver, forwarding rules compartilhadas e DNS Firewall.

☐ Sei diagnosticar private hosted zones, split-view e escolher a política de roteamento do Route 53.

☐ Sei montar a saída IPv4 e IPv6 de sub-redes privadas com NAT gateway e egress-only internet gateway.

☐ Sei escrever regras de security group e NACL, inclusive retorno por portas efêmeras e referência entre SGs.

☐ Sei escolher gateway ou interface endpoint e os endpoints necessários para ECR e API Gateway privado.

☐ Sei quando usar peering, Transit Gateway, VPC sharing e PrivateLink e como depurar rotas do TGW.

☐ Sei desenhar VPN com BGP e Direct Connect com o tipo certo de VIF, resiliência e criptografia.

- ☐ Sei configurar cache policy, TTLs, OAC, HTTPS e logs no CloudFront e interpretar 502 e 504.
- ☐ Sei escolher entre WAF, Shield Advanced, Network Firewall e DNS Firewall para cada ameaça.
- ☐ Sei escolher a ferramenta de diagnóstico e de monitoramento de rede para cada sintoma.

Questões de revisão do capítulo

R5.1 · Domínio 5: Redes e entrega de conteúdo

Uma aplicação web em instâncias Amazon EC2 atrás de um Application Load Balancer, em uma única região, atende usuários do mundo todo. A empresa quer reduzir a latência para usuários distantes sem replicar a aplicação em outras regiões e proteger o ambiente contra ataques DDoS de rede e de aplicação, com suporte especializado da AWS durante ataques. Qual solução atende a esses requisitos com o menor esforço operacional?

A. Colocar um Network Load Balancer na frente do ALB, associar uma web ACL do AWS WAF ao NLB e usar roteamento por geolocalização do Route 53
B. Criar uma distribuição do CloudFront com o ALB como origem, associar uma web ACL do AWS WAF a ela e proteger os recursos com Shield Advanced
C. Habilitar o AWS Shield Advanced no ALB e criar um registro de alias do Route 53 com roteamento por latência apontando para o load balancer
D. Manter o ALB com uma web ACL do AWS WAF associada e configurar failover do Route 53 para um segundo ALB em standby em outra região da AWS

R5.2 · Domínio 5: Redes e entrega de conteúdo

Uma empresa com várias contas no AWS Organizations quer que o tráfego de saída para a internet das instâncias EC2 de VPCs de várias contas seja inspecionado e filtrado em um único ponto, com políticas de firewall consistentes, sem implantar e administrar um firewall em cada VPC e com crescimento simples à medida que novas contas e VPCs forem adicionadas. Qual abordagem atende com o menor esforço operacional contínuo?

A. Rotear as VPCs spoke por um transit gateway até uma VPC de inspeção central com o AWS Network Firewall, que aplica as políticas ao tráfego de saída
B. Implantar o AWS Network Firewall em cada VPC das contas-membro, associado às sub-redes de saída, com as políticas administradas por cada conta
C. Associar web ACLs do AWS WAF a um Application Load Balancer em cada VPC, com regras que inspecionam o tráfego de saída das instâncias para a internet
D. Padronizar os security groups de saída das instâncias por meio de runbooks do SSM Automation executados em cada conta-membro da organização

R5.3 · Domínio 5: Redes e entrega de conteúdo

Uma aplicação de três camadas tem a camada web em sub-redes públicas e as camadas de aplicação e de banco em sub-redes privadas de duas zonas de disponibilidade. As instâncias privadas precisam de saída para a internet e de resolução de nomes DNS públicos. Para reduzir custo, a empresa aceita perder a saída para a internet se uma zona falhar, mas a resolução DNS precisa continuar disponível. Qual configuração atende com o menor custo?

A. Criar um NAT gateway em cada zona de disponibilidade e um inbound endpoint do Route 53 Resolver para as instâncias resolverem os nomes públicos
B. Apontar a rota 0.0.0.0/0 das sub-redes privadas para o internet gateway e configurar resolvedores DNS públicos nas instâncias pelo user data
C. Criar uma instância NAT em uma zona e um servidor DNS próprio em EC2, configurado nas opções de DHCP da VPC no lugar do resolvedor da Amazon
D. Criar um NAT gateway zonal em uma das zonas, apontar a rota 0.0.0.0/0 das sub-redes privadas para ele e usar o Route 53 Resolver da VPC

R5.4 · Domínio 5: Redes e entrega de conteúdo

Uma aplicação HTTP roda atrás de um Application Load Balancer em cada uma de três regiões da AWS. A equipe quer que o Amazon Route 53 envie cada usuário à região com melhor desempenho de rede para ele e que, se o ALB de uma região ficar sem targets saudáveis, os usuários sejam direcionados automaticamente às outras regiões. Qual configuração do Route 53 atende a esses requisitos?

A. Registros do tipo A com roteamento baseado em latência apontando para os endereços IP atuais dos nós de cada ALB, com TTL de 300 segundos
B. Registros de alias com roteamento baseado em latência para o ALB de cada região, com Evaluate Target Health habilitado em cada registro
C. Registros de alias com roteamento de failover, tendo o ALB da região principal como primário e os ALBs das outras regiões como secundários
D. Registros de alias com roteamento por geolocalização por continente para cada ALB e um registro default apontando para a região principal

Respostas comentadas

R5.1. B) Criar uma distribuição do CloudFront com o ALB como origem, associar uma web ACL do AWS WAF a ela e proteger os recursos com Shield Advanced

Por quê: O Amazon CloudFront entrega o conteúdo a partir das edge locations e usa a rede da AWS até a origem, reduzindo a latência sem replicar a aplicação. Uma web ACL do AWS WAF associada à distribuição bloqueia ataques de camada 7 na borda, e o AWS Shield Advanced protege a distribuição e o ALB com detecção e mitigação avançadas de DDoS e acesso ao Shield Response Team.

✗ A) Incorreta. O AWS WAF não pode ser associado a Network Load Balancers, e um NLB na frente do ALB não reduz a latência de usuários distantes.

✗ C) Incorreta. Com um único ALB, o roteamento por latência não tem outra região para escolher, e os usuários distantes continuam atravessando a internet até a região.

✗ D) Incorreta. Failover do Route 53 aumenta a disponibilidade, mas não aproxima o conteúdo dos usuários distantes, e exige manter um segundo ambiente.

> **Dica de prova:** Latência global + DDoS + WAF para app em uma região = CloudFront com ALB como origem, WAF na distribuição e Shield Advanced. WAF não se associa a NLB.

R5.2. A) Rotear as VPCs spoke por um transit gateway até uma VPC de inspeção central com o AWS Network Firewall, que aplica as políticas ao tráfego de saída

Por quê: Na saída centralizada, as VPCs spoke enviam o tráfego ao transit gateway, que o encaminha a uma VPC de inspeção com endpoints do AWS Network Firewall e, depois, ao NAT gateway e ao internet gateway. As regras ficam em uma única política de firewall. O Network Firewall também pode ser anexado diretamente ao transit gateway (transit gateway-attached firewall), sem uma VPC de inspeção gerenciada pelo cliente.

✗ B) Incorreta. Um firewall por VPC multiplica custo e administração, e políticas locais em cada conta dificultam a padronização.

✗ C) Incorreta. O AWS WAF inspeciona requisições HTTP que chegam a recursos como ALB, CloudFront e API Gateway; ele não filtra o tráfego de saída das instâncias.

✗ D) Incorreta. Security groups filtram por IP, porta e protocolo em cada instância, sem inspeção de domínio ou de conteúdo e sem um ponto central de controle.

> **Dica de prova:** Inspeção central de saída em várias contas = transit gateway + Network Firewall centralizado (VPC de inspeção ou firewall anexado ao transit gateway). WAF é para HTTP de entrada.

R5.3. D) Criar um NAT gateway zonal em uma das zonas, apontar a rota 0.0.0.0/0 das sub-redes privadas para ele e usar o Route 53 Resolver da VPC

Por quê: O Route 53 Resolver (AmazonProvidedDNS), no endereço base da VPC mais dois, é integrado a cada zona de disponibilidade e resolve nomes públicos sem configuração extra. Para a saída IPv4, um único NAT gateway reduz o custo, ao preço de a saída depender da zona onde ele está; a recomendação de um NAT gateway por zona vale quando a saída também precisa ser altamente disponível.

✗ A) Incorreta. Um NAT gateway por zona aumenta o custo além do necessário, e inbound endpoints servem para redes externas consultarem a VPC, não para as instâncias resolverem nomes públicos.

✗ B) Incorreta. Uma rota para o internet gateway torna as sub-redes públicas, o que contraria o desenho com camadas privadas.

✗ C) Incorreta. A instância NAT e o servidor DNS próprio exigem gerenciamento, e o DNS em uma única instância vira ponto único de falha.

> **Dica de prova:** DNS na VPC já é redundante: use o Route 53 Resolver. NAT gateway único = menor custo; um por zona = alta disponibilidade da saída.

R5.4. B) Registros de alias com roteamento baseado em latência para o ALB de cada região, com Evaluate Target Health habilitado em cada registro

Por quê: O roteamento baseado em latência do Route 53 responde com o registro da região que oferece a menor latência para o usuário. Em registros de alias para load balancers, habilitar Evaluate Target Health faz o Route 53 considerar a saúde do ALB e dos seus targets e afastar o tráfego de recursos não saudáveis, o que dá failover automático entre as regiões.

✗ A) Incorreta. Os IPs dos nós de um ALB mudam com o tempo, e registros sem health check continuam respondendo com a região que falhou.

✗ C) Incorreta. O failover é ativo-passivo: todos os usuários vão para a região primária enquanto ela estiver saudável, sem considerar a latência de cada um.

✗ D) Incorreta. A geolocalização escolhe pelo local do usuário, não pelo desempenho medido, e sem avaliação de saúde não desvia de uma região com falha.

> **Dica de prova:** Melhor desempenho + failover entre regiões no DNS = roteamento por latência com alias para os ALBs e Evaluate Target Health.

DIA DA PROVA

Estratégia de prova

A prova do SOA-C03 é operacional: quase todo enunciado descreve um ambiente em produção com um sintoma ou um requisito e pergunta qual ação o engenheiro deve tomar. Os critérios de desempate aparecem no fim do enunciado, como menor esforço operacional, menor custo, sem alterar código, sem downtime ou sem que administradores locais contornem. Várias alternativas funcionam, mas só uma respeita todas as restrições; as erradas costumam ser soluções próprias com Lambda, recursos detectivos quando se pede prevenção ou serviços vizinhos com nome parecido. Os detalhes cobrados são de comportamento: o que cria recurso novo, o que é por região, o que é stateful e qual role ou política precisa da permissão.

Como responder

✓ Administre o tempo: são 65 questões em 130 minutos, cerca de 2 minutos por questão; marque as longas para revisão e siga em frente.

✓ Responda todas: 15 das 65 questões não são pontuadas e você não sabe quais, e não há penalidade por chute; nunca deixe em branco.

✓ Leia primeiro a última frase do enunciado e sublinhe as restrições (menor custo, menor esforço, sem código, sem downtime, preventivo) antes de olhar as alternativas.

✓ Em múltipla resposta (2 ou mais corretas entre 5 ou mais), avalie cada alternativa de forma independente e confira se a combinação escolhida resolve o cenário inteiro, como as duas pontas de uma permissão ou de uma rota.

✓ Elimine alternativas que recriam com Lambda ou scripts algo que já existe de forma nativa quando o enunciado pede menor esforço operacional.

✓ Classifique o pedido em prevenir, detectar ou corrigir: SCP e Deny previnem, Config e Security Hub detectam, EventBridge com SSM Automation corrige.

✓ Desconfie de alternativas com números fora do limite (target de 95% no DynamoDB, 8 instâncias em spread por AZ, retry de 3 na Lambda assíncrona): limites exatos derrubam distratores.

✓ A nota é de 100 a 1000 com mínimo de 720 em modelo compensatório: não precisa gabaritar todos os domínios, mas os três de 22% somam 66% da prova e merecem a maior parte do estudo.

Erros que derrubam candidatos

✗ Escolher a solução que funciona mas ignora a restrição do enunciado (mais cara, com código próprio ou com downtime).

✗ Confundir controles detectivos com preventivos, respondendo AWS Config quando o requisito é impedir.

✗ Esquecer que vários recursos são por região (criptografia padrão do EBS, DHMC, CDK bootstrap, eventos do AWS Health) e que serviços globais entregam eventos em us-east-1.

✗ Colocar a permissão no lugar errado: execution role em vez da resource-based policy, task role em vez da execution role, trust policy em vez de iam:PassRole.

✗ Achar que restaurações de RDS, DynamoDB ou ElastiCache voltam o recurso original, sem repontar a aplicação.

✗ Trocar serviços de nome parecido: Database Insights x Enhanced Monitoring, inbound x outbound endpoint, cache policy x origin request policy, zonal shift x routing controls.

✗ Usar chaves gerenciadas pela AWS em cenários de compartilhamento entre contas ou de key policy restrita.

Na semana da prova

- ☐ Refaça as provas práticas e, em cada erro, anote o detalhe que separava a certa da errada em vez de decorar a resposta.
- ☐ Revise as listas de números e limites de cada domínio (períodos, retenções, máximos por AZ, portas) no dia anterior.
- ☐ Releia as comparações e pegadinhas dos três domínios de peso 22: monitoramento, confiabilidade e implantação.
- ☐ Treine o tempo fazendo um simulado completo de 65 questões em 130 minutos, deixando cerca de 15 minutos para revisar as marcadas.
- ☐ Na véspera, revise só os resumos de lembrar de cada capítulo e durma bem; não comece assuntos novos.

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: Monitoramento, registro em log, análise, correção e otimização de desempenho

‣ memória, espaço em disco ou um processo específico da instância → **CloudWatch agent (mem_used_percent, disk_used_percent, procstat) com configuração no Parameter Store**

‣ Top 10 dinâmico de instâncias no dashboard → **Widget com consulta do Metrics Insights usando ORDER BY e LIMIT**

‣ qual consulta SQL pesa no banco, wait events, DB Load → **CloudWatch Database Insights**

‣ processos do SO de uma instância RDS em segundos → **Enhanced Monitoring (1 a 60 segundos)**

‣ BurstBalance chegando a 0% em gp2 pequeno → **Migrar para gp3 com Elastic Volumes e provisionar os IOPS**

⚠ O detalhado só muda a frequência para 1 minuto; o hipervisor não enxerga a memória. Memória e disco exigem o CloudWatch agent.

⚠ Ele só lê memória publicada pelo agente no namespace CWAgent com a dimensão InstanceId; o enhanced infrastructure metrics só amplia o período analisado.

⚠ No básico a métrica chega a cada 5 minutos, então haveria pontos faltando. Case o período com a frequência da métrica.

2. Domínio 2: Confiabilidade e continuidade dos negócios

‣ RTO e RPO em horas e menor custo → **Backup and restore com cópias para a região de DR e recriação por CloudFormation**

‣ RPO de minutos e RTO de horas com aplicação desligada → **Pilot light com réplica assíncrona do banco**

‣ RTO próximo de zero e gravação nas duas regiões → **Multi-site active/active com DynamoDB global tables e Route 53 com health checks**

‣ backups imutáveis que nem o root apaga, em outra região ou conta → **AWS Backup com Vault Lock em modo compliance e regra de cópia**

‣ FreeStorageSpace zerando no RDS → **Storage autoscaling com maximum storage threshold**

⚠ Ele só adiciona ou remove réplicas Aurora; para CPU e memória do writer acompanharem a carga, use Aurora Serverless v2 com faixa de ACUs.

⚠ Storage autoscaling só aumenta armazenamento (no RDS, com maximum storage threshold); não muda CPU nem memória.

⚠ Multi-Attach só funciona entre instâncias da mesma AZ; snapshots regionais (DLM ou AWS Backup) permitem recriar em outra zona.

3. Domínio 3: Implantação, provisionamento e automação

‣ trazer bucket ou tabela existente para uma stack sem recriar → **Resource import (change set IMPORT + DeletionPolicy + identificador)**

‣ InsufficientCapabilitiesException → **CAPABILITY_IAM, CAPABILITY_NAMED_IAM ou CAPABILITY_AUTO_EXPAND conforme o template**

‣ UPDATE_ROLLBACK_FAILED → **Continue update rollback, corrigindo a causa ou com resources to skip**

‣ stack presa esperando custom resource com Lambda na VPC → **A função não alcança a ResponseURL do S3: NAT ou gateway endpoint do S3**

‣ reverter a stack se um alarme da aplicação disparar após o deploy → **Rollback triggers com tempo de monitoramento**

⚠ validate-template só confere sintaxe; regras de segurança se validam com cfn-guard no pipeline.

⚠ Drift detection compara o que já foi implantado com o template; é pós-implantação.

⚠ Ela só impede excluir a stack; updates se controlam com stack policy e retenção com DeletionPolicy ou UpdateReplacePolicy.

4. Domínio 4: Segurança e conformidade

‣ proibir algo em todas as contas sem que admins locais contornem → **SCP na raiz ou na OU, com exceção por aws:PrincipalArn**

‣ desenvolvedores criando roles sem escalar privilégio → **Permissions boundary obrigatório via iam:PermissionsBoundary**

‣ terceiro que assume roles em várias contas de clientes → **sts:ExternalId na trust policy**

‣ acesso por classificação sem editar política a cada recurso novo → **ABAC com aws:ResourceTag igual a aws:PrincipalTag**

‣ menor privilégio a partir do uso real → **Geração de política do IAM Access Analyzer com base no CloudTrail**

⚠ SCP é teto, não concessão; sem política IAM concedendo, o AccessDenied continua.

⚠ SCPs nunca afetam usuários e roles da conta de gerenciamento.

⚠ Key policy de chave gerenciada pela AWS não é editável; copie ou recriptografe com chave gerenciada pelo cliente.

5. Domínio 5: Redes e entrega de conteúdo

‣ private hosted zone não resolve nas instâncias → **enableDnsSupport e enableDnsHostnames da VPC**

‣ mesmo nome com respostas diferentes dentro e fora da VPC → **Split-view: zonas pública e privada com o mesmo nome**

‣ apex do domínio apontando para ALB ou CloudFront → **Registro alias A/AAAA**

‣ migração gradual por porcentagem no DNS → **Weighted routing com TTL baixo**

‣ ErrorPortAllocation no NAT gateway → **Associar IPv4 secundários ao NAT gateway**

⚠ Inbound é o caminho inverso; consultas da VPC para fora usam outbound endpoint com forwarding rule.

⚠ Toda a resolução passaria a depender do link e as VPCs perderiam nomes da AWS e de private hosted zones.

⚠ NAT gateway não tem security group; confira a rota da sub-rede do NAT para o internet gateway.

GABARITO

Respostas do diagnóstico

D1. C) Criar uma Maintenance Window com agendamento cron no fuso America/Sao_Paulo, duração de 3 horas e cutoff de 30 minutos, com alvos pela tag e uma tarefa do tipo Automation

→ revise o capítulo 1

Por quê: Maintenance Windows definem quando ações potencialmente disruptivas podem rodar: agendamento cron ou rate com fuso horário, duração máxima, cutoff (tempo antes do fim em que não se iniciam tarefas novas), alvos registrados e tarefas. Os tipos de tarefa suportados são Run Command, Automation, Lambda e Step Functions.

✗ A) Incorreta. O rate control limita concorrência e erros; não impõe janela de duração nem cutoff, e o cron em UTC exige ajustar o fuso manualmente.

✗ B) Incorreta. O Run Command executa documentos de comando sob demanda; não tem agendamento próprio e não roda runbooks de Automation.

✗ D) Incorreta. O Change Calendar indica quando mudanças podem ou não ocorrer; ele não agenda nem dispara a execução do runbook.

> **Dica de prova:** Runbook em janela fixa com duração e cutoff: Maintenance Window com tarefa de Automation.

D2. C) Compartilhar o dashboard com os e-mails dos três auditores, para que cada um crie a própria senha de acesso → revise o capítulo 1

Por quê: O CloudWatch permite compartilhar um dashboard com até cinco e-mails (com senha individual, via Amazon Cognito), publicamente por link ou, para todos os dashboards, via provedor SSO com SAML. Quem recebe o compartilhamento ganha permissões como cloudwatch:GetMetricData e ec2:DescribeTags para a conta inteira, que não podem ser restritas a métricas específicas; por isso convém compartilhar com o menor grupo de pessoas possível.

✗ A) Incorreta. Contraria a restrição de não criar usuários IAM, e dar acesso ao console amplia a superfície de acesso.

✗ B) Incorreta. A observabilidade entre contas exige que os auditores tenham uma conta AWS monitora e acesso a ela.

✗ D) Incorreta. No compartilhamento público, qualquer pessoa que obtiver o link vê o dashboard, o que a segurança não aceita.

> **Dica de prova:** Dashboard para quem não tem conta AWS: compartilhamento do CloudWatch por e-mail (até 5), público ou via SSO.

D3. D) Parar as 8 instâncias do grupo, iniciá-las de novo e repetir o lançamento, o que pode levar o grupo a hardware com capacidade → revise o capítulo 1

Por quê: Em cluster placement groups, o ideal é lançar todas as instâncias em uma única solicitação e com o mesmo tipo; adicionar instâncias depois aumenta a chance de erro de capacidade. Quando o erro ocorre com instâncias já em execução, parar e iniciar todas e repetir o lançamento pode resolver. Para garantir capacidade, a AWS recomenda On-Demand Capacity Reservations no cluster placement group.

✗ A) Incorreta. Spread placement groups aceitam no máximo 7 instâncias por AZ e separam o hardware, abrindo mão da baixa latência do cluster.

✗ B) Incorreta. Um cluster placement group fica em uma única zona de disponibilidade e não pode abranger outra.

✗ C) Incorreta. Limite de conta gera InstanceLimitExceeded; InsufficientInstanceCapacity indica falta de capacidade física disponível naquele momento.

> **Dica de prova:** InsufficientInstanceCapacity em cluster placement group: parar e iniciar o grupo e relançar; para garantir, Capacity Reservation.

D4. D) Criar uma política do Data Lifecycle Manager com snapshots do volume a cada hora e, na falha, restaurar o snapshot mais recente em outra AZ → revise o capítulo 2

Por quê: O Amazon Data Lifecycle Manager automatiza a criação, a retenção e a exclusão de snapshots do EBS, com intervalos de 1, 2, 3, 4, 6, 8, 12 ou 24 horas ou com expressão cron. Os snapshots são armazenados de forma regional, então o volume pode ser recriado em qualquer AZ da região. Com snapshots horários, o RPO fica em até 1 hora, e o RTO depende de criar o volume e a instância, o que cabe em algumas horas.

✗ A) Incorreta. A replicação contínua atende ao RPO, mas mantém uma segunda instância e um volume ligados o tempo todo, com custo maior que o necessário para um RTO de horas.

✗ B) Incorreta. Com backups diários, a perda de dados pode chegar a 24 horas, acima do RPO de 1 hora.

✗ C) Incorreta. O Multi-Attach só anexa um volume io1 ou io2 a instâncias da mesma AZ; ele não protege contra a falha da zona.

> **Dica de prova:** RPO de horas com custo baixo para EBS = snapshots agendados pelo DLM (ou AWS Backup) na frequência certa. Multi-Attach não cruza AZ.

D5. A) Usar um Application Load Balancer nas sub-redes públicas das duas zonas, com as instâncias no target group e listener HTTPS com certificado do ACM → revise o capítulo 2

Por quê: Para tráfego web com alta disponibilidade, o Application Load Balancer em pelo menos duas zonas distribui as requisições entre os targets registrados e usa health checks para retirar os que falham. O listener HTTPS termina o TLS no balanceador com um certificado do ACM, centralizando o gerenciamento e a renovação dos certificados.

✗ B) Incorreta. Com o TLS terminado nas instâncias, cada servidor precisa do seu certificado instalado e renovado, o oposto do gerenciamento centralizado pedido.

✗ C) Incorreta. Um origin group alterna para a origem secundária apenas após falha da primária, não balanceia o tráfego entre instâncias e não substitui os health checks de um target group.

✗ D) Incorreta. O DNS de failover não distribui a carga entre as instâncias, depende do TTL para retirar um endereço com falha e exige atualizar registros a cada substituição de instância.

> **Dica de prova:** Web multi-AZ + health check de aplicação + TLS centralizado = ALB com listener HTTPS e certificado do ACM. Route 53 é DNS e não balanceia requisições entre instâncias.

D6. A) Configurar reserved concurrency de 100 na função, que limita as execuções simultâneas e reserva essa capacidade para ela → revise o capítulo 2

Por quê: Reserved concurrency reserva uma parte da concorrência da conta para a função e, ao mesmo tempo, define o teto de execuções simultâneas dela. É a forma de proteger recursos downstream, como conexões de banco. Provisioned concurrency resolve cold start, não limita. Para gerenciar muitas conexões, RDS Proxy é complementar.

✗ B) Incorreta. Escalar o banco para o pico aumenta o custo e não limita a função; o próximo pico poderia ser maior.

✗ C) Incorreta. Provisioned concurrency pré-inicializa ambientes, mas não limita a função; acima do valor provisionado, ela continua escalando com concorrência sob demanda.

✗ D) Incorreta. A cota é da conta inteira e afetaria as demais funções, contrariando o requisito.

> **Dica de prova:** Limitar uma função para proteger o banco = reserved concurrency. Provisioned concurrency = ambientes pré-aquecidos.

D7. B) Registrar o bloco como módulo do CloudFormation no registry privado e fazer os times o incluírem nos templates pelo nome do tipo terminado em MODULE → revise o capítulo 3

Por quê: Módulos do CloudFormation empacotam configurações de recursos reutilizáveis, ficam registrados no registry com controle de versão e podem ser usados em vários templates e contas. O template consumidor declara o módulo como um recurso cujo tipo termina em ::MODULE e pode combiná-lo com seus próprios recursos; quando o autor registra nova versão, a manutenção é centralizada.

✗ A) Incorreta. StackSets implantam a mesma stack em várias contas e regiões, mas não deixam o bloco ser combinado com outros recursos dentro do template de cada time.

✗ C) Incorreta. Copiar o template do produto cria cópias independentes em cada template; uma nova versão do bloco teria de ser recopiada por cada time.

✗ D) Incorreta. Forks criam divergência entre os times; cada mudança no bloco precisaria ser propagada para todas as cópias, e não há um tipo versionado para consumir.

> **Dica de prova:** Bloco reutilizável e versionado dentro de templates = CloudFormation modules. Mesma stack em muitas contas/regiões = StackSets. Catálogo de produtos aprovados para autoatendimento = Service Catalog.

D8. A) Usar uma AMI base com um script de user data que lê os parâmetros do Parameter Store na inicialização e aplica a configuração → revise o capítulo 3

Por quê: Separar a imagem da configuração evita reconstruir AMIs: a AMI base traz o sistema e a aplicação, e o user data, executado na primeira inicialização, lê os parâmetros com a role da instância (ssm:GetParameter) e aplica a configuração. O Parameter Store mantém o histórico de versões, e o script pode pedir uma versão específica no formato nome:versão.

✗ B) Incorreta. Automatiza a criação, mas continua gerando uma AMI por mudança, com custo e tempo de build a cada alteração de parâmetro.

✗ C) Incorreta. A instância pode ficar até um dia sem a configuração correta depois de lançada, o que não atende ao requisito de configurar no boot.

✗ D) Incorreta. Cada mudança exige gerar, testar e distribuir uma AMI nova, justamente o que a equipe quer evitar.

> **Dica de prova:** Configuração que muda muito não vai na AMI: AMI base + user data lendo o Parameter Store no boot. State Manager mantém o estado depois, mas com o atraso do agendamento.

D9. C) Criar um agendamento do EventBridge Scheduler com cron e fuso America/Sao_Paulo que inicie o runbook AWS-StopEC2Instance pela tag → revise o capítulo 3

Por quê: O EventBridge Scheduler cria agendamentos com expressões cron ou rate e fuso horário, e invoca destinos com templated targets ou universal targets (por exemplo, arn:aws:scheduler:::aws-sdk:ssm:startAutomationExecution). Chamando StartAutomationExecution com um runbook gerenciado como AWS-StopEC2Instance e destinos por tag, o Systems Manager Automation executa a ação e mantém o histórico por recurso.

✗ A) Incorreta. Exige código próprio e uma instância sempre ligada para o cron, além de não deixar histórico por instância no Systems Manager.

✗ B) Incorreta. Ações agendadas do Auto Scaling só atuam sobre Auto Scaling groups, e essas instâncias não pertencem a nenhum.

✗ D) Incorreta. Alarmes de CPU baixa não seguem horário e poderiam parar instâncias durante o expediente, além de exigir um alarme por instância.

> **Dica de prova:** Parar instâncias em horário fixo com histórico = EventBridge Scheduler (cron + fuso) iniciando um runbook do Automation.

D10. D) Excluir a chave exposta, verificar no CloudTrail e nos consoles o uso não autorizado e habilitar MFA no usuário root da conta → revise o capítulo 4

Por quê: A verificação Exposed Access Keys procura chaves publicadas em repositórios públicos e uso irregular de EC2; a ação recomendada é excluir a chave afetada e revisar a conta em busca de uso não autorizado (instâncias, Spot, usuários IAM, cobranças). A verificação MFA on Root Account alerta quando o root não tem MFA; em organizações, é possível centralizar e remover as credenciais root das contas-membro.

✗ A) Incorreta. Marcar como resolvido não corrige nada, e SCPs não se aplicam a uma conta fora de uma organização.

✗ B) Incorreta. Tirar a chave do repositório não a invalida, pois ela já pode ter sido copiada; tag policies padronizam tags e não exigem MFA.

✗ C) Incorreta. Manter ativa uma chave exposta prolonga o risco, e criar chaves de acesso para o root contraria as boas práticas.

> **Dica de prova:** Trusted Advisor com chave exposta: exclua a chave e investigue uso indevido. Root sem MFA: habilite MFA no root.

D11. D) Permitir iam:CreateRole com a condição iam:PermissionsBoundary igual ao ARN de DevMaxPermissions e negar a remoção ou troca do boundary → revise o capítulo 4

Por quê: Permissions boundaries definem o máximo de permissões que uma identidade pode ter; as permissões efetivas são a interseção entre as políticas da identidade e o boundary. Para delegar a criação de roles com segurança, permita a criação só com o boundary obrigatório (condição iam:PermissionsBoundary) e negue ações que removam ou alterem o boundary e a política que o define.

✗ A) Incorreta. A validação ajuda a escrever boas políticas, mas não impede um desenvolvedor de criar uma role com permissões amplas.

✗ B) Incorreta. A SCP limitaria todos os principais da conta, inclusive administradores e cargas de produção, e iam:* daria aos desenvolvedores controle sobre qualquer outra política.

✗ C) Incorreta. O controle é só detectivo: a role com privilégios excessivos já existiria quando o alerta chegasse.

> **Dica de prova:** Delegar criação de roles sem escalada: permissions boundary obrigatório via iam:PermissionsBoundary.

D12. A) Anexar SCPs às OUs com os controles obrigatórios e deixar que cada conta defina políticas IAM adicionais para os seus workloads → revise o capítulo 4

Por quê: Service control policies não concedem permissões: elas definem o limite máximo do que usuários e roles das contas podem fazer, inclusive administradores. Anexadas a OUs, aplicam-se às contas abaixo delas de forma central e auditável. Dentro desse limite, cada conta continua usando políticas IAM próprias para dar as permissões dos workloads.

✗ B) Incorreta. Uma política IAM gerenciada pertence a uma única conta e não pode ser anexada a roles de outras contas; além disso, administradores locais poderiam remover a boundary.

✗ C) Incorreta. Regras do Config são detectivas: avisam depois que a mudança aconteceu e não impedem que um administrador local interrompa o CloudTrail.

✗ D) Incorreta. O Security Hub CSPM avalia a configuração e gera achados, mas não bloqueia chamadas de API.

> **Dica de prova:** Guardrail preventivo multi-conta que nem o admin local derruba = SCP na OU. Config e Security Hub detectam; não bloqueiam.

Agora, treine no formato real

Faça as 6 provas completas do curso SOA-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.