

Não é possível criar um CNAME record para o top node of DNS namespace, conhecido como Zone Apex:

Se for registrado o DNS name para example.com, o zone apex será example.com. Entretanto é possível criar um ALIAS record para example.com que roteia o tráfego para [www.example.com](http://www.example.com)

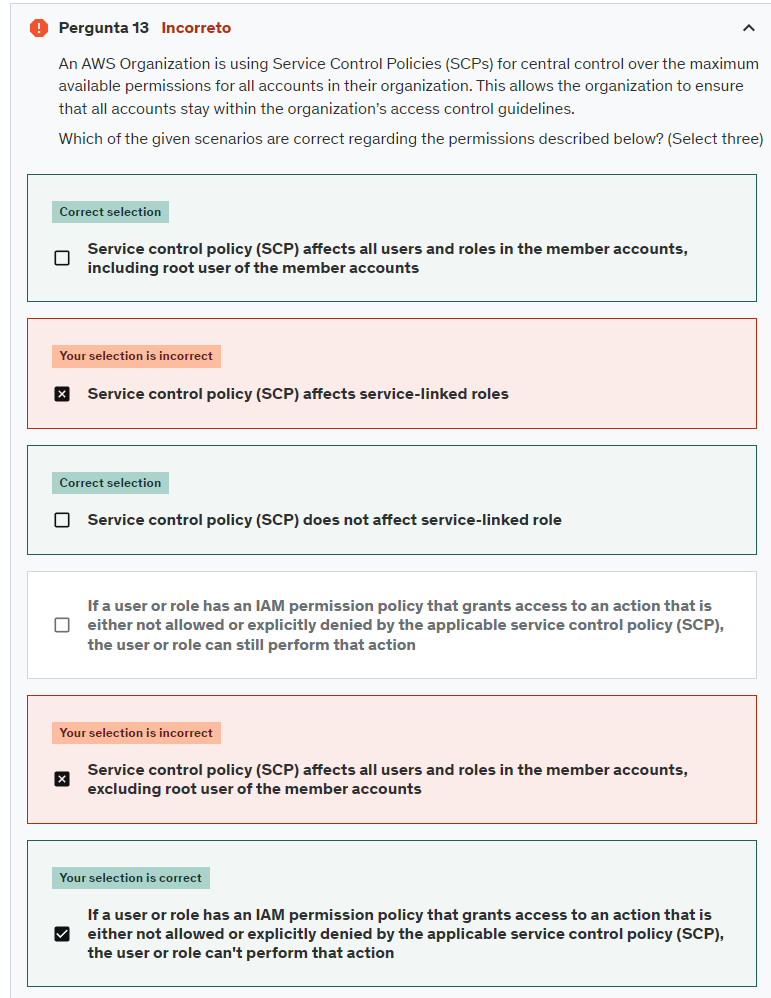
OUTRO: R53 não cobra por alias queries para recursos da AWS, mas CNAME queries é cobrado. E também, um alias record só pode redirecionar queries para recursos selecionados da AWS como Cloudfront, S3 Buckets e outro record na MESMA HOSTED ZONE. CNAMES contudo podem redirecionar DNS queries para qualquer DNS record.

A screenshot of a computer

Description automatically generated

Aurora (normal) não é fully managed e nem é uma solução de auto scaling completa.

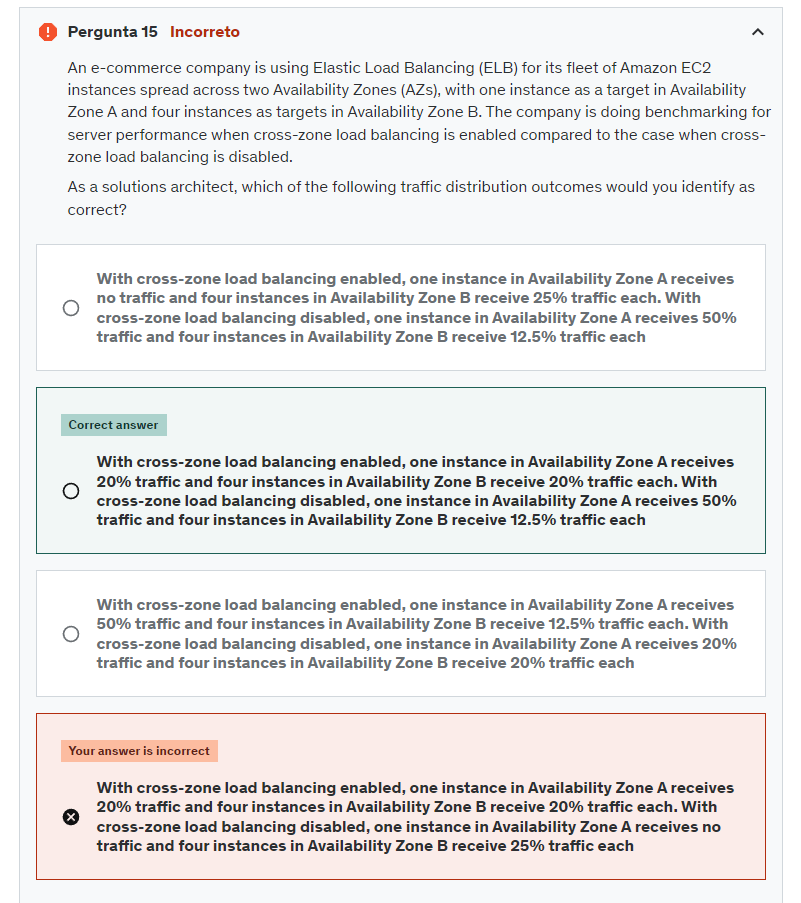
Aurora Serverless é uma solução “fully managed” o que remove do desenvolvedor a responsabilidade de provisionar estrutura, capacidade, ou outros pormenores. E providência autoscaling capabilities, designado para workloads variados regulando capacidade “up or down” baseado na demanda.



SCPs servem um propósito de restrição de permissões primariamente, que sobreescreve toda e qualquer permissão dada a usuários ou roles nas contas membro de uma organização INCLUINDO o usuário ROOT.

SCPs não afetam, contudo, a conta de gerenciamento ou a conta “raiz” da organização, de nenhuma forma, somente seus membros.

Service-linked roles (que são diferentes de Service Roles) não são afetadas pelas SCPs pois somente os serviços em si podem usá-las, Administradores por exemplo, só podem ver, mas não modificar.



Erro por falta de atenção

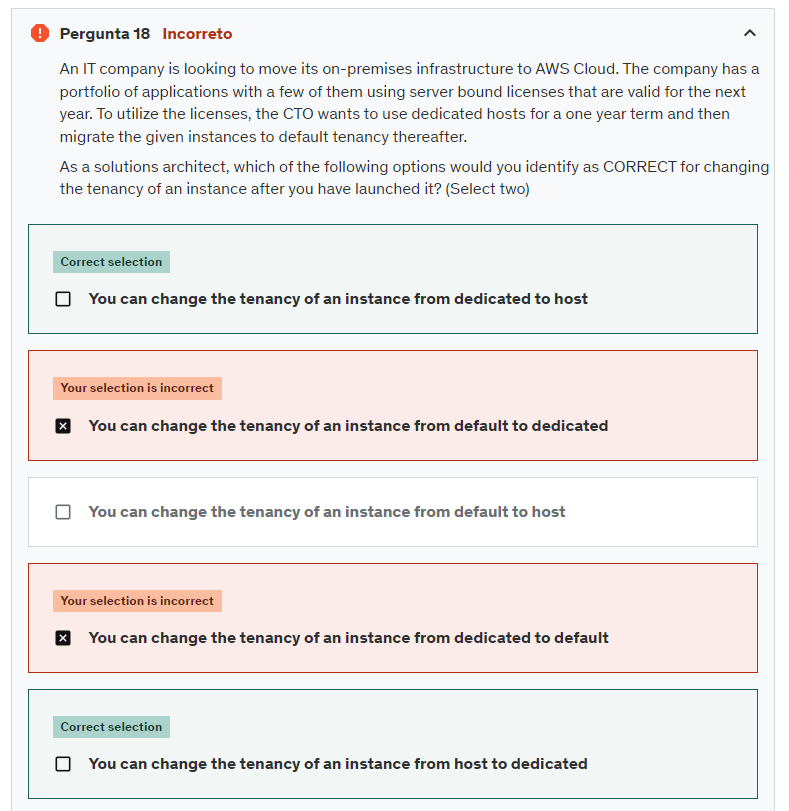
A screenshot of a computer

Description automatically generated

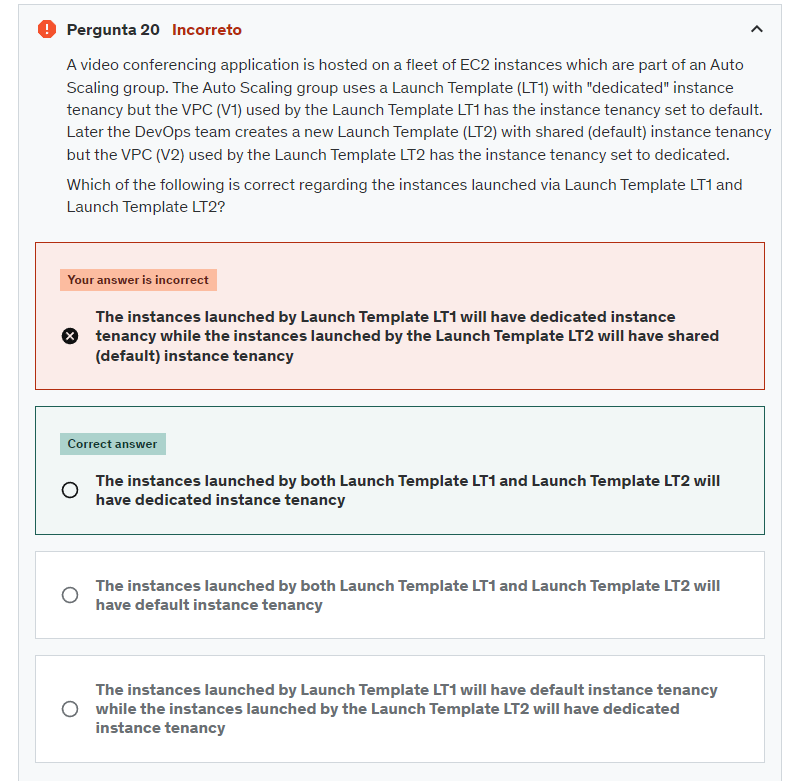
Connection Draining permite que o loadbalancer complete as requisições em tráfego (in-flight requests) feitas a instancias que estão sendo deregistradas ou estão unhealthy.

Quando o timeout máximo chega no limite, o loadbalancer fecha as conexões pendentes a força para as instâncias sendo deregistradas.

Connection Draining é a melhor solução para o problema, pois previne o “drop” das requests, pois mesmo quando a instância se torna unhealthy, o loadbalancer aguarda que a requisição seja completada.



As respostas são estas, mas até o momento desta revisão eu não pude encontrar as fontes que confirmem as respostas na doc oficial.



A questão é entender a ordem de prioridade do tenancy de uma instância:

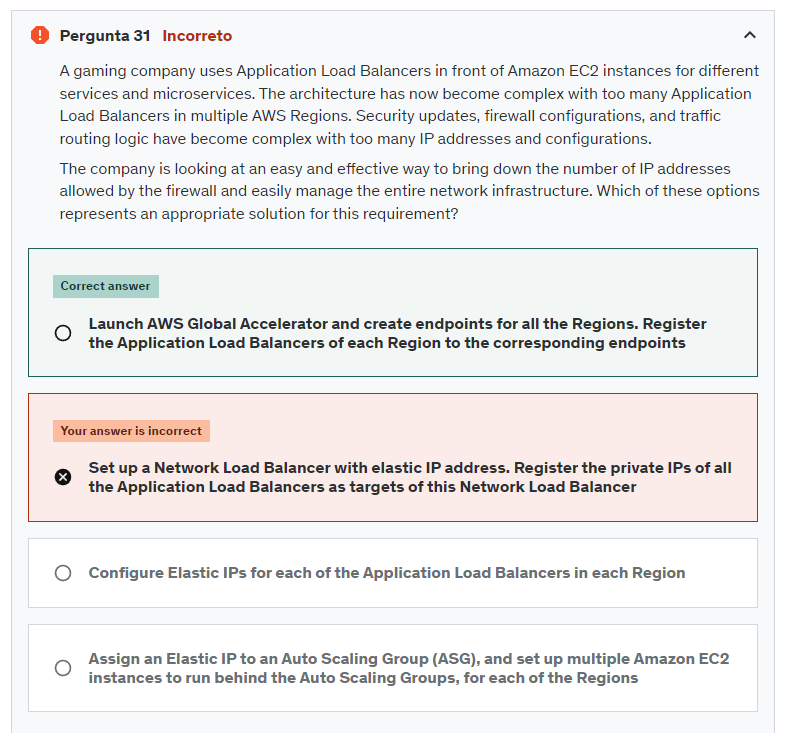
1. Se o launch template especificar um tenancy, então o tenancy daquela instância será sobreescrito, independentemente do tenancy configurado na VPC
2. Se o launch template não especificar o tenancy, or configurá-lo como “default”, a instância herdará o tenancy da VPC

A screenshot of a computer

Description automatically generated

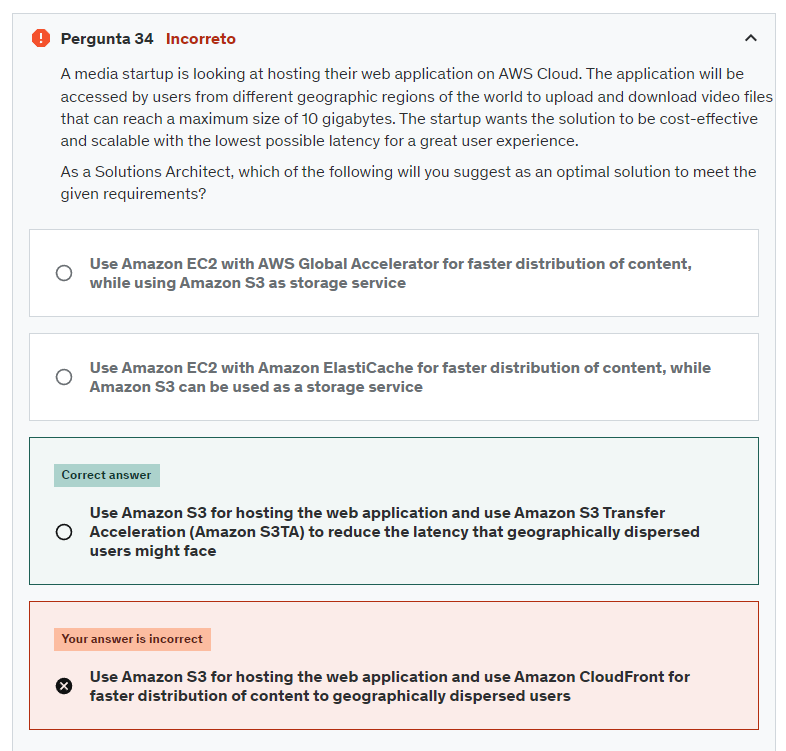
S3 somente suporta requests através de standards-based REST web services interfaces e não suporta SMB protocol.

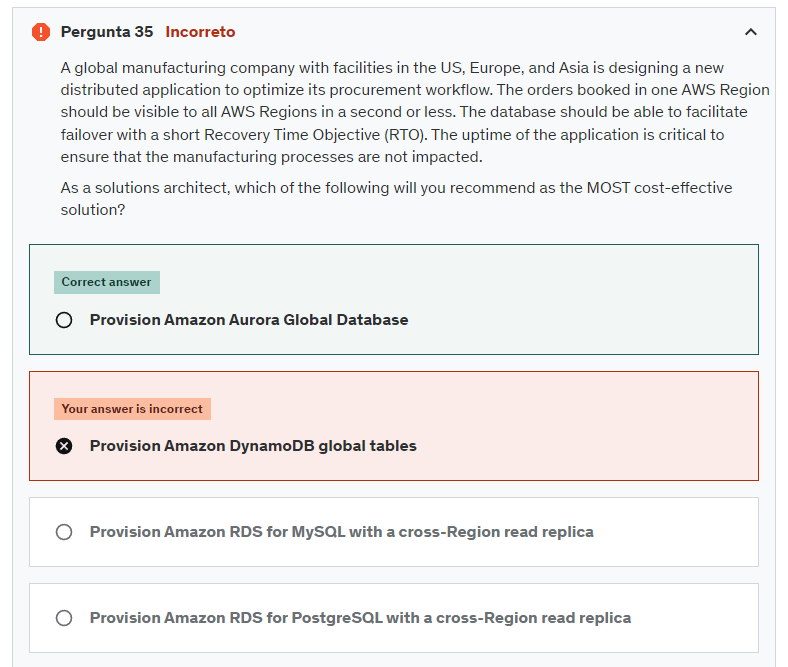
File Gateway ou S3 File Gateway suporta NFS protocol e SMB protocol



Para esta solução, Global Accelerator seria melhor pedida pois utiliza dois Ips globais e estáticos (utilizando Anycast IP) que serão user-facing, o que simplificará o gerenciamento de tráfego.

Simplesmente adicione ao backend do Ips suas AWS application origins, como NLB, ALB, Static Ips ou EC2 Instances

O problema relatado aqui é na diminuição da latência entre o bucket S3 e o usuário final. Utilizar S3TA ajuda tanto em upload quanto em DOWNLOAD, porque se utiliza da rede interna da AWS ou AWS Backbones/Network



Aurora é a melhor opção na questão pois seu RTO se dá em questão de minutos, ou menos, e principalmente por conta do REPLICATION LAG. DynamoDB global tables são automaticamente replicadas para todas as regiões, porém, seu tempo de replicação pode facilmente exceder os 1 segundos requeridos pela questão.

A screenshot of a survey

Description automatically generated

Para entender esta questão:

Request Types:

1. One-time: a spot instance é lançada, e permanece até que, o usuário a termine ou seja reclamada pela AWS
2. Persistent: a spot instance é lançada, e quando termina por quaisquer motivos, a AWS tenta repor a instância, pois o request continua ativo

Cancelar qualquer request, seja one-time ou persistent, não terminará nenhuma instancia que já foi provisionada, elas precisam ser manualmente terminadas. Cancelamento de request somente previne que NOVAS spot instances sejam lançadas

A screenshot of a web page

Description automatically generated

KDS é a melhor opção pois, os shards armazenam os records e o KDS providencia ordeção de records, assim como a habilidade de read/replay records na mesma ordem.

KDS também armazena dados por até 365 dias, o que possibilita com que o clickstream seja auditado após 7 dias.

EXAM TIP:

1. KD Firehose carrega streaming data em data stores como o Redshift, S3, OpenSearch, Splunk, HTTP Endpoints
2. KD Analytics é pra quando precisa-se rodar queries SQL ou construir aplicações JAVA

A screenshot of a computer

Description automatically generated

O processo é simples, R53 tem duas features: inbound and outbound endpoints.

1. Para rotear tráfego do servidor on-premise para VPC:
   1. Inbound endpoint no Route53
   2. DNS resolvers no servidor on-remises pode enviar DNS queries para este endpoint específico
2. Para rotear tráfego da VPC para o servidor on-premise:
   1. Outbound endpoint no Route53
   2. R53 pode condicionalmente (conditionally) enviar queries para os resolvers no servidor on-premise por este endpoint
   3. “conditionally forward queries” é preciso criar Resolver rules que especifica os domain names para as DNS queries que você quer enviar e o IP address do DNS resolver no servidor on-premise para onde as queries irão

A screenshot of a computer

Description automatically generated

Basicamente a resposta foi que: se você está utilizando ECS utilize a métrica do ECS

A screenshot of a computer

Description automatically generated

Volume Gateway tem dois modos: cached mode and stored mode.

1. Cached mode armazena todo o volume em um bucket S3 e dados acessados recentemente, ele retém no gateway’s local cache for low-latency access
2. Stored mode armazena todo o dataset on-site, e é assincronamente também armazenado no S3

A screenshot of a computer

Description automatically generated

Trocar para um NoSQL database introduz mudanças significativas no código, pois todas as queries precisariam ser reescritas, os endpoints trocados.

As outras opções introduzem demasiado trabalho também.

A melhor opção é trocar do MySQL convencional pelo Aurora

A screenshot of a computer

Description automatically generated

Se uma instância falha o SYSTEM status check, é possível utilizar um CloudWatch alarm para executar uma recuperação automática da instância.

Instâncias terminadas não podem ser recuperadas. Uma instancia recuperada é identica a instancia original, incluindo o instance ID, private IP adresses, EIPs, e todo o metadata da instancia, se a instancia estava em um placement group, a instancia recuperada rodará no placement group.

PS:. Recovery option works only for system check failures not for instance status check failures!!!!

A screenshot of a computer

Description automatically generated

CHUTEI BEM ABESSA!!

1. AD Connector use case:
   1. If you only need to allow you on-premises users to log in to AWS applications with their AD credentials.
2. Simple AD use case:
   1. Least expensive option
   2. If you have 5000 or fewer users and don’t need the more advanced Microsoft AD features such as trust relationships with other domains
3. AWS Managed Microsoft AD use case:
   1. Directory-aware workloads in the AWS Cloud such as SQL Server-based applications
   2. Trust relationship between AWS Managed Microsoft AD and your existing on-remises Microsoft AD, providing users and groups with access to resources in either domain, using single sign-on (SSO).

A screenshot of a computer

Description automatically generated

Scale up é pra aumentar o tamanho da instancia

Scale out é pra aumentar a quantidade de instancias

A screenshot of a computer

Description automatically generated

Target Tracking Scaling Policy fica observando a métrica customizada criada para o SQS, e a partir daquela métrica, ele aumenta ou diminui as instância de forma necessário e rápida.

Se a métrica determina que há poucas requests na fila, ele pode diminuir as instancias, e o inverso também pode acontecer.

A screenshot of a computer

Description automatically generated

Quando a route table é atualizada, todo e qualquer tráfego originado da VPC que tenha como destino o S3, será roteado pelo VPC gateway endpoint diretamente, sem a necessidade de um NAT gateway, ou acessar a internet para tal.

A screenshot of a mail

Description automatically generated

Delay queues permite atrasar a entrega de novas mensagens para consumidores. Para os consumers, mesmo que haja requests na fila, elas se tornam “invisiveis”.

Delay queues são utilizadas para processamento de requests que podem demandar mais tempo que o normal para serem completadas.

A screenshot of a computer

Description automatically generated

Requisições entre o Cloudfront e o S3 são grátis, somente paga-se por aqui que é entregue para a internet a partir do Cloudfront, reduzindo a carga no S3.

Esta opção também engloba a requisição de baixa latencia, pois o Cloudfront armazena o arquivo que está sendo solicitado nas edge locations mais próximas do usuário final

A screenshot of a computer

Description automatically generated

Cloudformation stacksets permite a criação de recursos iguais em diversas contas e regiões.

A screenshot of a computer

Description automatically generated

EFA é a melhor escolha de interface de rede anexadas a instancias EC2 para High Performance Computing, pois ele melhora a performance de inter-instance communication que é critico para o escalonamento de HPC e machine learning applications.

EFA tem tudo que o ENA tem. A escolha entre um e outro depende das especificações do workload, em caso de workloads menos latency-sensitive, ENA é uma opção melhor pois é mais barato. Também, quando não são necessárias as features avançadas do EFA, como OS-bypass e Message Passing Interface (MPI)