RESUMO EXPLICAÇÕES:

QUESTÃO 1:

Uma agência governamental planeja armazenar documentos fiscais confidenciais na AWS. Devido às informações confidenciais contidas nos arquivos, o Arquiteto de Soluções deve restringir as solicitações de acesso a dados feitas à solução de armazenamento a uma VPC específica da Amazon. A solução também deve impedir que os arquivos sejam excluídos ou substituídos para atender ao requisito regulatório de ter um modelo de armazenamento WORM (gravação única, leitura múltipla).
Qual combinação das seguintes opções o Arquiteto deve implementar? (Selecione DUAS.)

Crie um novo bucket do Amazon S3 com o recurso Bloqueio de Objetos do S3 habilitado. Armazene os documentos no bucket e defina a opção "Retenção Legal" para retenção de objetos.

Configure um ponto de acesso do Amazon S3 para o bucket S3 para restringir o acesso aos dados somente a uma VPC específica.

A pergunta exige uma solução que:
• Restrinja o acesso ao armazenamento somente a uma VPC específica
• Impeça a exclusão ou modificação de dados (modelo WORM)
• Esteja em conformidade com regulamentações governamentais de proteção de dados

✓ O S3 Object Lock (Bloqueio de Objetos):
• Permite armazenar objetos no modo WORM
• Suporta retenção por tempo definido e retenção legal, que impede modificações e exclusões
• Atende aos requisitos regulatórios de integridade e imutabilidade de dados

✓ O S3 Access Point com restrição à VPC:
• Permite criar um endpoint dedicado para acesso ao bucket
• Restringe o acesso ao bucket somente a uma VPC específica, conforme exigido na questão
• Melhora o controle de acesso sem alterar a política global do bucket

Explicação Corretas:

- **Pontos de Acesso S3:** Endpoints para buckets com políticas próprias; podem restringir acesso via VPC e bloquear acesso público.
- **Multirregionais:** Endpoint global que usa AWS Global Accelerator para direcionar ao bucket mais próximo, com menor latência.
- **Object Lock:** Modelo WORM que impede exclusão/alteração de objetos por tempo definido ou indefinido; obrigatório usar versionamento; irreversível após habilitação.

Explicação Incorretas:

- **Firewall de Rede da AWS:** Não é usado para restringir diretamente o acesso a um bucket do S3 de uma VPC. Para isso, o método correto é usar um **Ponto de Acesso do Amazon S3**. Um Firewall de Rede é integrado à sua VPC, não ao bucket.

- **S3 Glacier Instant Retrieval e Políticas de Acesso:** Embora essa classe de armazenamento seja para dados de longa duração, ela não é a melhor escolha para documentos de acesso frequente. Para restringir o acesso de uma VPC, a forma mais eficiente é usar um **Ponto de Acesso do S3**, e não apenas uma política de bucket.

- **Bloqueio de Objetos (Object Lock):** Para usar o **Object Lock**, que fornece o modelo de armazenamento WORM (Write Once, Read Many), é obrigatório que o **Controle de Versão de Objetos** também esteja habilitado. Você não pode desativar o versionamento se o Object Lock estiver ativo.

QUESTÃO 2:

Uma empresa hospedava um site de comércio eletrônico em um grupo de Auto Scaling de instâncias do Amazon EC2, atrás de um Application Load Balancer. O Arquiteto de Soluções notou que o site estava recebendo um alto número de solicitações externas ilegítimas de vários sistemas com endereços IP que mudavam com frequência. Para resolver os problemas de desempenho, o Arquiteto de Soluções precisa implementar uma solução que bloqueie essas solicitações, com impacto mínimo no tráfego legítimo.
Qual das seguintes opções atende a esse requisito?

Crie uma regra baseada em taxa no AWS WAF e associe a ACL da Web a um Application Load Balancer.

A PERGUNTA EXIGE UMA SOLUÇÃO QUE:
• Bloqueie requisições suspeitas e abusivas
• Minimize o impacto em usuários legítimos
• Funcione com IPs que mudam com frequência
• Seja eficaz contra picos de tráfego malicioso

✓ O AWS WAF com regras baseadas em taxa:
•Monitora a taxa de requisições por IP em tempo real
•Bloqueia IPs que excedem limites configurados
•Protege contra ataques de negação de serviço em nível de aplicação (Layer 7)
•É altamente configurável e granular, permitindo regras específicas
•Pode ser associado diretamente ao ALB, garantindo proteção integrada
•Não afeta tráfego legítimo se corretamente calibrado

Explicação Corretas:

- **O AWS WAF** protege aplicações integrando-se ao CloudFront, ALB, API Gateway e AppSync.

- No **CloudFront**, as regras são aplicadas nos Edge Locations, bloqueando acessos maliciosos antes de chegar ao servidor sem prejudicar o desempenho.
- Nos serviços regionais (ALB, API Gateway, AppSync), as regras atuam diretamente na região, protegendo recursos internos e externos.
- Para limitar acessos excessivos, usa-se regras baseadas em taxa, que controlam o número de requisições por IP em intervalos de 5 minutos. Assim, IPs que ultrapassarem o limite são temporariamente bloqueados até voltarem ao normal.
- Esse tipo de regra deve ser configurado em uma ACL da Web, que depois é associada ao recurso (ex.: ALB).

Explicação Incorretas:

- **AWS WAF:** É a ferramenta mais adequada para gerenciar e limitar a taxa de requisições. Você deve usar uma **regra baseada em taxa** para isso, e não uma regra comum.
- **Listas de Controle de Acesso de Rede (NACLs):** Não são ideais para limitar requisições de IPs dinâmicos, pois são mais eficientes para bloquear tráfego de IPs estáticos.
- **Grupos de Segurança:** Servem apenas para **permitir** tráfego de entrada e não podem ser usados para **negar** ou limitar a taxa de requisições.

QUESTÃO 3:

Uma empresa precisa que todos os dados armazenados na nuvem sejam criptografados em repouso. Para integrar isso facilmente com outros serviços da AWS, é necessário ter controle total sobre a criptografia das chaves criadas e também a capacidade de remover imediatamente o material da chave do AWS KMS. A solução também deve ser capaz de auditar o uso da chave independentemente do AWS CloudTrail.
Qual das seguintes opções atenderá a esse requisito?

Use o AWS Key Management Service para criar uma chave KMS em um armazenamento de chaves personalizado e armazenar o material de chave não extraível no AWS CloudHSM.

A PERGUNTA EXIGE UMA SOLUÇÃO QUE:
• Garanta criptografia em repouso
•Permita integração com outros serviços AWS
•Ofereça controle total sobre as chaves, incluindo a remoção imediata do material criptográfico
•Suporte auditoria independente do CloudTrail

✓ O uso de uma Custom Key Store com AWS CloudHSM oferece:
•Chaves gerenciadas pelo cliente com controle total sobre o ciclo de vida do material criptográfico
•Capacidade de remover imediatamente o material da chave (desligando ou excluindo o HSM)
•Auditoria independente, pois o cliente pode configurar logs diretamente no CloudHSM
•Integração com serviços AWS como S3, RDS, EBS, etc., usando KMS

Explicação Corretas:

O **AWS KMS** com repositório de chaves personalizado permite usar seu próprio cluster **CloudHSM** para armazenar e gerenciar chaves em vez do KMS padrão. Assim, o material das chaves nunca sai do HSM, garantindo mais segurança e controle. Esse recurso é útil quando:

- É preciso chaves em HSM dedicado sob controle próprio.
- É necessário revogar/remover chaves rapidamente e de forma independente.
- Há exigências de conformidade que pedem auditoria independente do KMS/CloudTrail.

Explicação Incorretas:

Para ter total controle sobre a criptografia e gerenciar as chaves de forma independente, você deve usar o **AWS Key Management Service (KMS)** em conjunto com o **AWS CloudHSM**.

- O **AWS CloudHSM** é o serviço correto para armazenar o material de chave de forma segura, pois oferece um **módulo de segurança de hardware (HSM)** dedicado. O **Amazon S3** não é adequado para essa finalidade.
- As **chaves gerenciadas pela AWS** ou de **propriedade da AWS** não permitem o controle total que o cenário exige, pois a AWS gerencia o ciclo de vida dessas chaves. Apenas um **repositório de chaves personalizado** (custom key store) com o CloudHSM permite o controle e a auditoria independentes.

QUESTÃO 4:

Uma empresa está migrando seus aplicativos para a AWS. Um de seus sistemas requer um banco de dados com capacidade de escala global e que lide com mudanças frequentes de esquema. O aplicativo não deve apresentar tempo de inatividade ou problemas de desempenho sempre que houver uma mudança de esquema no banco de dados. Ele também deve fornecer uma resposta de baixa latência para consultas de alto tráfego.
Qual é a solução de banco de dados mais adequada para atender a esse requisito?

Amazon DynamoDB

A PERGUNTA EXIGE UMA SOLUÇÃO QUE:
• Escale globalmente
• Suporte mudanças frequentes de esquema
• Não gere downtime ou degradação de performance com mudanças
• Atenda grandes volumes de tráfego com baixa latência

✓ O Amazon DynamoDB é:•Um banco de dados NoSQL gerenciado, com arquitetura serverless

• Altamente escalável globalmente com o recurso Global Tables
• Esquema flexível, permitindo alterações sem interrupção ou reestruturação
• Projetado para baixa latência de milissegundos, mesmo sob carga pesada
• Ideal para aplicações modernas que exigem alta disponibilidade e desempenho

Explicação Corretas:

Esquema é a estrutura/modelo dos dados em um banco.

- **Relacional (SQL):** esquema rígido, exige normalização e várias tabelas, usa joins caros, segue ACID, pouco flexível e difícil de escalar. Bom para consultas complexas com muitos relacionamentos.

- **NoSQL (ex: DynamoDB):** esquema flexível, fácil adicionar/remover dados, suporta dados hierárquicos em um único item, evita joins, tem baixa latência e é altamente escalável.

- **DynamoDB:** usa chaves compostas para organizar dados relacionados e otimizar consultas mais comuns, equilibrando custo e desempenho.

Explicação Incorretas:

A instância **Amazon RDS Multi-AZ** e o **Amazon Aurora com réplicas de leitura** não são incorretos por serem bancos de dados relacionais. Eles são serviços da AWS que oferecem alta disponibilidade e escalabilidade.

O **Amazon Redshift** está incorreto porque é um **data warehouse** projetado para cargas de trabalho de **OLAP (Processamento Analítico Online)** e não para bancos de dados transacionais.

QUESTÃO 5:

Uma plataforma de negociação de criptomoedas utiliza uma API construída em AWS Lambda e API Gateway. Devido às notícias e rumores recentes sobre a iminente alta de preços do Bitcoin, Ethereum e outras criptomoedas, espera-se que a plataforma de negociação tenha um aumento significativo de visitantes e novos usuários nos próximos dias.
Nesse cenário, como você pode proteger os sistemas de backend da plataforma contra picos de tráfego?

Habilite limites de limitação e armazenamento em cache de resultados no API Gateway.

A pergunta exige uma solução que:
• Proteja o backend (Lambda e outros serviços) contra sobrecarga em momentos de pico
• Reduza a latência e melhora a experiência do usuário
• Mantenha resiliência e escalabilidade sob demanda variável

✓ O uso de throttling e caching no API Gateway:
• Throttling limita o número de requisições por segundo, evitando sobrecarga nas funções Lambda e nos sistemas downstream•Caching armazena temporariamente as respostas de APIs, reduzindo chamadas repetitivas ao backend
• Ambos os recursos são fáceis de configurar diretamente no API Gateway
• Reduzem custos, já que evitam execuções desnecessárias do Lambda
• Melhoram a performance e estabilidade em cenários de tráfego explosivo

Explicação Corretas:

Amazon API Gateway permite **controle de fluxo** definindo limites de taxa (ex.: 1.000 req/s) e picos temporários (ex.: 2.000 req/s). Se passar do limite, retorna HTTP 429, e os SDKs podem repetir a chamada automaticamente.
Além disso, é possível usar **cache em estágios específicos**, escolhendo tamanho, chave e TTL, o que melhora desempenho e reduz tráfego no back-end. Também há APIs para **invalidar o cache** quando necessário.

Explicação Incorretas:

As informações fornecidas estão incorretas em sua totalidade. As afirmações são contraditórias e imprecisas. Vamos ver o porquê:

- A afirmativa sobre **Lambda, API Gateway, EC2, ELB e Auto Scaling** está incorreta. A arquitetura com Lambda e API Gateway já é por si só altamente escalável e disponível. Mudar para uma arquitetura baseada em EC2 com ELB e Auto Scaling pode, na verdade, aumentar a complexidade de gerenciamento e o custo, sem necessariamente trazer benefícios de escalabilidade ou disponibilidade para a maioria dos casos de uso de microsserviços e aplicações serverless.

- A afirmativa sobre **CloudFront e API Gateway** está incorreta. Usar o CloudFront antes do API Gateway é uma prática comum e correta para melhorar o desempenho e a segurança. O CloudFront funciona como um cache de borda (edge caching) para as respostas da API, o que não apenas acelera a entrega de conteúdo (reduzindo a latência para o usuário), mas também alivia a carga do backend do API Gateway e do Lambda.

- A afirmativa sobre **Lambda e VPC** está incorreta. Mover uma função Lambda para dentro de uma VPC (Virtual Private Cloud) é uma prática comum e muitas vezes necessária. Ela é usada para que a função Lambda possa acessar recursos de rede privados, como bancos de dados em uma sub-rede privada, ou serviços com endpoints privados, como uma API que está na mesma VPC. É uma configuração relevante para muitos cenários de aplicação.

QUESTÃO 6:

Um Arquiteto de Soluções está projetando uma solução de banco de dados relacional de alta disponibilidade para mitigar o risco de falha multirregional. O banco de dados deve atender a um Objetivo de Ponto de Recuperação (RPO) de 1 segundo e um Objetivo de Tempo de Recuperação (RTO) de menos de 1 minuto. O arquiteto precisa de um plano de recuperação de desastres que permita a replicação automática entre regiões com perda mínima de dados e recuperação rápida em caso de falha.
Qual recurso da AWS atende melhor a esse requisito?

Banco de dados global do Amazon Aurora

A PERGUNTA EXIGE UMA SOLUÇÃO QUE OFEREÇA:
• Alta disponibilidade multirregional
• RPO de 1 segundo
• RTO inferior a 1 minuto
• Replicação automática entre regiões
• Recuperação de desastres com mínima perda de dados

✓ O Amazon Aurora Global Database é:
• Projetado especificamente para replicação entre regiões com latência inferior a 1 segundo
• Capaz de atingir RPO de ~1 segundo e RTO inferior a 1 minuto
• Ideal para recuperação de desastres com failover rápido entre regiões
• Gerenciado pela AWS, com baixa sobrecarga operacional
• Baseado em infraestrutura distribuída e altamente resiliente do Aurora

Explicação Corretas:

O **Amazon Aurora Global Database** permite que um único banco de dados seja usado em várias regiões da AWS, com replicação rápida (latência <1s) sem impacto no desempenho. Oferece **leituras locais com baixa latência**, além de **recuperação de desastres entre regiões**, podendo promover uma região secundária para leitura e gravação em menos de 1 minuto (RPO ~1s, RTO <1min), garantindo continuidade de negócios global.

Explicação Incorretas:

Resumo das Opções de Banco de Dados na AWS

- **Tabela Global do DynamoDB:** Inadequada para bancos de dados relacionais. É otimizada para cargas de trabalho NoSQL e replicação multirregional.
- **Amazon RDS para PostgreSQL com Réplicas de Leitura:** Não atende aos requisitos de RPO (Objetivo de Ponto de Recuperação) e RTO (Objetivo de Tempo de Recuperação). O atraso de replicação é de minutos, não de segundos.
- **Amazon Timestream for Analytics:** Inadequado. É um serviço de banco de dados de séries temporais, voltado para dados de IoT e operacionais.
- **A solução correta para este cenário é:**
- **Amazon Aurora Global Database:** Atende aos requisitos de RPO e RTO de baixa latência, sendo a opção mais adequada para o cenário em questão.

QUESTÃO 7:

Uma empresa de varejo recebe .csv arquivos de dados brutos em seu bucket do Amazon S3 de diversas fontes a cada hora, com um tamanho médio de arquivo de 2 GB. Um processo automatizado deve ser implementado para converter esses .csv arquivos para o formato mais eficiente do Apache Parquet e armazenar os arquivos convertidos em outro bucket do S3. Além disso, o processo de conversão deve ser iniciado automaticamente sempre que um novo arquivo for carregado no bucket do S3.
Qual das seguintes opções deve ser implementada para atender a esses requisitos com a MENOR sobrecarga operacional?

utilize uma tarefa de extração, transformação e carregamento (ETL) do AWS Glue para processar e converter s .csvar-quivos para o formato Apache Parquet e, em seguida, armazenar os arquivos de saída no bucket S3 de destino. Configure uma Notificação de Eventos do S3 para rastrear cada S3 PUTevento e invocar a tarefa ETL no Glue por meio do Amazon SQS.

A PERGUNTA EXIGE UMA SOLUÇÃO QUE:
• Converta arquivos .csv automaticamente em Parquet
• Inicie o processo ao detectar novos uploads no S3
• Tenha baixa sobrecarga operacional
• Armazene os resultados em outro bucket do S3

✓ A combinação S3 + SQS + AWS Glue oferece:•Automação total com eventos do S3 ativando a tarefa ETL via SQS
• O AWS Glue é totalmente gerenciado e especializado em tarefas de ETL com conversão entre formatos como CSV e Parquet
• O formato Apache Parquet é otimizado para análise, reduzindo custo de armazenamento e aumentando performance em consultas
• Baixa complexidade operacional, já que todos os componentes são gerenciados pela AWS e integrados entre si
• Alta escalabilidade para processar arquivos de grande volume (2 GB+)

Explicação Corretas:

O A**WS Glue** é um serviço de **ETL** que facilita a extração, transformação e carregamento de dados entre diferentes repositórios (S3, RDS, Redshift). Ele detecta e mapeia esquemas automaticamente e permite transformar dados conforme necessário.

Já o **Apache Parquet** é um formato de armazenamento em colunas, mais eficiente que o .csv, pois oferece **maior compactação e consultas mais rápidas**. Compatível com Spark, Hive e Hadoop, é muito usado em big data. Com o AWS Glue, é possível converter arquivos .csv para Parquet e armazená-los no S3, otimizando desempenho e acesso.

Explicação Incorretas:

- **Opção 1 (AWS Lambda e AWS Transfer Family):** Incorreta porque uma função **Lambda** tem limites de tempo de execução e memória, tornando-a ineficiente para processar arquivos **CSV** grandes. Além disso, o **AWS Transfer Family** é desnecessário para mover os arquivos.

- **Opção 2 (Apache Spark no EC2 com EventBridge):** Incorreta porque o **Spark no EC2** exige gerenciamento manual (provisionamento, monitoramento), e a combinação com o **EventBridge** e o **Lambda** para acionamento introduz complexidade e pontos de falha desnecessários.
- **Opção 3 (AWS Glue com agendamento):** Incorreta porque, embora o **AWS Glue** seja a ferramenta certa, acionar a tarefa por meio de um agendamento (a cada hora) é ineficiente. A abordagem ideal seria usar um **gatilho de evento do S3** para iniciar o processo de conversão imediatamente após o upload do arquivo, proporcionando maior eficiência.

QUESTÃO 8:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 9:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 10:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 11:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 12:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 13:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 14:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 15:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 16:

Explicação Corretas:

Explicação Incorretas:

QUESTÃO 17:

Explicação Corretas:

Explicação Incorretas: