



Simulado AWS Developer-Associate

Pergunta 1:

Um desenvolvedor está criando uma aplicação que precisa armazenar grandes volumes de dados não estruturados, como arquivos de áudio e vídeo. Esses arquivos precisam ser facilmente acessíveis para leitura e escrita. Qual serviço da AWS é mais adequado para essa aplicação?

- A) **Amazon RDS** – Um banco de dados relacional gerenciado, ideal para dados estruturados e relacionais, mas não adequado para grandes volumes de dados não estruturados.
 - B) **Amazon S3** – Serviço de armazenamento de objetos que oferece alta durabilidade e escalabilidade, perfeito para grandes volumes de dados não estruturados, como imagens, áudio e vídeo.
 - C) **Amazon DynamoDB** – Um banco de dados NoSQL, adequado para dados com chave-valor e documentos, mas não ideal para armazenamento de grandes arquivos não estruturados.
 - D) **Amazon EBS** – Um serviço de armazenamento em blocos, mais indicado para discos de instâncias EC2, mas não para armazenar grandes volumes de dados acessados frequentemente.
-

Pergunta 2:

Uma aplicação está gravando mensagens em um banco de dados DynamoDB. À medida que o tráfego aumenta, o desenvolvedor percebe que algumas requisições estão sendo limitadas (throttled). Qual abordagem pode ser utilizada para evitar esse problema?

- A) **Ativar Auto Scaling no DynamoDB** – O Auto Scaling ajusta automaticamente a capacidade provisionada de leitura e escrita com base na demanda, evitando que as requisições sejam limitadas.
- B) **Utilizar um endpoint VPC para o DynamoDB** – Isso pode aumentar a segurança e performance dentro da VPC, mas não resolve problemas de throttling.
- C) **Aumentar a capacidade provisionada manualmente** – Uma solução válida, mas não automatizada, o que pode ser ineficiente para lidar com picos imprevisíveis.

- D) **Configurar um sistema de cache com o Amazon ElastiCache** – Caching pode reduzir a pressão no banco de dados, mas não resolve diretamente o problema de capacidade provisionada limitada.
-

Pergunta 3:

Um desenvolvedor precisa executar uma função em resposta a eventos, como alterações em um bucket S3 ou novas mensagens em uma fila SQS, sem gerenciar servidores. Qual serviço AWS é mais adequado para esse caso?

- A) **Amazon EC2** – Embora seja flexível, requer o gerenciamento de servidores, o que não atende ao requisito de "sem servidor".
 - B) **AWS Lambda** – Um serviço de computação sem servidor que permite executar código em resposta a eventos gerados por outros serviços AWS, como S3 e SQS.
 - C) **AWS Elastic Beanstalk** – Embora automatize a gestão de infraestrutura, não é um serviço sem servidor e não é ideal para a execução de código em resposta a eventos específicos.
 - D) **Amazon EMR** – Focado em processamento de grandes volumes de dados com Hadoop, não é ideal para processamento de eventos em tempo real.
-

Pergunta 4:

Um desenvolvedor está implementando uma aplicação que requer criptografia de dados sensíveis. Ele deseja centralizar a gestão das chaves de criptografia usadas por vários serviços AWS. Qual serviço AWS é mais apropriado para isso?

- A) **AWS CloudHSM** – Fornece um hardware dedicado para gerenciamento de chaves, mas é mais complexo e caro do que as soluções baseadas em software.
 - B) **Amazon S3** – Oferece criptografia de objetos em repouso, mas não gerencia centralmente as chaves.
 - C) **AWS KMS (Key Management Service)** – Permite criar e gerenciar chaves de criptografia que podem ser usadas em diversos serviços da AWS de forma centralizada e integrada.
 - D) **AWS Secrets Manager** – Ideal para gerenciar credenciais e segredos de forma segura, mas não para chaves de criptografia.
-

Pergunta 5:

Uma empresa está migrando sua aplicação web para AWS e precisa garantir alta disponibilidade e escalabilidade automática. Eles preferem minimizar o gerenciamento de servidores e infraestrutura. Qual serviço da AWS deve ser utilizado?

- A) **Amazon EC2** – Fornece escalabilidade, mas requer o gerenciamento de instâncias e infraestrutura, o que pode não ser ideal para minimizar operações.
 - B) **AWS Lambda** – Permite executar funções em resposta a eventos sem necessidade de gerenciar servidores, mas pode não ser ideal para aplicações que requerem um ambiente de execução persistente.
 - C) **Amazon RDS** – Um banco de dados gerenciado com suporte a alta disponibilidade, mas não resolve o problema de escalabilidade automática da aplicação.
 - D) **AWS Elastic Beanstalk** – Facilita o deploy, gerenciamento e escalabilidade de aplicações sem exigir a gestão direta de servidores.
-

Pergunta 6:

Você está desenvolvendo uma aplicação que precisa gravar mensagens em uma fila para processamento posterior. O aplicativo precisa garantir que cada mensagem seja processada apenas uma vez. Qual serviço AWS deve ser utilizado?

- A) **Amazon SQS (fila padrão)** – Oferece processamento escalável de mensagens, mas não garante que cada mensagem seja entregue apenas uma vez.
 - B) **Amazon SQS (fila FIFO)** – Garante que as mensagens sejam processadas exatamente uma vez e na ordem correta.
 - C) **Amazon SNS** – É um serviço de notificações, ideal para casos de broadcast, mas não é adequado para filas que garantem processamento único.
 - D) **AWS Lambda** – Pode processar mensagens, mas não gerencia filas diretamente, sendo mais útil quando usado com SQS.
-

Pergunta 7:

Um desenvolvedor está construindo uma aplicação que precisa consultar grandes volumes de dados armazenados no Amazon S3 usando SQL. Qual serviço AWS é mais apropriado para essa tarefa?

- A) **Amazon RDS** – Oferece consultas SQL, mas requer que os dados sejam armazenados em um banco de dados relacional, não no S3.

- B) **Amazon Athena** – Permite consultas SQL diretamente em dados armazenados no Amazon S3, sem a necessidade de movê-los para outro banco de dados.
 - C) **Amazon DynamoDB** – Um banco de dados NoSQL que não é adequado para consultas SQL em dados armazenados no S3.
 - D) **Amazon Redshift** – Oferece consultas SQL em grandes volumes de dados, mas exige que os dados sejam carregados para o cluster Redshift, não no S3.
-

Pergunta 8:

Uma aplicação utiliza Amazon S3 para armazenar arquivos sensíveis. O desenvolvedor deseja garantir que esses arquivos sejam criptografados automaticamente ao serem carregados no S3. Qual opção de criptografia deve ser utilizada?

- A) **Criptografia no lado do cliente (CSE)** – Exige que o cliente gerencie a criptografia e as chaves, o que adiciona complexidade.
 - B) **Criptografia no lado do servidor com chaves gerenciadas pelo S3 (SSE-S3)** – O Amazon S3 gerencia automaticamente as chaves de criptografia para criptografar e descriptografar os dados.
 - C) **Criptografia no lado do servidor com chaves gerenciadas pelo KMS (SSE-KMS)** – Oferece maior controle sobre as chaves de criptografia, com integração com o AWS KMS.
 - D) **Criptografia no lado do servidor com chaves fornecidas pelo cliente (SSE-C)** – Requer que o cliente forneça as chaves para criptografar e descriptografar os dados.
-

Pergunta 9:

Um desenvolvedor está criando uma API RESTful que precisa gerenciar a autenticação e autorização de usuários. Qual serviço da AWS pode simplificar a autenticação de usuários em uma API?

- A) **Amazon Cognito** – Facilita a adição de autenticação e autorização a APIs, permitindo que os desenvolvedores integrem usuários com provedores de identidade externos.
- B) **Amazon API Gateway** – Gerencia APIs, mas precisa ser integrado a um serviço como o Cognito para autenticação de usuários.

- C) **AWS IAM** – Gerencia permissões de usuários dentro da AWS, mas não é usado diretamente para autenticar usuários finais em uma API.
 - D) **AWS Secrets Manager** – Gerencia segredos, mas não é utilizado para autenticar usuários diretamente em uma API.
-

Pergunta 10:

Você está criando uma função Lambda que será acionada por eventos em um bucket Amazon S3. A função precisa processar grandes volumes de dados e deve escalar automaticamente conforme necessário. Como o AWS Lambda gerencia a escalabilidade nesse cenário?

- A) **Auto Scaling Lambda** – Não é um conceito do Lambda, já que a escalabilidade do Lambda é gerenciada automaticamente.
- B) **Lambda Provisioned Concurrency** – Define a capacidade mínima de execução da função, mas a escalabilidade em resposta a eventos é gerenciada automaticamente pelo Lambda.
- C) **Lambda Reserved Instances** – Não existe esse conceito no Lambda.
- D) **Lambda Concurrent Executions** – O Lambda escala automaticamente com base no número de eventos, gerenciando execuções concorrentes para atender à demanda.

Sim, é possível chegar a 15 perguntas com respostas mais detalhadas e dentro do limite de caracteres. Vou adicionar mais 5 perguntas ao simulado.

Pergunta 11:

Você está desenvolvendo uma aplicação sem servidor que armazena dados em um bucket Amazon S3 e usa o AWS Lambda para processar esses dados quando novos objetos são adicionados. Qual configuração você deve usar para garantir que os eventos do S3 acionem a função Lambda corretamente?

- A) **Configurar notificações no bucket S3** – O Amazon S3 pode ser configurado para enviar notificações ao Lambda sempre que um novo objeto é adicionado ao bucket.
- B) **Configurar o Lambda para monitorar eventos do CloudWatch** – CloudWatch monitora logs e métricas, mas não aciona diretamente funções Lambda com base em eventos do S3.

- **C) Utilizar AWS Step Functions** – Embora útil para orquestrar fluxos de trabalho complexos, Step Functions não gerencia diretamente eventos do S3.
 - **D) Usar SNS para enviar notificações ao Lambda** – O SNS pode ser usado para notificações, mas não é necessário para integrar diretamente o S3 ao Lambda.
-

Pergunta 12:

Um desenvolvedor precisa criar um pipeline CI/CD para sua aplicação sem servidor. Ele quer garantir que a infraestrutura e o código sejam implementados juntos de forma automatizada. Qual serviço da AWS deve ser utilizado para gerenciar a infraestrutura como código?

- **A) AWS CodePipeline** – Orquestra o fluxo de trabalho de CI/CD, mas não gerencia infraestrutura diretamente.
 - **B) AWS CloudFormation** – Permite gerenciar infraestrutura como código, criando e provisionando recursos AWS automaticamente.
 - **C) AWS CodeBuild** – Automatiza o processo de build e testes, mas não gerencia infraestrutura como código.
 - **D) Amazon S3** – Um serviço de armazenamento de objetos que não gerencia pipelines ou infraestrutura como código.
-

Pergunta 13:

Uma aplicação web construída com AWS Lambda precisa acessar uma tabela DynamoDB para ler e gravar dados. Qual é a maneira mais segura e eficiente de conceder permissão à função Lambda para acessar o DynamoDB?

- **A) Utilizar credenciais hardcoded no código** – Armazenar credenciais no código é uma prática insegura e deve ser evitada.
- **B) Criar uma policy no IAM e atribuí-la à função Lambda** – A maneira mais segura é associar uma policy IAM à função Lambda, concedendo permissões mínimas necessárias para acessar o DynamoDB.
- **C) Configurar um arquivo de configuração no Amazon S3** – Embora o S3 possa armazenar dados, essa não é uma prática recomendada para gerenciar credenciais de acesso.

- **D) Armazenar as credenciais no AWS Secrets Manager** – O Secrets Manager é útil para gerenciar segredos, mas o acesso ao DynamoDB deve ser feito via roles IAM para minimizar a exposição de credenciais.
-

Pergunta 14:

Você está desenvolvendo uma API usando o Amazon API Gateway e o AWS Lambda. À medida que o uso da API aumenta, você percebe que os tempos de resposta estão começando a aumentar. Qual abordagem pode ser usada para melhorar a performance da API?

- **A) Aumentar a memória das funções Lambda** – Isso pode melhorar o desempenho das funções, mas não resolve o problema da latência de rede.
 - **B) Ativar o cache no API Gateway** – Isso pode reduzir o número de chamadas feitas ao Lambda e melhorar significativamente a performance da API ao armazenar em cache as respostas.
 - **C) Migrar para Amazon EC2** – Embora mais flexível, migrar para EC2 aumenta a complexidade de gerenciamento e pode não ser a solução mais eficiente.
 - **D) Aumentar o timeout no API Gateway** – Isso pode reduzir os erros por timeout, mas não melhora a performance da API.
-

Pergunta 15:

Uma aplicação está utilizando AWS Lambda para processar dados de uma fila SQS. À medida que o número de mensagens aumenta, o desenvolvedor precisa garantir que o Lambda consiga processar as mensagens rapidamente, sem falhas. Qual abordagem pode ser usada para garantir que o Lambda escale corretamente com base no número de mensagens?

- **A) Aumentar o número de instâncias EC2** – Não é necessário para Lambda, pois é um serviço sem servidor.
- **B) Configurar a concorrência provisionada no Lambda** – Isso garante que o Lambda tenha a capacidade mínima sempre disponível para lidar com grandes volumes de mensagens.
- **C) Ativar Auto Scaling no SQS** – O SQS não exige Auto Scaling, já que ele automaticamente ajusta a fila conforme a demanda.
- **D) Aumentar a capacidade provisionada no DynamoDB** – Isso pode ser útil em outro contexto, mas não afeta diretamente a escalabilidade do Lambda em relação ao SQS.

Gabarito

- 1 - B
- 2 - A
- 3 - B
- 4 - C
- 5 - D
- 6 - B
- 7 - B
- 8 - B
- 9 - A
- 10 - D
- 11 - A
- 12 - B
- 13 - B
- 14 - B
- 15 - B

Explicação das questões

1- **Amazon S3** é o mais adequado porque é um serviço de armazenamento de objetos que oferece alta durabilidade e escalabilidade, ideal para grandes volumes de dados não estruturados, como áudio e vídeo.

2- **Auto Scaling no DynamoDB** ajusta automaticamente a capacidade de leitura e escrita com base na demanda, evitando throttling. Outras opções não lidam com esse problema de forma tão eficaz.

3- **AWS Lambda** é um serviço sem servidor que executa código em resposta a eventos, como alterações no S3 ou novas mensagens no SQS, sem necessidade de gerenciar servidores.

4- **AWS KMS** permite gerenciar chaves de criptografia centralizadas, oferecendo integração com diversos serviços da AWS para criptografia de dados de forma segura e centralizada.

5- **AWS Elastic Beanstalk** oferece alta disponibilidade e escalabilidade automática, além de minimizar o gerenciamento de servidores, o que é o foco da questão.

6- **Amazon SQS (fila FIFO)** garante que as mensagens sejam processadas exatamente uma vez e na ordem correta, o que atende ao requisito da pergunta.

7- **Amazon Athena** permite consultas SQL diretamente em dados armazenados no S3, sem a necessidade de movê-los para outro banco de dados, tornando-o ideal para essa tarefa.

8- **Criptografia no lado do servidor com chaves gerenciadas pelo S3 (SSE-S3)** permite que o Amazon S3 gerencie automaticamente as chaves de criptografia, simplificando o processo.

9- **Amazon Cognito** facilita a adição de autenticação e autorização a APIs, integrando usuários com provedores de identidade externos e simplificando a gestão de usuários.

10- **Lambda Concurrent Executions** permite que o AWS Lambda escale automaticamente com base no número de eventos, gerenciando execuções simultâneas para lidar com a demanda crescente.

11- **Configurar notificações no bucket S3** garante que o S3 acione a função Lambda automaticamente quando novos objetos forem adicionados ao bucket.

12- **AWS CloudFormation** é a melhor escolha para gerenciar a infraestrutura como código, provisionando recursos AWS automaticamente junto com a aplicação.

13- **Criar uma policy no IAM e atribuí-la à função Lambda** é a maneira mais segura e eficiente de conceder permissão à função Lambda, evitando o uso de credenciais hardcoded.

14- **Ativar o cache no API Gateway** reduz o número de chamadas ao Lambda, melhorando a performance ao armazenar respostas em cache e diminuir a latência.

15- **Configurar a concorrência provisionada no Lambda** garante que a função Lambda tenha capacidade mínima disponível para processar grandes volumes de mensagens de forma rápida e eficiente.