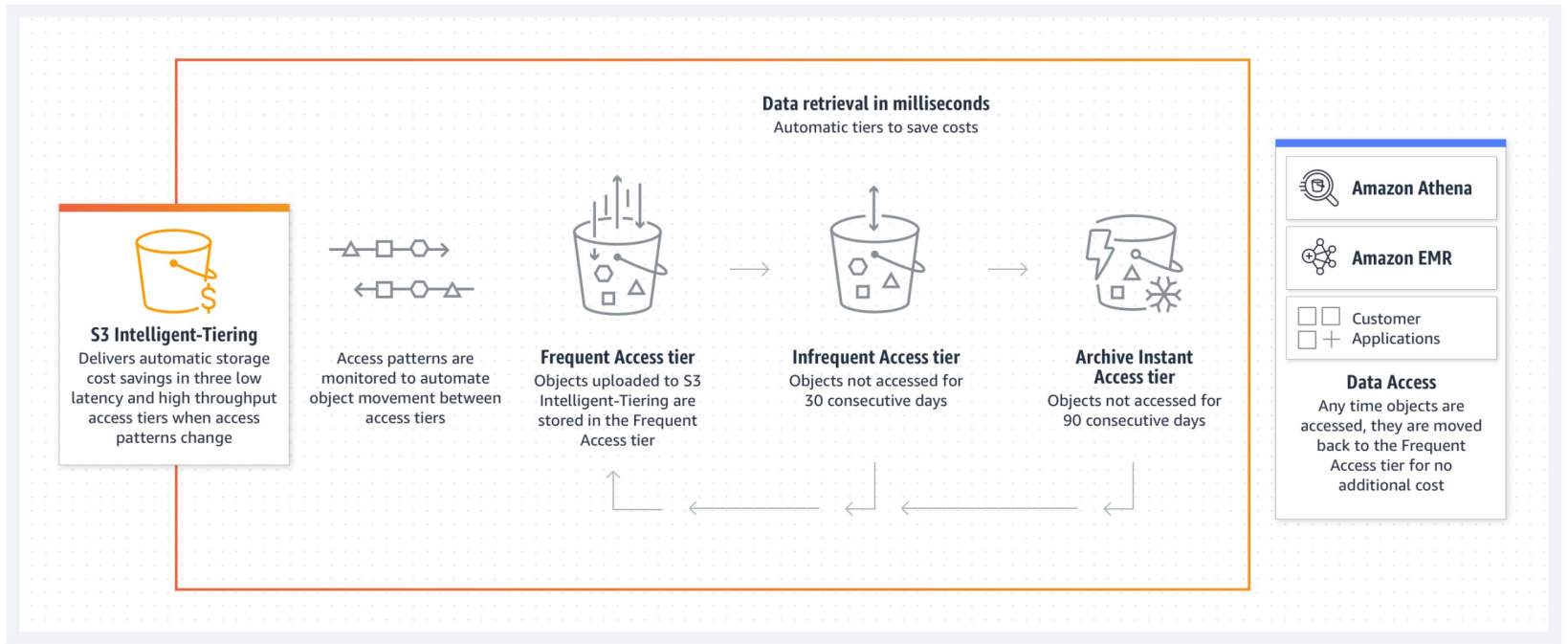
Etapa 7

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html

Feature	Cluster Placement Group	Spread Placement Group	Partition Placement Group
Description	Agrupa instâncias próximas umas das outras para minimizar a latência de rede e maximizar a largura de banda.	Distribui instâncias em hardware subjacente distinto para reduzir o risco de falhas simultâneas.	Divide instâncias em grupos lógicos (partições) que são isolados uns dos outros no hardware para reduzir o impacto de falhas de hardware.
Use Case	Computação de alto desempenho (HPC), requisitos de baixa latência, aplicações com nós fortemente acoplados.	Aplicações que exigem alta disponibilidade e podem tolerar menor densidade de instâncias, por exemplo, cargas de trabalho críticas onde o isolamento de instâncias é essencial.	Grandes cargas de trabalho distribuídas e replicadas, como big data, clusters Hadoop ou bancos de dados distribuídos.
Instance Spread	As instâncias são colocadas dentro de uma única Zona de Disponibilidade (AZ).	As instâncias podem ser distribuídas por várias AZs.	As instâncias são distribuídas por partições dentro de uma única AZ, mas cada partição é isolada das outras.
Maximum Instances	Normalmente suporta até 500 instâncias por grupo, mas isso pode variar.	Um máximo de 7 instâncias por AZ por grupo.	Até 7 partições por grupo, com cada partição contendo várias instâncias.


Data retrieval in milliseconds
Automatic tiers to save costs
S3 Intelligent-Tiering
Delivers automatic storage cost savings in three low latency and high throughput access tiers when access patterns change
Access patterns are monitored to automate object movement between access tiers
Frequent Access tier
Objects uploaded to S3 Intelligent-Tiering are stored in the Frequent Access tier
Infrequent Access tier
Objects not accessed for 30 consecutive days
Archive Instant Access tier
Objects not accessed for 90 consecutive days
Amazon Athena
Amazon EMR
Customer Applications
Data Access
Any time objects are accessed, they are moved back to the Frequent Access tier for no additional cost

Prova são 65 questões em 130 minutos -> 2 min por questão na média

Questão 1

Cenário:

Você está trabalhando como Arquiteto de Soluções para uma empresa que lida com dados sensíveis de clientes. Sua empresa armazena grandes quantidades desses dados em buckets do Amazon S3. Devido a exigências rigorosas de conformidade, todos os dados armazenados no S3 devem ser criptografados em repouso usando criptografia do lado do servidor com chaves gerenciadas pelo AWS Key Management Service (KMS).
Para garantir que todos os objetos enviados ao bucket S3 sejam criptografados usando a criptografia do lado do servidor com uma chave KMS específica, você decide impor isso exigindo o uso dos cabeçalhos HTTP apropriados nas solicitações de upload.

Questão:

Quais dos seguintes cabeçalhos HTTP devem ser incluídos na solicitação PUT para garantir que todos os objetos enviados ao seu bucket S3 sejam criptografados com a chave KMS especificada?

A. x-amz-server-side-encryption: AES256
B. x-amz-server-side-encryption: aws:kms
C. x-amz-server-side-encryption-aws-kms-key-id: <Seu-KMS-Key-ID>
D. Ambos B e C

Resposta Correta: D. Ambos B e C

Explicação:

•**A.** x-amz-server-side-encryption: AES256 - Este cabeçalho especifica que o Amazon S3 deve usar criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3), e não pelo AWS KMS. Portanto, esta não é a resposta correta para impor a criptografia com KMS.

•**B.** x-amz-server-side-encryption: aws:kms - Este cabeçalho especifica que o Amazon S3 deve usar criptografia do lado do servidor com chaves gerenciadas pelo AWS KMS (SSE-KMS). Este cabeçalho por si só indica que o KMS deve ser usado, mas você também precisa especificar a chave KMS exata que deve ser utilizada.

•**C.** x-amz-server-side-encryption-aws-kms-key-id: <Seu-KMS-Key-ID> - Este cabeçalho especifica o ID da chave AWS KMS que deve ser usada para criptografia. No entanto, sem especificar que você deseja usar o KMS (opção B), este cabeçalho sozinho não seria suficiente.

•**D.** Ambos B e C - Para garantir que os objetos sejam criptografados usando a chave AWS KMS especificada, ambos os cabeçalhos devem ser incluídos na solicitação. x-amz-server-side-encryption: aws:kms garante o uso do KMS, e x-amz-server-side-encryption-aws-kms-key-id: <Seu-KMS-Key-ID> especifica qual chave KMS deve ser usada. Portanto, esta é a resposta correta.

If you require your data uploads to be encrypted using only Amazon S3 managed keys, you can use the following bucket policy. For example, the following bucket policy denies permissions to upload an object unless the request includes the `x-amz-server-side-encryption` header to request server-side encryption:

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
  ]
}
```

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Note

When you upload an object, you can specify the KMS key by using the `x-amz-server-side-encryption-aws-kms-key-id` header or rely on your default bucket encryption configuration. If your PutObject request specifies `aws:kms` in the `x-amz-server-side-encryption` header, but does not specify the `x-amz-server-side-encryption-aws-kms-key-id` header, then Amazon S3 assumes that you want to use the AWS managed key. Regardless, the AWS KMS key ID that Amazon S3 uses for object encryption must match the AWS KMS key ID in the policy, otherwise Amazon S3 denies the request.

Questão 2

Cenário:
Você está projetando uma solução de armazenamento para uma empresa que armazena grandes arquivos de vídeo no Amazon S3. Alguns desses arquivos têm vários gigabytes de tamanho, e você deseja garantir que eles possam ser carregados de forma eficiente e resiliente, mesmo com possíveis falhas de rede.
Para otimizar o processo de upload desses arquivos grandes, você decide usar o upload multipart do S3, que divide o arquivo em partes menores e as carrega separadamente.

Questão:
Qual das seguintes afirmações sobre o processo de upload multipart no Amazon S3 é **verdadeira**?

A. Se uma parte do upload multipart falhar, todo o upload será cancelado automaticamente e você precisará reiniciar o processo do zero.

B. Você pode reiniciar o upload multipart de uma parte específica sem afetar as outras partes que já foram carregadas com sucesso.

C. O upload multipart é obrigatório para arquivos maiores que 5 GB e recomendado para arquivos maiores que 100 MB.

D. O upload multipart só pode ser usado com arquivos de até 5 TB, e todas as partes devem ser do mesmo tamanho.

Resposta Correta:

B. Você pode reiniciar o upload multipart de uma parte específica sem afetar as outras partes que já foram carregadas com sucesso.

Explicação:

•**A.** Se uma parte do upload multipart falhar, o upload **não** será cancelado automaticamente. Você pode tentar reenviar apenas a parte que falhou, sem precisar reiniciar todo o processo.

•**B.** Esta afirmação é verdadeira. O upload multipart permite que você reinicie o upload de uma parte específica se ela falhar, sem afetar as outras partes que já foram carregadas com sucesso. Isso torna o processo de upload mais resiliente a falhas.

•**C.** O upload multipart **não** é obrigatório para arquivos maiores que 5 GB, mas é altamente recomendado para arquivos maiores que 100 MB para melhorar a eficiência e resiliência.

•**D.** Embora o upload multipart possa ser usado com arquivos de até 5 TB, as partes **não** precisam ter o mesmo tamanho. A única exceção é que a última parte pode ser menor.
Esta questão aborda a compreensão das características e vantagens do upload multipart do S3, que é uma prática recomendada ao lidar com arquivos grandes na nuvem.

Carregar e copiar objetos usando multipart upload

O multipart upload permite que você faça upload de um único objeto como um conjunto de partes. Cada parte é uma parte contígua de dados do objeto. O upload dessas partes de objetos pode ser feito de maneira independente e em qualquer ordem. Se a transmissão de alguma parte falhar, você poderá retransmitir essa parte sem afetar outras partes. Depois que todas as partes do objeto forem carregadas, o Amazon S3 montará essas partes e criará o objeto. Geralmente, quando seu objeto alcança 100 MB de tamanho, você deve considerar o uso de multipart uploads em vez de fazer upload do objeto em uma única operação.

Usar o multipart upload fornece as seguintes vantagens:

- **Improved throughput** (Throughput aprimorada): você pode carregar as partes em paralelo para melhorar o throughput.
- **Quick recovery from any network issues** (Recuperação rápida de qualquer problema de rede): o tamanho menor das partes minimiza o impacto de reiniciar um carregamento que falhou devido a um erro de rede.
- **Pause and resume object uploads** (Pausar e retomar carregamentos de objetos): você pode carregar as partes do objeto ao longo do tempo. Após ser iniciado um carregamento fracionado, ele não expira; você deve concluir ou interromper explicitamente o carregamento fracionado.
- **Begin an upload before you know the final object size** (Começar um carregamento antes de saber o tamanho final do objeto): você pode carregar um objeto à medida que ele é criado.

Questão 3

Cenário:

Você está projetando a arquitetura de um sistema de e-commerce que precisa armazenar e consultar dados de produtos, pedidos e clientes. O sistema exige baixa latência para leituras e escritas, alta escalabilidade para lidar com picos de tráfego, e precisa ser altamente disponível em várias regiões. Além disso, o sistema deve ser capaz de suportar consultas complexas, como junções entre tabelas.
Você está considerando o uso do Amazon DynamoDB, mas está comparando com outras opções de banco de dados, como o Amazon RDS (que pode executar MySQL ou PostgreSQL) e o Amazon Aurora.

Questão:

Qual das seguintes afirmações é **correta** ao comparar o Amazon DynamoDB com outras opções de banco de dados, como Amazon RDS ou Amazon Aurora?

A. O DynamoDB é ideal para cenários que exigem consultas SQL complexas com junções entre várias tabelas.

B. O DynamoDB oferece escalabilidade automática sem a necessidade de gerenciar instâncias subjacentes, enquanto o Amazon RDS requer ajustes manuais de capacidade.

C. O DynamoDB armazena dados em um formato relacional e suporta completamente as mesmas funcionalidades que bancos de dados como MySQL ou PostgreSQL.

D. O Amazon RDS é a escolha ideal para aplicações que exigem baixa latência de leitura/escrita e escalabilidade horizontal automática, como o DynamoDB.

Resposta Correta:

B. O DynamoDB oferece escalabilidade automática sem a necessidade de gerenciar instâncias subjacentes, enquanto o Amazon RDS requer ajustes manuais de capacidade.

Explicação:

•**A.** O DynamoDB não é adequado para consultas SQL complexas com junções entre várias tabelas, pois é um banco de dados NoSQL que funciona melhor com consultas simples baseadas em chave.

•**B.** Esta afirmação é verdadeira. O DynamoDB foi projetado para ser altamente escalável, oferecendo escalabilidade automática de leitura/escrita sem a necessidade de gerenciar a infraestrutura subjacente. Já o Amazon RDS, embora possa escalar, geralmente requer ajustes manuais na capacidade das instâncias.

•**C.** O DynamoDB não armazena dados em um formato relacional e, portanto, não suporta todas as funcionalidades de bancos de dados relacionais como MySQL ou PostgreSQL. Ele é um banco de dados NoSQL que usa armazenamento baseado em chave-valor e documentos.

•**D.** Embora o Amazon RDS possa ser configurado para baixa latência, ele não suporta escalabilidade horizontal automática da mesma maneira que o DynamoDB, que foi projetado para escalar horizontalmente automaticamente. Esta questão ajuda a avaliar o conhecimento sobre a escolha correta de banco de dados com base em requisitos específicos do sistema, como escalabilidade, complexidade das consultas e gerenciamento de infraestrutura.

Algumas considerações

O Amazon RDS agora oferece suporte ao Storage Auto Scaling

Publicado: Jun 20, 2019

A partir de hoje, o Amazon RDS for MariaDB, o Amazon RDS for MySQL, o Amazon RDS for PostgreSQL, o Amazon RDS for SQL Server e o Amazon RDS for Oracle oferecem suporte ao RDS Storage Auto Scaling. O RDS Storage Auto Scaling dimensiona automaticamente a capacidade de armazenamento em resposta às crescentes cargas de trabalho do banco de dados, sem tempo de inatividade.

Anteriormente, você precisava provisionar manualmente a capacidade de armazenamento de acordo com a previsão das demandas dos aplicativos. O provisionamento insuficiente poderia resultar em inatividade de aplicativos, e o provisionamento em excesso poderia resultar em recursos subutilizados e custos maiores. Com o RDS Storage Auto Scaling, basta definir o limite máximo de armazenamento desejado e o Auto Scaling cuida do resto.

O RDS Storage Auto Scaling monitora continuamente o consumo real de armazenamento e dimensiona automaticamente a capacidade quando a utilização real se aproxima da capacidade de armazenamento provisionada. O Auto Scaling funciona com instâncias de banco de dados novas e existentes. Você pode ativer o Auto Scaling com apenas alguns cliques no Console de Gerenciamento da AWS. Não há custo adicional para o RDS Storage Auto Scaling. Você paga apenas pelos recursos do RDS de que precisa para executar seus aplicativos. Para saber mais, acesse a página de documentação sobre o armazenamento para o Amazon RDS.

O RDS Storage Auto Scaling está disponível em todas as regiões comerciais da AWS, exceto Ásia-Pacífico (Hong Kong) e AWS GovCloud.

Questão 4

Cenário:

Você está desenvolvendo a arquitetura de um sistema de gerenciamento de conteúdo (CMS) para uma grande empresa de mídia. Este sistema será utilizado para armazenar e gerenciar uma vasta quantidade de artigos, imagens, vídeos e metadados associados. A empresa precisa de um banco de dados que suporte transações complexas, consultas SQL avançadas, e que possa lidar com uma carga de trabalho variável ao longo do dia, com picos significativos durante as manhãs e noites.
Você está considerando o uso do Amazon RDS (executando MySQL ou PostgreSQL) e comparando-o com outras opções, como o Amazon DynamoDB e o Amazon Aurora.

Questão:

Qual das seguintes afirmações é **correta** ao comparar o Amazon RDS com outras opções de banco de dados, como o Amazon DynamoDB ou o Amazon Aurora?

A. O Amazon RDS é ideal para workloads que exigem alta escalabilidade horizontal e latência de leitura extremamente baixa, como no caso de aplicativos com milhões de solicitações por segundo.
B. O Amazon RDS oferece suporte nativo para consultas SQL complexas, transações ACID e integrações com ferramentas de BI, tornando-o uma escolha adequada para bancos de dados relacionais tradicionais.
C. O Amazon DynamoDB é mais adequado do que o Amazon RDS para workloads que exigem consultas SQL avançadas e processamento transacional complexo.
D. O Amazon Aurora sempre oferece melhor desempenho e menor custo do que o Amazon RDS, independentemente do tipo de carga de trabalho e do volume de dados.

Resposta Correta:

B. O Amazon RDS oferece suporte nativo para consultas SQL complexas, transações ACID e integrações com ferramentas de BI, tornando-o uma escolha adequada para bancos de dados relacionais tradicionais.

Explicação:

•**A.** O Amazon RDS não é a melhor opção para alta escalabilidade horizontal e latência de leitura extremamente baixa. Para esses casos, serviços como o Amazon DynamoDB ou soluções baseadas em NoSQL podem ser mais adequados.

•**B.** Esta afirmação é verdadeira. O Amazon RDS foi projetado para suportar bancos de dados relacionais tradicionais, com suporte completo para consultas SQL complexas, transações ACID, e fácil integração com ferramentas de Business Intelligence (BI). Isso o torna ideal para sistemas que exigem consistência e transações complexas, como um CMS.

•**C.** O Amazon DynamoDB é um banco de dados NoSQL que é ideal para alta escalabilidade e desempenho rápido, mas não suporta consultas SQL avançadas ou processamento transacional complexo da mesma forma que o Amazon RDS ou Aurora.

•**D.** Embora o Amazon Aurora ofereça melhorias significativas de desempenho e pode ser mais econômico para cargas de trabalho específicas, ele não é sempre a melhor ou mais barata opção em todos os cenários. O custo e o desempenho dependem do tipo de carga de trabalho e do volume de dados.
Esta questão avalia o entendimento sobre as diferentes características e casos de uso para Amazon RDS, DynamoDB e Aurora, ajudando a escolher a solução de banco de dados mais adequada para cenários específicos.

https://docs.aws.amazon.com/pt_br/whitepapers/latest/using-power-bi-with-aws-cloud/appendix-microsoft-power-bi-supported-aws-data-sources.html

Questão 5

Cenário:

Você está trabalhando como Arquiteto de Soluções para uma empresa que opera uma plataforma de streaming de vídeo. A plataforma experimenta um tráfego variável, com picos previsíveis durante o lançamento de novos episódios de séries populares e eventos ao vivo. Para otimizar os custos e garantir que a plataforma possa lidar com esses picos de tráfego, você precisa escolher o modelo de precificação do Amazon EC2 mais adequado para diferentes partes da aplicação. Algumas partes do sistema requerem alta disponibilidade, enquanto outras podem tolerar interrupções.

Questão:

Qual dos seguintes modelos de precificação do Amazon EC2 seria o mais econômico e adequado para o cenário descrito?

A. Instâncias On-Demand para todos os componentes da aplicação, garantindo disponibilidade e flexibilidade.

B. Instâncias Reservadas para os servidores de streaming que precisam estar disponíveis 24/7, e Instâncias Spot para tarefas de transcodificação que podem tolerar interrupções.

C. Instâncias Dedicadas para todos os componentes, para garantir isolamento físico, independentemente do custo.

D. Instâncias Spot para todos os componentes da aplicação, visando minimizar os custos, desde que a plataforma possa lidar com possíveis interrupções.

Resposta Correta:

B. Instâncias Reservadas para os servidores de streaming que precisam estar disponíveis 24/7, e Instâncias Spot para tarefas de transcodificação que podem tolerar interrupções.

Explicação:

•**A.** As Instâncias On-Demand oferecem flexibilidade e são adequadas para workloads variáveis, mas não são a opção mais econômica para uso previsível e de longo prazo, onde as Instâncias Reservadas poderiam proporcionar economias significativas.

•**B.** Esta é a resposta correta. As Instâncias Reservadas são ideais para workloads que exigem alta disponibilidade, como os servidores de streaming que precisam estar operacionais 24/7. As Instâncias Spot são econômicas para workloads que podem tolerar interrupções, como tarefas de transcodificação.

•**C.** Instâncias Dedicadas são usadas principalmente quando há necessidade de isolamento físico por razões de conformidade, o que não é necessário neste cenário e resultaria em custos desnecessários.

•**D.** Embora as Instâncias Spot ofereçam o menor custo, elas vêm com o risco de interrupções, o que as torna inadequadas para componentes que exigem alta disponibilidade. Esta opção não seria adequada para partes críticas da aplicação.
Esta questão avalia a capacidade de escolher o modelo de precificação do EC2 correto, com base nas necessidades específicas de diferentes componentes de um sistema, equilibrando custo e disponibilidade.

Questão 6

Cenário:

Você está projetando a arquitetura de uma aplicação web que será implantada em uma infraestrutura na AWS. A aplicação possui uma API que recebe um alto volume de solicitações HTTPS e precisa de um balanceador de carga que suporte a terminação SSL/TLS. Além disso, a aplicação precisa de um balanceador de carga para distribuir o tráfego de rede entre servidores que se comunicam usando o protocolo TCP, garantindo baixa latência para conexões em tempo real.
Você está avaliando o uso de diferentes tipos de balanceadores de carga disponíveis na AWS: Application Load Balancer (ALB), Network Load Balancer (NLB) e Classic Load Balancer (CLB).

Questão:

Qual das seguintes afirmações é **correta** ao comparar os tipos de balanceadores de carga na AWS?

A. O Application Load Balancer (ALB) é mais adequado para roteamento de solicitações HTTP/HTTPS baseadas em conteúdo e suporta terminação SSL/TLS nativa.
B. O Network Load Balancer (NLB) é mais adequado para aplicações baseadas em HTTP que exigem roteamento inteligente de solicitações baseado em caminhos ou cabeçalhos.
C. O Classic Load Balancer (CLB) é a melhor escolha para aplicações que exigem roteamento de tráfego em camadas de aplicação (camada 7) com suporte avançado para regras de roteamento.
D. O Network Load Balancer (NLB) é a escolha ideal para distribuir tráfego TCP com baixa latência e alta performance, mas não suporta endereços IP estáticos.

Resposta Correta:

A. O Application Load Balancer (ALB) é mais adequado para roteamento de solicitações HTTP/HTTPS baseadas em conteúdo e suporta terminação SSL/TLS nativa.

Explicação:

•**A.** Esta afirmação é verdadeira. O ALB é projetado para roteamento de tráfego HTTP/HTTPS com base em regras específicas de conteúdo, como caminhos de URL ou cabeçalhos, e também suporta terminação SSL/TLS, sendo ideal para aplicações web.

•**B.** O NLB não é utilizado para roteamento inteligente de solicitações HTTP, mas sim para distribuir tráfego TCP e UDP, proporcionando baixa latência e alta performance.

•**C.** O CLB oferece capacidades básicas tanto para roteamento na camada de transporte (camada 4) quanto na camada de aplicação (camada 7), mas não possui o suporte avançado de roteamento oferecido pelo ALB.

•**D.** Embora o NLB seja ideal para tráfego TCP com baixa latência, ele **suporta** endereços IP estáticos, ao contrário do que foi afirmado na opção.

Questão 7

Você foi designado para configurar o acesso seguro a um bucket do Amazon S3 para uma aplicação web. A aplicação precisa acessar objetos no bucket e permitir que usuários autenticados enviem novos arquivos, mas deve garantir que nenhum outro tipo de operação seja permitida. Além disso, um serviço de análise de logs da empresa precisa acessar o bucket para ler os arquivos armazenados. Qual das políticas a seguir atende a esses requisitos?

a)
{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:*", "Resource": "arn:aws:s3:::meu-bucket/*" }] }

b)
{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:GetObject", "s3:PutObject"], "Resource": "arn:aws:s3:::meu-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::meu-bucket" }] }

c)
{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:PutObject", "Resource": "arn:aws:s3:::meu-bucket/*" }, { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::meu-bucket/*", "Condition": { "StringEquals": { "aws:username": "servico-analise-logs" } } }] }

d)
{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:GetObject", "s3:PutObject"], "Resource": "arn:aws:s3:::meu-bucket/*" }, { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::meu-bucket/*", "Condition": { "StringEquals": { "aws:username": "servico-analise-logs" } } }] }

Resposta Correta: d)

Explicação: A opção **d)** atende a todos os requisitos. Ela permite que a aplicação web faça operações de GetObject e PutObject no bucket, mas limita a leitura de arquivos pelo serviço de análise de logs com a condição StringEquals para o usuário específico. A opção **a)** concede permissões excessivas (s3:*), enquanto a opção **b)** permite o acesso a ListBucket, o que não é necessário. A opção **c)**, embora próxima, não concede acesso GetObject para a aplicação web, o que é essencial.

Questão 8

Pergunta:

Você está projetando uma arquitetura na AWS que utiliza instâncias EC2 para hospedar uma aplicação web de alto desempenho. A aplicação precisa de armazenamento temporário de alta velocidade para processamento de dados, mas também requer persistência de dados para o sistema operacional e arquivos de configuração. Qual das seguintes combinações de volumes e tipos de armazenamento seria a mais adequada para essa situação, considerando a volatilidade dos dados?

a) Instância EC2 com um volume de armazenamento temporário Instance Store para o sistema operacional e um volume EBS adicional para o armazenamento temporário dos dados de processamento.

b) Instância EC2 com um volume de armazenamento temporário Instance Store para o armazenamento temporário dos dados de processamento e um volume EBS para o sistema operacional.

c) Instância EC2 com volumes EBS tanto para o sistema operacional quanto para o armazenamento temporário dos dados de processamento.

d) Instância EC2 com volumes Instance Store tanto para o sistema operacional quanto para o armazenamento temporário dos dados de processamento.

Resposta Correta: b)

Explicação: A opção **b)** é a mais adequada, pois utiliza o volume de Instance Store, que oferece alta velocidade, para o armazenamento temporário dos dados de processamento, que podem ser descartados após o uso. Para o sistema operacional e arquivos de configuração, um volume EBS é usado, garantindo que os dados persistam mesmo se a instância for interrompida ou terminada. A opção **a)** não é recomendada porque o sistema operacional seria armazenado em um Instance Store, que é volátil e perderia todos os dados em caso de reinicialização da instância. A opção **c)** utiliza volumes EBS para tudo, o que é persistente, mas não aproveita o alto desempenho dos Instance Store. A opção **d)** é inadequada porque ambos os volumes são voláteis, o que pode resultar em perda de dados importantes.

Questão 9

Você está desenvolvendo uma aplicação de comércio eletrônico na AWS que processa pedidos de clientes. A arquitetura envolve vários microserviços: um para receber pedidos, outro para processar pagamentos e outro para gerenciar o estoque. É crucial que os pedidos sejam processados na ordem em que foram recebidos, mas as notificações sobre os status dos pedidos devem ser enviadas para vários sistemas, como para o departamento de logística e para o sistema de faturamento. Qual das seguintes arquiteturas melhor atende a esses requisitos, utilizando SQS e SNS?

a) Usar uma fila SQS FIFO para enfileirar os pedidos, garantindo a ordem de processamento, e SNS para enviar notificações sobre o status dos pedidos para múltiplos sistemas através do padrão fan-out.
b) Usar uma fila SQS Standard para enfileirar os pedidos, permitindo alto throughput e SNS para enviar mensagens diretamente ao sistema de pagamento e ao sistema de estoque, garantindo entrega simultânea.
c) Usar SNS para distribuir os pedidos para múltiplas filas SQS FIFO, garantindo que cada microserviço receba os pedidos na ordem correta, e SQS Standard para enviar notificações de status para os sistemas de logística e faturamento.
d) Usar SNS para enviar mensagens diretamente aos microserviços, permitindo que cada um processe os pedidos de forma assíncrona e SQS FIFO para enfileirar as notificações de status dos pedidos, garantindo que sejam entregues na ordem correta.

Resposta Correta: a)

Explicação: A opção **a)** é a mais adequada para este cenário. A fila SQS FIFO garante que os pedidos sejam processados na ordem exata em que foram recebidos, o que é crucial para a consistência da aplicação de comércio eletrônico. O SNS é utilizado para o padrão de fan-out, enviando notificações sobre o status dos pedidos para múltiplos sistemas, como logística e faturamento, garantindo que todos os sistemas relevantes sejam notificados ao mesmo tempo. A opção **b)** não garante a ordem de processamento dos pedidos, o que pode causar inconsistências. A opção **c)** adiciona complexidade desnecessária ao usar múltiplas filas SQS FIFO, enquanto a opção **d)** não é ideal, pois SNS não garante a ordem de entrega, o que é crucial para o processamento de pedidos nesta arquitetura.

Questão 10

Uma empresa de tecnologia está migrando sua infraestrutura para a AWS e deseja implementar uma política rigorosa de auditoria e conformidade. A equipe de segurança precisa monitorar todas as chamadas de API feitas em suas contas da AWS para detectar atividades suspeitas, rastrear mudanças em recursos críticos e garantir que as práticas recomendadas de segurança sejam seguidas. Qual das seguintes configurações atenderia melhor a esses requisitos utilizando AWS CloudTrail?

a) Ativar o AWS CloudTrail em todas as regiões e configurar o armazenamento dos logs em um bucket do Amazon S3 com políticas de acesso restrito, permitindo que a equipe de segurança audite todas as chamadas de API realizadas na conta.

b) Ativar o AWS CloudTrail apenas na região principal onde os recursos estão implantados e configurar a integração com o AWS Config para rastrear mudanças em recursos específicos, como instâncias EC2 e grupos de segurança.

c) Configurar múltiplas trilhas (trails) no AWS CloudTrail para cada região, garantindo que todas as chamadas de API, inclusive as de contas de IAM, sejam monitoradas e enviadas para um bucket do S3 centralizado.

d) Ativar o AWS CloudTrail e configurar a opção de eventos de gerenciamento, mas excluir eventos de leitura de dados para reduzir custos e focar apenas em eventos de gravação que alteram recursos.

Resposta Correta: a)

Explicação: A opção **a)** é a mais adequada, pois ativar o AWS CloudTrail em todas as regiões e armazenar os logs em um bucket do S3 com políticas de acesso restrito garante que todas as chamadas de API sejam monitoradas, proporcionando visibilidade completa sobre as atividades em toda a conta da AWS. Isso é essencial para a auditoria e conformidade. A opção **b)** limita a visibilidade apenas à região principal e não cobre todas as atividades em outras regiões. A opção **c)**, embora forneça monitoramento detalhado por região, pode adicionar complexidade desnecessária. A opção **d)** exclui eventos de leitura de dados, o que pode omitir informações críticas para a auditoria e detecção de atividades suspeitas.