Semana 5

Guia do Exame

Declaração de tarefa 2.2: Projetar arquiteturas altamente disponíveis e/ou tolerantes a falhas.

Conhecimento sobre:

- Infraestrutura global da AWS (por exemplo, Zonas de Disponibilidade, Regiões AWS, Amazon Route 53).
- AWS Managed Services com casos de uso apropriados (por exemplo, Amazon Comprehend, Amazon Polly).
- Conceitos básicos de redes (por exemplo, tabelas de rotas).
- Estratégias de recuperação de desastres (DR) (por exemplo, backup e restauração, luz piloto, warm standby, failover ativo-ativo, objetivo de ponto de recuperação [RPO], objetivo de tempo de recuperação [RTO]).
- Padrões de design distribuídos.
- Estratégias de failover.
- Infraestrutura imutável.
- Conceitos de balanceamento de carga (por exemplo, Application Load Balancer).
- Conceitos de proxy (por exemplo, Proxy do Amazon RDS).
- Cotas de serviço e limitação de largura de banda (por exemplo, como configurar as cotas de serviço para uma carga de trabalho em um ambiente de standby).

Guia do Exame

- Opções e características de armazenamento (por exemplo, durabilidade, replicação).
- Visibilidade da carga de trabalho (por exemplo, AWS X-Ray).

Habilidades em:

- Determinar estratégias de automação para garantir a integridade da infraestrutura.
- Determinar os serviços da AWS necessários para fornecer uma arquitetura altamente disponível e/ou tolerante a falhas nas Zonas de Disponibilidade ou Regiões AWS.
- Identificar métricas com base nos requisitos empresariais para oferecer uma solução altamente disponível.
- Implementar designs para mitigar pontos únicos de falha.
- Implementar estratégias para garantir a durabilidade e a disponibilidade dos dados (por exemplo, backups).
- Selecionar uma estratégia de DR apropriada para atender aos requisitos empresariais.
- Usar serviços da AWS que melhoram a confiabilidade de aplicações legados e aplicações que não foram criadas para a nuvem (por exemplo, quando não é possível fazer alterações nas aplicações).
- Usar serviços da AWS com propósito específico para cargas de trabalho.

Route 53

O que é o Amazon Route 53?

PDF | RSS

O Amazon Route 53 é um serviço web de Sistema de Nomes de Domínio (DNS) altamente disponível e escalável. Você pode usar o Route 53 para realizar três funções principais em qualquer combinação: registro de domínio, DNS roteamento e verificação de integridade.

O que é o DNS?

O DNS (Domain Name System – Sistema de nome de domínio) converte nomes de domínio legíveis por humanos (por exemplo, www.amazon.com) em endereços IP legíveis por máquina (por exemplo, 192.0.2.44).

Route 53

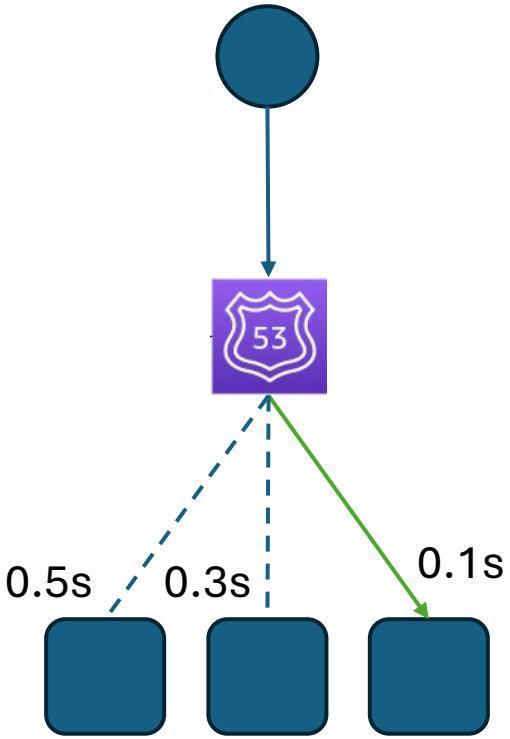
política de roteamento

Uma configuração para registros que determina como o Route 53 responde às DNS consultas. O Route 53 oferece suporte às seguintes políticas de roteamento:

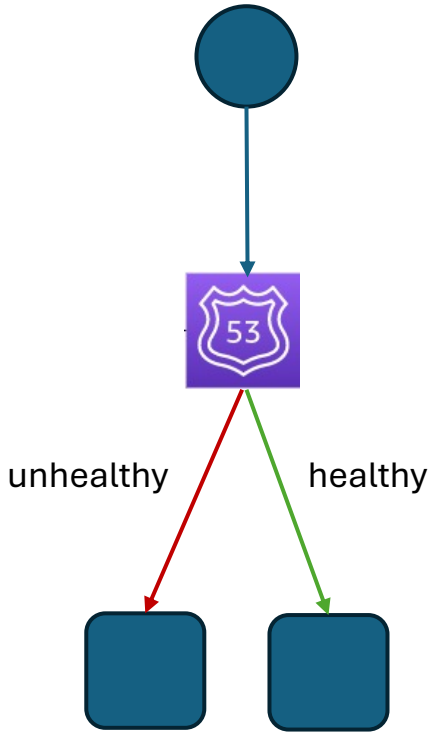
- **Simple routing policy** (Política de roteamento simples): use para encaminhar o tráfego da Internet para um único recurso que executa uma determinada função para seu domínio, por exemplo, um servidor Web que fornece conteúdo para o site example.com.
- **Failover routing policy** (Política de roteamento de failover): use quando quiser configurar o failover ativo-passivo.
- **Geolocation routing policy** (Política de roteamento de localização geográfica): use quando quiser encaminhar o tráfego da Internet para seus recursos com base na localização dos usuários.
- **Geoproximity routing policy** (Política de roteamento de proximidade geográfica): use quando quiser encaminhar o tráfego com base no local de seus recursos e, opcionalmente, alternar o tráfego de recursos em um local para recursos em outro local.
- **Latency routing policy** (Política de roteamento de latência): use quando você tiver recursos em vários locais e quiser encaminhar o tráfego para o recurso que fornece a melhor latência.
- **IP-based routing policy** (Política de roteamento baseado em IP): use quando quiser rotear o tráfego com base no local dos usuários e tiver os endereços IP de origem do tráfego.
- **Política de roteamento de respostas de vários** valores — Use quando quiser que o Route 53 responda a DNS consultas com até oito registros íntegros selecionados aleatoriamente.
- **Weighted routing policy** (Política de roteamento ponderado): use para encaminhar o tráfego para vários recursos nas proporções que você especificar.

Route 53

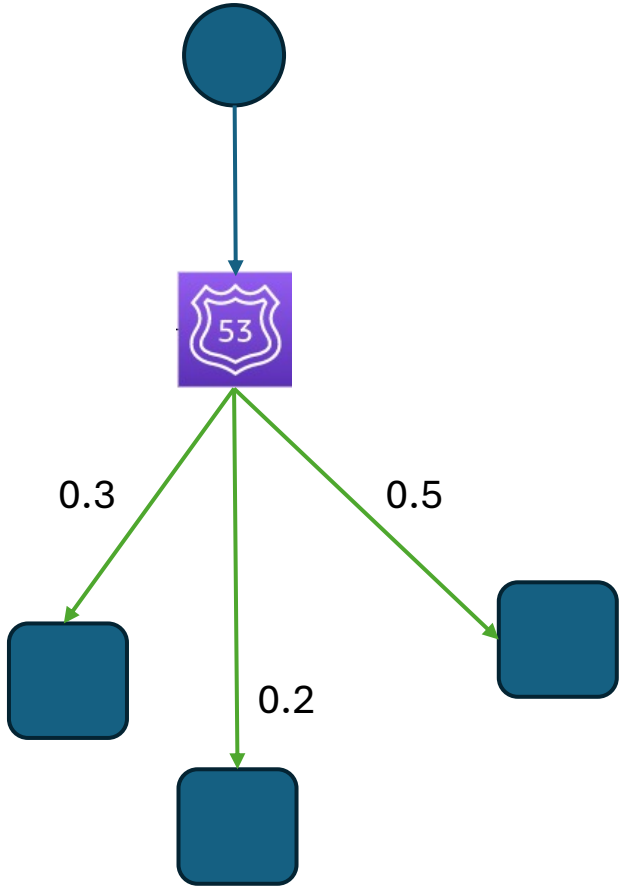
1 - Latency routing policy



2 – Failover routing policy



3 – Weighted routing policy



Route 53

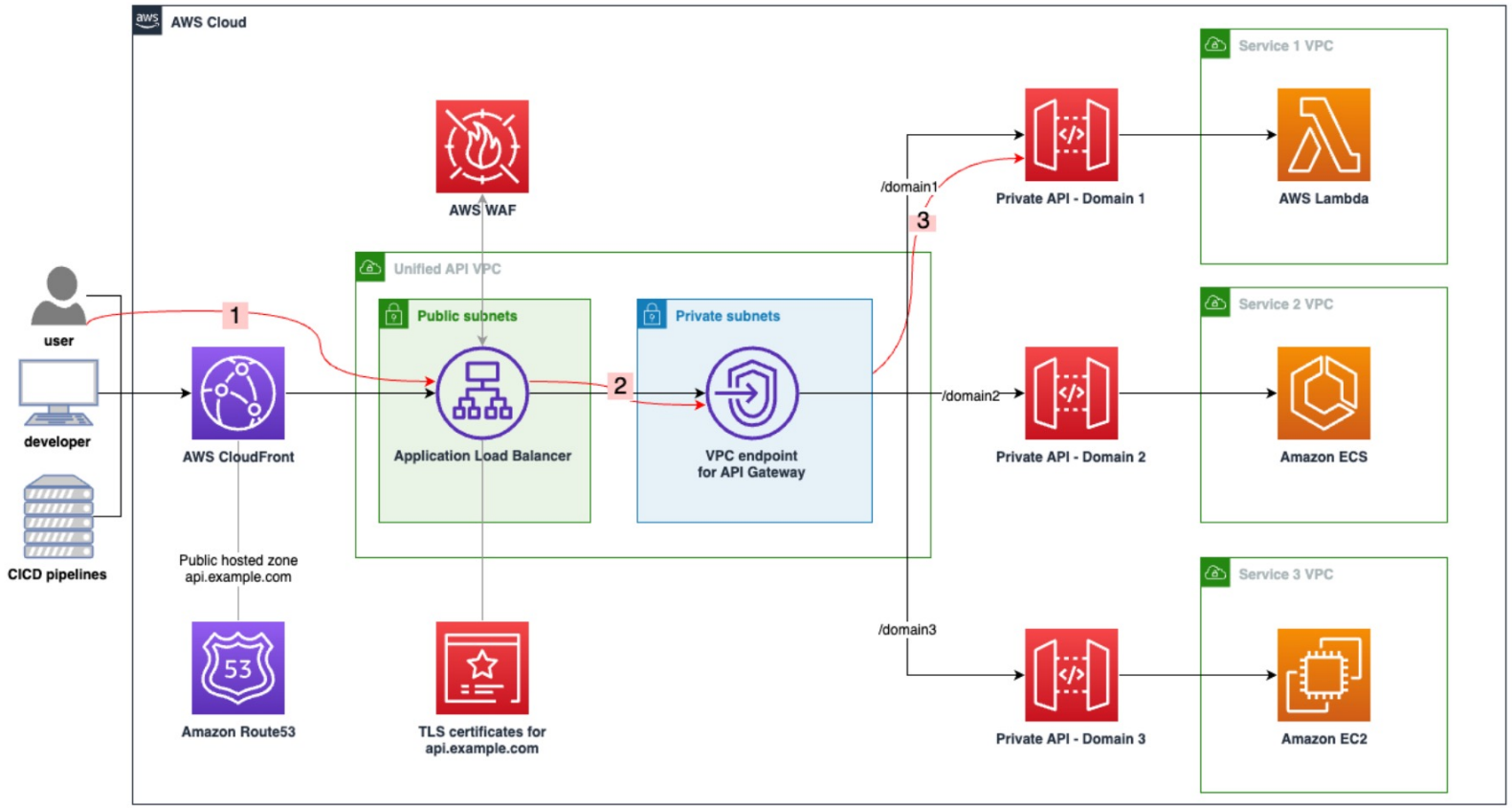


Figure 1. Unified API architecture

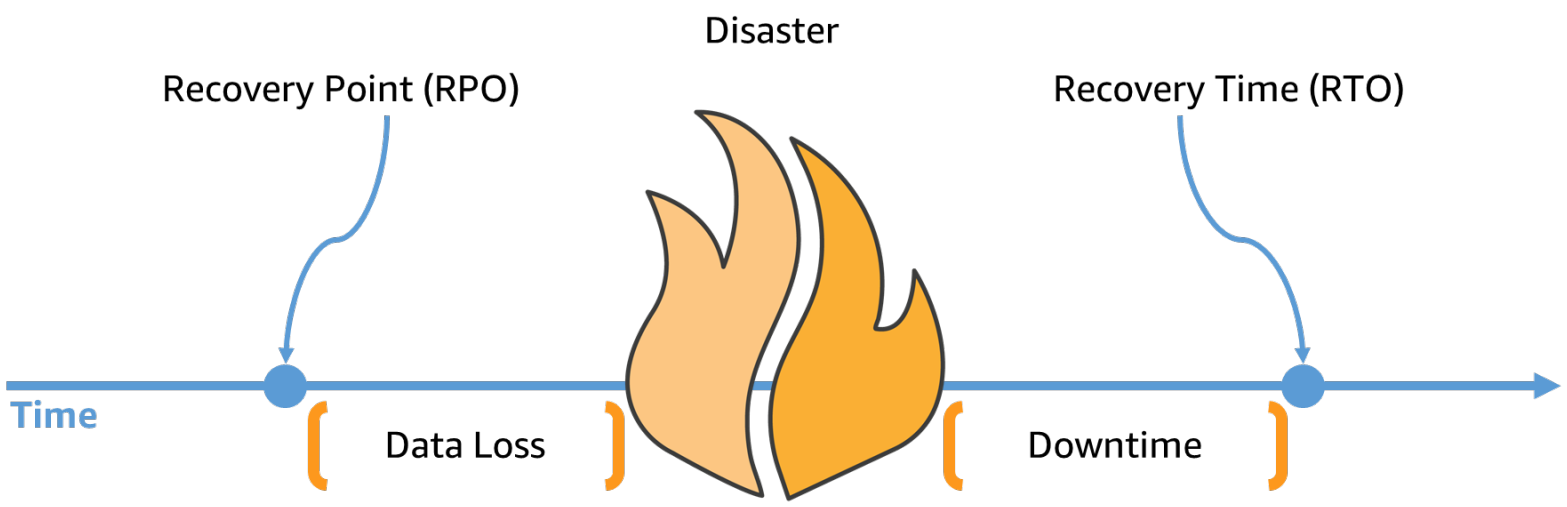
Figure 1. Unified API architecture

https://aws.amazon.com/blogs/architecture/how-sonar-built-a-unified-api-on-aws/

Estratégias de recuperação de desastre

How much data can you afford to recreate or lose?

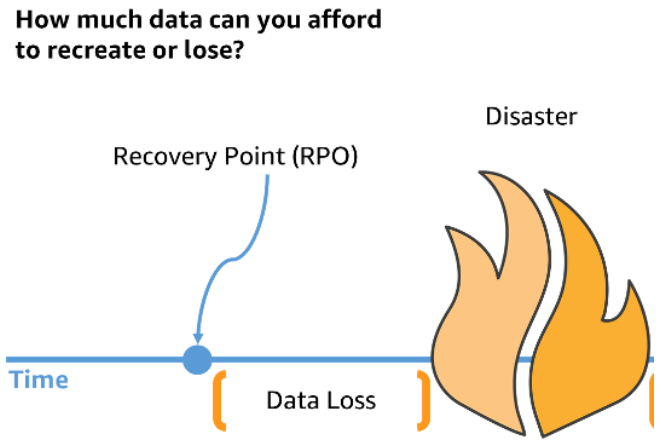
How quickly must you recover? What is the cost of downtime?



RPO: Recovery Point Objective

ESTAMOS OLHANDO PARA O PASSADO

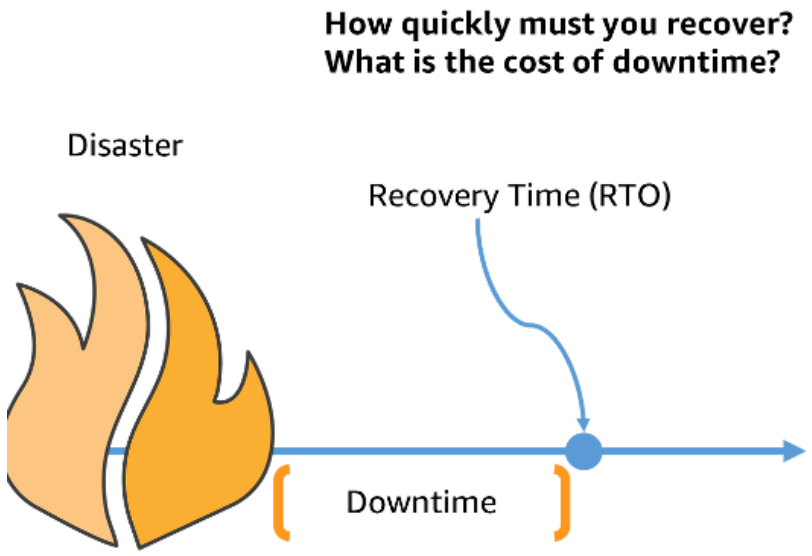
RPO, ou Recovery Point Objective, é a quantidade máxima de dados que sua aplicação pode perder sem causar grandes impactos. Em outras palavras, o RPO define até que ponto no tempo os dados podem ser recuperados, indicando o **quão recentes eles precisam estar no momento da recuperação**.



RTO: Recovery Time Objective

ESTAMOS OLHANDO PARA O FUTURO

RTO significa Recovery Time Objective e é uma medida de quão rapidamente após uma interrupção um aplicativo deve estar disponível novamente.

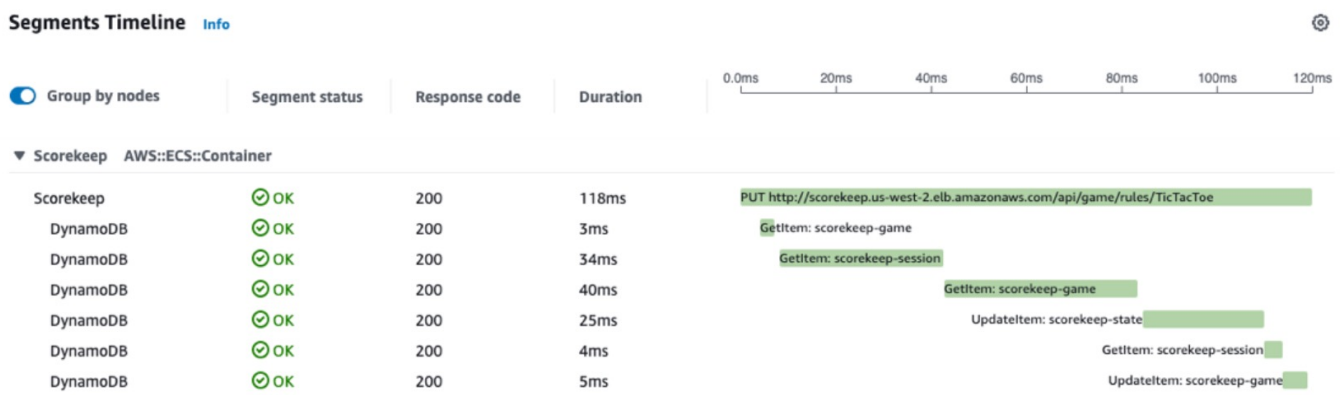


X-Ray

O que é AWS X-Ray?

PDF | RSS

AWS X-Ray é um serviço que coleta dados sobre solicitações atendidas pelo seu aplicativo e fornece ferramentas que você pode usar para visualizar, filtrar e obter informações sobre esses dados para identificar problemas e oportunidades de otimização. Para qualquer solicitação rastreada para seu aplicativo, você pode ver informações detalhadas não apenas sobre a solicitação e a resposta, mas também sobre as chamadas que seu aplicativo faz para AWS recursos downstream, microsserviços, bancos de dados e web. APIs



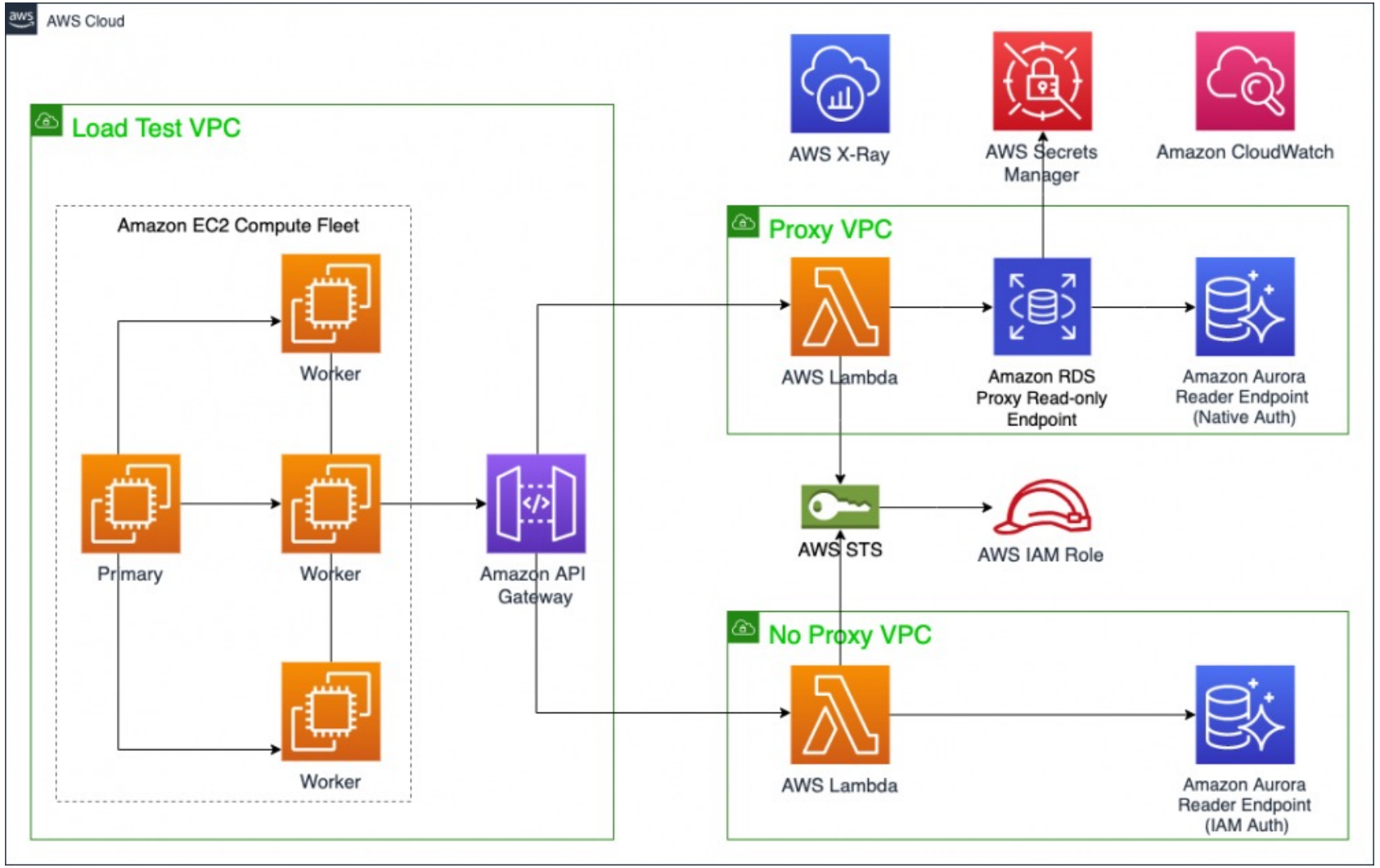
Usar o Amazon RDS Proxy

PDF | RSS

Com o proxy do Amazon RDS, você pode permitir que suas aplicações agrupem e compartilhem conexões de banco de dados para melhorar sua capacidade de escala. O proxy do RDS torna as aplicações mais resilientes a falhas de banco de dados conectando-se automaticamente a uma instância de banco de dados em espera e preservando as conexões de aplicações. Ao usar o RDS Proxy, você também pode impor a autenticação do AWS Identity and Access Management (IAM) para bancos de dados e armazenar credenciais com segurança no AWS Secrets Manager.

Com o RDS Proxy, você pode lidar com picos imprevisíveis no tráfego de banco de dados. Caso contrário, esses picos podem causar problemas devido a conexões com excesso de assinaturas ou à criação rápida de conexões. O RDS Proxy estabelece um grupo de conexões de banco de dados e reutiliza conexões nesse grupo. Essa abordagem evita sobrecarregar a memória e a CPU de abrir uma nova conexão de banco de dados todas as vezes. Para proteger um banco de dados contra o excesso de assinaturas, é possível controlar o número de conexões do banco de dados criadas.

RDS Proxy



https://aws.amazon.com/blogs/database/build-and-load-test-a-multi-tenant-saas-database-proxy-solution-with-amazon-rds-proxy/

Exercícios

Questão 1 - Cenário:

Uma grande empresa de comércio eletrônico armazena diariamente enormes volumes de dados sensíveis em um bucket do Amazon S3. Esses dados incluem informações pessoais dos clientes, como números de cartão de crédito e dados de contato. A empresa precisa garantir que esses dados sensíveis sejam identificados, classificados e protegidos de forma adequada para cumprir as regulamentações de segurança e privacidade de dados. Além disso, eles desejam receber alertas automaticamente sempre que dados sensíveis forem detectados em novos arquivos carregados no bucket S3.

Pergunta:

Como arquiteto de soluções, qual das opções abaixo representa a melhor solução para atender às necessidades da empresa, utilizando AWS Macie e S3?

A. Configurar o AWS Macie para monitorar continuamente o bucket S3, identificar dados sensíveis e enviar notificações via Amazon SNS quando dados confidenciais forem detectados.
B. Utilizar AWS Lambda para analisar os arquivos no bucket S3 e identificar dados sensíveis, e em seguida, configurar regras de bucket para bloquear o acesso a esses arquivos.
C. Configurar políticas de bucket S3 para criptografar todos os arquivos carregados e garantir que apenas usuários autenticados possam acessar os dados.
D. Utilizar Amazon GuardDuty para monitorar o tráfego de rede do bucket S3 e enviar alertas se detectar acessos não autorizados aos dados sensíveis.

Resposta Correta: A. Configurar o AWS Macie para monitorar continuamente o bucket S3, identificar dados sensíveis e enviar notificações via Amazon SNS quando dados confidenciais forem detectados.

Explicação:
O AWS Macie é um serviço gerenciado de segurança de dados e privacidade que usa machine learning para descobrir, classificar e proteger dados sensíveis armazenados em buckets S3. No contexto do cenário apresentado, o AWS Macie pode ser configurado para monitorar continuamente os dados armazenados no S3 e identificar informações confidenciais, como números de cartão de crédito ou dados pessoais. Quando o Macie detecta dados sensíveis, ele pode gerar alertas e enviar notificações via Amazon SNS, permitindo que a empresa tome as medidas de segurança necessárias.
Outras opções mencionadas, como o uso de AWS Lambda ou Amazon GuardDuty, não são as mais adequadas para este cenário específico, já que o Macie é especialmente projetado para lidar com a descoberta e classificação de dados sensíveis em S3. A criptografia com políticas de bucket S3, embora importante, não aborda diretamente o requisito de identificação e notificação de dados sensíveis.

Referência na Documentação da AWS:

- AWS Macie - Visão Geral
- Amazon S3 - Boas Práticas de Segurança
- Amazon SNS - Serviço de Notificação Simples

Esse exemplo ajuda a pensar em como usar o AWS Macie em um cenário real de segurança e conformidade de dados, aplicando as práticas recomendadas de arquitetura em nuvem.

Questão 2 - Cenário:

Uma empresa de software financeiro utiliza um banco de dados relacional na AWS, hospedado no Amazon RDS, para gerenciar dados críticos de transações financeiras. Para garantir alta disponibilidade e minimizar o tempo de inatividade em caso de falhas, a empresa configurou uma instância RDS Multi-AZ. Recentemente, eles passaram por uma falha de hardware que afetou a instância principal do RDS.
Como arquiteto de soluções, você foi chamado para revisar a configuração e garantir que a arquitetura esteja preparada para futuras falhas. A equipe de operações deseja entender melhor como o Amazon RDS lida com failovers em uma configuração Multi-AZ e como minimizar o impacto em seus aplicativos.

Pergunta:
Qual das seguintes afirmações descreve corretamente como o Amazon RDS Multi-AZ lida com failovers e quais práticas recomendadas você deve seguir para minimizar o impacto em seus aplicativos durante um failover?

a) O Amazon RDS Multi-AZ automaticamente realiza o failover para a réplica em standby em outra Zona de Disponibilidade em caso de falha da instância principal, sem nenhuma intervenção manual. Durante o failover, os endpoints de conexão do banco de dados mudam automaticamente para a nova instância primária.

b) O Amazon RDS Multi-AZ exige intervenção manual para realizar o failover, e os endpoints de conexão devem ser atualizados manualmente nos aplicativos para apontar para a nova instância primária.

c) O Amazon RDS Multi-AZ automaticamente realiza o failover, mas a réplica em standby pode estar em outra região da AWS, o que pode resultar em maior latência para os aplicativos durante o failover.

d) O Amazon RDS Multi-AZ realiza automaticamente o failover, mas o backup automático deve ser desativado durante o processo de failover para evitar perda de dados.

Resposta Correta: a) O Amazon RDS Multi-AZ automaticamente realiza o failover para a réplica em standby em outra Zona de Disponibilidade em caso de falha da instância principal, sem nenhuma intervenção manual. Durante o failover, os endpoints de conexão do banco de dados mudam automaticamente para a nova instância primária.

Explicação:
O Amazon RDS Multi-AZ é projetado para fornecer alta disponibilidade e resiliência a falhas. Em uma configuração Multi-AZ, o RDS mantém uma réplica em standby em uma Zona de Disponibilidade (AZ) diferente da instância primária. Se a instância principal falhar devido a problemas de hardware, falhas na rede, ou manutenção planejada, o Amazon RDS executa automaticamente um failover para a réplica em standby. Durante esse processo, o endpoint de conexão do banco de dados é atualizado automaticamente para apontar para a nova instância primária, minimizando o impacto para os aplicativos que estão conectados ao banco de dados.
As práticas recomendadas incluem:
•Garantir que os aplicativos usem o endpoint fornecido pelo RDS, e não o endereço IP diretamente, para permitir a transição automática durante o failover.
•Monitorar os eventos do RDS para entender o tempo de failover e ajustar os parâmetros de timeout dos aplicativos para lidar com a breve indisponibilidade durante o failover.
As outras opções estão incorretas porque:
•b) O failover no RDS Multi-AZ não requer intervenção manual e o endpoint é automaticamente atualizado.
•c) A réplica em standby em um cenário Multi-AZ está sempre na mesma região, mas em uma Zona de Disponibilidade diferente, evitando latência adicional significativa.
•d) Os backups automáticos não precisam ser desativados durante um failover e continuarão a proteger os dados conforme configurado.

Referência na Documentação da AWS:
•Amazon RDS Multi-AZ Deployments
•Best Practices for Working with Amazon RDS
Esse exemplo aborda as estratégias de failover no Amazon RDS e ajuda os alunos a entender como projetar para alta disponibilidade e minimizar o impacto no desempenho dos aplicativos durante interrupções.

Q3 - Cenário:

Uma empresa de tecnologia está migrando sua infraestrutura para a AWS e configurando suas redes virtuais usando Amazon VPC. Como parte do processo, eles precisam garantir que suas instâncias EC2 sejam seguras, controlando o tráfego de rede que entra e sai dessas instâncias. A equipe de arquitetura está discutindo a utilização de **Security Groups**e **Network ACLs (NACLs)** para implementar essas medidas de segurança.
A equipe tem dúvidas sobre como configurar corretamente essas ferramentas para garantir a proteção adequada das instâncias EC2, considerando as diferenças entre grupos de segurança stateful e ACLs de rede stateless.

Pergunta:

Com base nesse cenário, qual das seguintes afirmações descreve corretamente o comportamento dos Security Groups e das NACLs e como eles devem ser usados em conjunto para proteger as instâncias EC2?

a) **Security Groups** são stateful, o que significa que se uma regra permitir o tráfego de entrada, o tráfego de saída correspondente será automaticamente permitido. As **NACLs** são stateless, o que significa que regras de entrada e saída devem ser explicitamente definidas. Para máxima segurança, use **Security Groups** para controlar o tráfego de entrada e **NACLs** para controlar o tráfego de saída.
b) **Security Groups** são stateless e exigem que regras de entrada e saída sejam definidas separadamente. As **NACLs** são stateful e permitem que o tráfego de retorno seja automaticamente permitido. Use **Security Groups** para definir regras granulares de tráfego e as **NACLs** para garantir que todo o tráfego de retorno seja permitido.
c) **Security Groups** e **NACLs** são ambos stateful e, portanto, as regras de entrada automaticamente permitem o tráfego de saída correspondente. Use **NACLs** para aplicar regras de segurança ao nível da instância EC2 e **Security Groups** para aplicar regras ao nível da sub-rede.
d) **Security Groups** são stateless e aplicam regras de segurança ao nível da sub-rede. As **NACLs** são stateful e aplicam regras ao nível da instância EC2. Use **NACLs** para controlar o tráfego de saída e **Security Groups** para controlar o tráfego de entrada.

Resposta Correta:

a) **Security Groups** são stateful, o que significa que se uma regra permitir o tráfego de entrada, o tráfego de saída correspondente será automaticamente permitido. As **NACLs** são stateless, o que significa que regras de entrada e saída devem ser explicitamente definidas. Para máxima segurança, use **Security Groups** para controlar o tráfego de entrada e **NACLs** para controlar o tráfego de saída.

Explicação:

• **Security Groups** são stateful, o que significa que eles rastreiam as conexões. Se uma regra de segurança permitir o tráfego de entrada, o tráfego de saída correspondente (resposta) será automaticamente permitido, e vice-versa. **Security Groups** são aplicados ao nível da instância EC2 e são usados principalmente para controlar o tráfego que entra e sai das instâncias.
• **NACLs** (Network ACLs) são stateless, o que significa que eles não rastreiam o estado das conexões. Para permitir tráfego bidirecional, é necessário criar regras separadas para entrada e saída. **NACLs** são aplicados ao nível da sub-rede, afetando todas as instâncias EC2 dentro da sub-rede.
• A combinação de **Security Groups** e **NACLs** permite um controle de segurança detalhado. Por exemplo, você pode usar **Security Groups** para definir regras mais específicas para cada instância EC2 e **NACLs** para aplicar regras de segurança mais amplas e genéricas ao nível da sub-rede.
As outras opções estão incorretas porque:
• b) **Security Groups** são stateful, não stateless.
• c) **NACLs** são stateless, não stateful, e não são aplicados ao nível da instância.
• d) **Security Groups** são aplicados ao nível da instância EC2, não ao nível da sub-rede.

Referência na Documentação da AWS:
• Security Groups for Your VPC
• Network ACLs
Esse exemplo ajuda os alunos a entender a diferença entre o comportamento stateful dos Security Groups e o comportamento stateless das NACLs, além de como essas ferramentas devem ser usadas em conjunto para proporcionar segurança robusta na AWS.

Q4 - Cenário:

Uma startup está implementando sua infraestrutura na AWS para hospedar uma aplicação web crítica. Eles optaram por uma arquitetura que inclui uma VPC com duas sub-redes: uma sub-rede pública onde um balanceador de carga (ELB) é implantado e uma sub-rede privada onde as instâncias EC2 que executam o backend da aplicação estão localizadas. Por razões de segurança, as instâncias EC2 na sub-rede privada não têm endereços IP públicos.
Recentemente, a equipe de operações percebeu que as instâncias EC2 precisam baixar atualizações de segurança regulares e pacotes de software da internet, mas, por razões de segurança, elas não devem estar diretamente acessíveis pela internet.

Pergunta:
Qual solução você recomendaria para permitir que as instâncias EC2 na sub-rede privada baixem atualizações da internet, mantendo as boas práticas de segurança?

a) Associar um IP público às instâncias EC2 na sub-rede privada temporariamente para que possam acessar a internet e depois remover o IP público após as atualizações serem concluídas.
b) Criar uma nova sub-rede pública, mover as instâncias EC2 para essa sub-rede e associar IPs públicos para que possam acessar a internet.
c) Configurar um NAT Gateway na sub-rede pública e atualizar a tabela de rotas da sub-rede privada para direcionar o tráfego de saída para a internet através do NAT Gateway.
d) Implementar um bastion host na sub-rede privada para encaminhar o tráfego de internet para as instâncias EC2 usando regras de roteamento avançadas.

Resposta Correta: c) Configurar um NAT Gateway na sub-rede pública e atualizar a tabela de rotas da sub-rede privada para direcionar o tráfego de saída para a internet através do NAT Gateway.

Explicação:

Para permitir que instâncias EC2 em uma sub-rede privada acessem a internet sem expô-las diretamente, a solução recomendada é utilizar um **NAT Gateway**. O NAT Gateway é implantado em uma sub-rede pública e permite que instâncias em uma sub-rede privada iniciem conexões de saída para a internet (por exemplo, para baixar atualizações), mas impede que conexões não solicitadas da internet cheguem às instâncias.

Passos para Implementar a Solução:

1. **Criar um NAT Gateway**: Implante o NAT Gateway na sub-rede pública. Ele será associado a um Elastic IP (EIP) para se comunicar com a internet.
2. **Atualizar a Tabela de Rotas da Sub-rede Privada**: Modifique a tabela de rotas associada à sub-rede privada para direcionar o tráfego de saída destinado à internet (0.0.0.0/0) para o NAT Gateway.
3. **Manter a Segurança**: Essa abordagem mantém as instâncias EC2 na sub-rede privada inacessíveis diretamente da internet, alinhando-se às melhores práticas de segurança.

As outras opções estão incorretas porque:

- a) Associar um IP público diretamente às instâncias EC2 na sub-rede privada compromete a segurança, tornando-as acessíveis pela internet.
- b) Mover as instâncias para uma sub-rede pública as exporia diretamente à internet, o que não é recomendado para instâncias de backend sensíveis.
- d) Um bastion host não é apropriado para encaminhar tráfego de atualização de software para as instâncias EC2. Bastion hosts são geralmente usados para SSH/RDP em instâncias EC2 privadas de maneira segura.

Referência na Documentação da AWS:

- NAT Gateways
- Subnets in Your VPC

Esse exemplo ajuda os alunos a entender como utilizar um NAT Gateway para permitir acesso seguro à internet para instâncias em sub-redes privadas, enquanto explora conceitos de sub-redes públicas e privadas na AWS.

Q5 - Cenário:

Uma empresa de comércio eletrônico global utiliza o Amazon RDS para gerenciar seu banco de dados relacional que armazena informações críticas de transações de clientes. Para garantir alta disponibilidade e resiliência a desastres, a empresa está considerando diferentes estratégias de replicação de dados.
Eles estão avaliando entre replicação síncrona e assíncrona no Amazon RDS para atender às suas necessidades de consistência de dados e recuperação de desastres. A empresa tem escritórios em várias regiões e quer garantir que os dados estejam disponíveis com o menor tempo de recuperação possível em caso de falhas regionais, ao mesmo tempo em que minimizam a latência de escrita.

Pergunta:
Com base nesse cenário, qual das seguintes afirmações descreve corretamente as diferenças entre replicação síncrona e assíncrona no Amazon RDS e como cada uma pode impactar a arquitetura da empresa?

a) A replicação síncrona no RDS é usada principalmente em implantações Multi-AZ, garantindo que os dados sejam gravados simultaneamente em duas zonas de disponibilidade diferentes, o que proporciona alta disponibilidade e baixa latência de leitura. A replicação assíncrona é usada para replicação entre regiões (Cross-Region Replication), mas pode resultar em perda de dados em caso de falha na região primária.
b) A replicação assíncrona no RDS é usada principalmente em implantações Multi-AZ, garantindo que os dados sejam gravados simultaneamente em duas zonas de disponibilidade diferentes, o que proporciona alta disponibilidade e consistência forte. A replicação síncrona é usada para replicação entre regiões (Cross-Region Replication) para garantir que não haja perda de dados.
c) A replicação síncrona no RDS é usada para replicação entre regiões (Cross-Region Replication) para garantir que os dados estejam consistentemente disponíveis em múltiplas regiões, mas isso pode resultar em maior latência de escrita. A replicação assíncrona é usada em implantações Multi-AZ para garantir alta disponibilidade sem impactar a latência de escrita.
d) A replicação assíncrona no RDS é usada para replicação entre regiões (Cross-Region Replication) para garantir que os dados estejam disponíveis em múltiplas regiões, mas pode haver uma perda de dados em caso de falha na região primária. A replicação síncrona é usada em implantações Multi-AZ para garantir que os dados sejam gravados simultaneamente em duas zonas de disponibilidade diferentes, proporcionando alta disponibilidade sem perda de dados.

Resposta Correta: d) A replicação assíncrona no RDS é usada para replicação entre regiões (Cross-Region Replication) para garantir que os dados estejam disponíveis em múltiplas regiões, mas pode haver uma perda de dados em caso de falha na região primária. A replicação síncrona é usada em implantações Multi-AZ para garantir que os dados sejam gravados simultaneamente em duas zonas de disponibilidade diferentes, proporcionando alta disponibilidade sem perda de dados.

Explicação:

•**Replicação Síncrona**: No Amazon RDS, a replicação síncrona é utilizada em implantações Multi-AZ. Quando a replicação síncrona é habilitada, o RDS grava os dados simultaneamente na instância primária e na réplica em standby, localizada em uma zona de disponibilidade diferente. Isso garante que, em caso de falha na instância primária, uma réplica consistente esteja disponível para failover imediato, proporcionando alta disponibilidade sem perda de dados.

•**Replicação Assíncrona**: A replicação assíncrona é utilizada em configurações de replicação entre regiões (Cross-Region Replication), como ao utilizar réplicas de leitura em diferentes regiões. Nesse caso, há um pequeno atraso (lag) na replicação, o que significa que, em caso de falha na região primária, pode haver perda de dados que ainda não foram replicados para a região secundária. A replicação assíncrona é útil para cenários de recuperação de desastres, onde a consistência forte em tempo real entre regiões não é um requisito crítico, mas a disponibilidade global é.

As outras opções estão incorretas porque:
•a) Confunde os usos de replicação síncrona e assíncrona, atribuindo incorretamente a replicação assíncrona ao Multi-AZ.
•b) Também confunde os usos de replicação síncrona e assíncrona, atribuindo características incorretas à replicação entre regiões e Multi-AZ.
•c) Afirmar que a replicação síncrona é usada para replicação entre regiões está incorreto; essa replicação é tipicamente assíncrona.
Referência na Documentação da AWS:
•Multi-AZ Deployments for Amazon RDS
•Working with Read Replicas
Esse exemplo ajuda os alunos a entender as diferenças entre replicação síncrona e assíncrona no Amazon RDS, suas implicações para alta disponibilidade e recuperação de desastres, e como essas configurações impactam a arquitetura da AWS.

Q6 - Cenário:

Uma empresa global de mídia possui uma aplicação web que é acessada por usuários em várias regiões ao redor do mundo. Para garantir uma melhor experiência de usuário, a empresa decidiu distribuir sua aplicação em várias regiões da AWS, incluindo a América do Norte, Europa e Ásia. O objetivo é direcionar os usuários para a região mais próxima para minimizar a latência e melhorar o desempenho.
A empresa está utilizando o Amazon Route 53 como seu serviço de DNS e deseja implementar uma estratégia de roteamento que automaticamente direcione os usuários para a instância da aplicação na região geograficamente mais próxima.

Pergunta:

Qual política de roteamento do Route 53 você recomendaria para atender a esse requisito?
a) Política de Roteamento Ponderado (Weighted Routing Policy)
b) Política de Roteamento Baseado em Latência (Latency Routing Policy)
c) Política de Roteamento Geográfico (Geolocation Routing Policy)
d) Política de Roteamento Failover (Failover Routing Policy)

Resposta Correta: b) Política de Roteamento Baseado em Latência (Latency Routing Policy)

Explicação:

A **Política de Roteamento Baseado em Latência** do Route 53 permite que você direcione os usuários para a região da AWS que oferece a menor latência de rede, melhorando a experiência de uso ao reduzir o tempo de resposta da aplicação. Essa política não considera a localização geográfica exata dos usuários, mas sim a latência da rede entre o usuário e as regiões disponíveis.

•A **Política de Roteamento Geográfico (Geolocation Routing Policy)**, embora útil para direcionar usuários com base na localização geográfica, pode não ser tão eficiente quanto a política de latência quando o objetivo é otimizar o desempenho com base na latência.

•A **Política de Roteamento Ponderado (Weighted Routing Policy)** permite distribuir o tráfego com base em ponderações específicas, mas não otimiza a latência.

•A **Política de Roteamento Failover (Failover Routing Policy)** é usada para garantir alta disponibilidade ao redirecionar o tráfego para uma instância de backup em caso de falha, e não para otimizar o desempenho com base na latência.

Referência na Documentação da AWS:
•Choosing a Routing Policy
•Latency Routing

Q7 - Cenário:

Uma startup de e-commerce lançou recentemente sua plataforma online, que atende a clientes em várias partes do mundo. A equipe de TI quer garantir que, em certas regiões específicas, como a Europa e a Ásia, os usuários sejam sempre direcionados para servidores localizados naquelas regiões, independentemente da latência da rede. Para regiões onde a empresa ainda não possui servidores, eles querem que o tráfego seja redirecionado para o servidor principal nos Estados Unidos.
Eles estão usando o Amazon Route 53 para gerenciar o DNS e precisam configurar o serviço para atender a esse requisito.

Pergunta:
Qual política de roteamento do Route 53 você deve recomendar para garantir que os usuários sejam direcionados com base na sua localização geográfica, conforme descrito no cenário?

a) Política de Roteamento Ponderado (Weighted Routing Policy)
b) Política de Roteamento Baseado em Latência (Latency Routing Policy)
c) Política de Roteamento Geográfico (Geolocation Routing Policy)
d) Política de Roteamento Multi-valor (Multivalue Answer Routing Policy)

Resposta Correta: c) Política de Roteamento Geográfico (Geolocation Routing Policy)

Explicação:

A **Política de Roteamento Geográfico** no Route 53 permite direcionar os usuários com base na localização geográfica do solicitante, ou seja, onde a solicitação DNS está sendo originada. No cenário apresentado, essa política é ideal para garantir que usuários em regiões específicas, como Europa e Ásia, sejam direcionados para servidores locais nessas regiões. Para usuários em outras regiões, onde não há servidores disponíveis, o tráfego pode ser direcionado para o servidor principal nos Estados Unidos.

- A **Política de Roteamento Baseado em Latência (Latency Routing Policy)** prioriza a menor latência e não a localização geográfica.
- A **Política de Roteamento Ponderado (Weighted Routing Policy)** permite distribuir o tráfego com base em ponderações, mas não leva em conta a localização geográfica do usuário.
- A **Política de Roteamento Multi-valor (Multivalue Answer Routing Policy)** é usada para retornar vários registros de IP com verificações de saúde associadas, mas não é projetada para direcionar com base na localização geográfica.

Referência na Documentação da AWS:

- Geolocation Routing
- Choosing a Routing Policy

Essas perguntas ajudam a explorar os diferentes cenários de roteamento que podem ser implementados usando o Amazon Route 53, permitindo que os alunos compreendam melhor como utilizar essas políticas de roteamento em situações reais.

Cenário:

Uma empresa de tecnologia está desenvolvendo uma aplicação web que se conecta a um banco de dados MySQL hospedado no Amazon RDS. Por razões de segurança, a equipe de engenharia quer garantir que as credenciais do banco de dados (como nome de usuário e senha) sejam armazenadas de forma segura e que possam ser facilmente recuperadas pela aplicação sem a necessidade de hardcoding.
A empresa já utiliza o AWS Certificate Manager (ACM) para gerenciar certificados SSL/TLS para suas aplicações web e o AWS Key Management Service (KMS) para criptografar dados sensíveis. Eles também estão cientes do AWS Systems Manager Parameter Store, mas não têm certeza de qual serviço utilizar para armazenar e gerenciar as credenciais do banco de dados.

Pergunta:

Qual solução você recomendaria para armazenar e gerenciar as credenciais do banco de dados de maneira segura, permitindo que a aplicação as recupere dinamicamente quando necessário?
a) Armazenar as credenciais no **AWS Certificate Manager (ACM)**, pois ele já está sendo utilizado para gerenciar certificados SSL/TLS e pode lidar com informações sensíveis.
b) Utilizar o **AWS Key Management Service (KMS)** para criptografar e armazenar as credenciais, uma vez que ele é usado para gerenciar chaves de criptografia e pode proteger dados sensíveis.
c) Armazenar as credenciais no **AWS Systems Manager Parameter Store** com a opção de parâmetro seguro (SecureString) e criptografá-las usando o KMS.
d) Utilizar o **AWS Secrets Manager** para armazenar e gerenciar as credenciais do banco de dados, aproveitando a rotação automática de credenciais e a fácil integração com o RDS.

Resposta Correta: d) Utilizar o **AWS Secrets Manager** para armazenar e gerenciar as credenciais do banco de dados, aproveitando a rotação automática de credenciais e a fácil integração com o RDS.

Explicação:

O **AWS Secrets Manager** é a solução ideal para armazenar e gerenciar credenciais de banco de dados de forma segura. Ele permite armazenar as credenciais de forma criptografada e oferece a funcionalidade de rotação automática de credenciais, o que melhora a segurança e reduz o risco de comprometimento de credenciais. Além disso, o Secrets Manager se integra facilmente com o Amazon RDS, facilitando a recuperação das credenciais pela aplicação de forma dinâmica.

- **AWS Certificate Manager (ACM)** é utilizado para gerenciar certificados SSL/TLS, mas não é adequado para armazenar credenciais de banco de dados.
- **AWS Key Management Service (KMS)** é usado para gerenciar chaves de criptografia e pode ser utilizado em conjunto com outros serviços (como Parameter Store), mas não oferece as funcionalidades específicas de gerenciamento de credenciais que o Secrets Manager oferece.
- **AWS Systems Manager Parameter Store** com SecureString é uma opção válida para armazenar dados sensíveis, mas não oferece a mesma integração e funcionalidades avançadas de rotação automática que o Secrets Manager oferece para credenciais de banco de dados.

Referência na Documentação da AWS:

- AWS Secrets Manager Documentation
- Managing Secrets for Amazon RDS Databases
- AWS Key Management Service (KMS)
- AWS Systems Manager Parameter Store

Essa pergunta ajuda os alunos a entender como escolher o serviço AWS mais adequado para gerenciar e proteger credenciais de banco de dados, destacando a importância da rotação automática e da integração fácil com outros serviços AWS.

Cenário:

Uma empresa de comércio eletrônico está lançando uma nova plataforma web que permite que os clientes façam compras online. A equipe de segurança está preocupada com a proteção da aplicação contra ameaças comuns na web, como cross-site scripting (XSS) e injeção de SQL (SQLi), que podem comprometer a segurança dos dados dos clientes.
A empresa já utiliza diversos serviços de segurança da AWS, como AWS WAF, AWS Firewall Manager, AWS GuardDuty e Network ACLs (NACLs), mas a equipe de engenharia não tem certeza de qual combinação de serviços e práticas é mais eficaz para mitigar esses tipos de ataques.

Pergunta:

Qual das seguintes combinações de serviços e práticas você recomendaria para proteger a aplicação web contra ataques de cross-site scripting (XSS) e injeção de SQL (SQLi)?
a) Utilizar **AWS WAF** para criar regras específicas que bloqueiem padrões de ataque XSS e SQLi, e **AWS GuardDuty**para monitorar atividades maliciosas em todo o ambiente.
b) Configurar **Network ACLs (NACLs)** para bloquear o tráfego de entrada suspeito e **AWS Firewall Manager** para gerenciar regras de segurança centralizadas em todas as contas.
c) Usar **AWS GuardDuty** para detectar tentativas de ataque XSS e SQLi e **AWS Firewall Manager** para bloquear automaticamente essas tentativas com base nas detecções.
d) Implementar **AWS WAF** para bloquear XSS e SQLi e configurar **Network ACLs (NACLs)** para controlar o tráfego de entrada e saída nas sub-redes da VPC.

Cenário:

Uma empresa de comércio eletrônico está lançando uma nova plataforma web que permite que os clientes façam compras online. A equipe de segurança está preocupada com a proteção da aplicação contra ameaças comuns na web, como cross-site scripting (XSS) e injeção de SQL (SQLi), que podem comprometer a segurança dos dados dos clientes.
A empresa já utiliza diversos serviços de segurança da AWS, como AWS WAF, AWS Firewall Manager, AWS GuardDuty e Network ACLs (NACLs), mas a equipe de engenharia não tem certeza de qual combinação de serviços e práticas é mais eficaz para mitigar esses tipos de ataques.

Pergunta:

Qual das seguintes combinações de serviços e práticas você recomendaria para proteger a aplicação web contra ataques de cross-site scripting (XSS) e injeção de SQL (SQLi)?
a) Utilizar **AWS WAF** para criar regras específicas que bloqueiem padrões de ataque XSS e SQLi, e **AWS GuardDuty**para monitorar atividades maliciosas em todo o ambiente.
b) Configurar **Network ACLs (NACLs)** para bloquear o tráfego de entrada suspeito e **AWS Firewall Manager** para gerenciar regras de segurança centralizadas em todas as contas.
c) Usar **AWS GuardDuty** para detectar tentativas de ataque XSS e SQLi e **AWS Firewall Manager** para bloquear automaticamente essas tentativas com base nas detecções.
d) Implementar **AWS WAF** para bloquear XSS e SQLi e configurar **Network ACLs (NACLs)** para controlar o tráfego de entrada e saída nas sub-redes da VPC.

Resposta Correta: a) Utilizar **AWS WAF** para criar regras específicas que bloqueiem padrões de ataque XSS e SQLi, e **AWS GuardDuty** para monitorar atividades maliciosas em todo o ambiente.

Explicação:

•**AWS WAF** (Web Application Firewall) é a ferramenta ideal para proteger aplicações web contra ameaças como cross-site scripting (XSS) e injeção de SQL (SQLi). Ele permite que você crie regras personalizadas para bloquear padrões de ataque específicos, como strings maliciosas que indicam XSS ou SQLi. O WAF é altamente configurável e pode ser integrado diretamente ao seu Application Load Balancer (ALB) ou API Gateway.

•**AWS GuardDuty** é um serviço de detecção de ameaças que monitora continuamente o seu ambiente AWS em busca de atividades maliciosas e comportamentos anômalos. Embora não bloqueie ataques diretamente, ele pode alertar sobre possíveis compromissos de segurança e ajudar a identificar comportamentos suspeitos que podem indicar tentativas de XSS ou SQLi.

As outras opções estão incorretas porque:

•b) **Network ACLs (NACLs)** são usadas para controlar o tráfego de entrada e saída em sub-redes, mas não oferecem a granularidade necessária para bloquear especificamente ataques de XSS e SQLi.

•c) **AWS GuardDuty** é uma ferramenta de detecção e não de bloqueio. Ele não pode bloquear tentativas de ataque diretamente, e **AWS Firewall Manager** é usado para gerenciar regras de segurança em larga escala, mas não especificamente para mitigar XSS e SQLi.

•d) Embora os **NACLs** ajudem a controlar o tráfego na VPC, eles não substituem o **AWS WAF** quando se trata de proteção contra ameaças de camada de aplicação, como XSS e SQLi.

Referência na Documentação da AWS:

•AWS WAF – Web Application Firewall

•AWS GuardDuty – Detecção de Ameaças

•Network ACLs

•AWS Firewall Manager

Essa pergunta ajuda os alunos a entender a melhor combinação de serviços AWS para proteger uma aplicação web contra ataques comuns como XSS e SQLi, destacando a importância de usar o AWS WAF para a defesa da camada de aplicação e o AWS GuardDuty para a detecção de atividades maliciosas em todo o ambiente.