

## 1 Uso de nuvem:

Ao optar por implementar a solução na nuvem em vez de on-premise, a decisão foi principalmente orientada pela flexibilidade, escalabilidade e custo-benefício que a nuvem oferece. Implementações na nuvem reduzem a necessidade de investimentos iniciais em hardware, além de oferecerem flexibilidade para escalar os recursos conforme a demanda varia. Além disso, a nuvem permite uma continuidade de negócios melhorada com capacidades integradas de backup e recuperação de desastre.

## 2 Implantação na AWS:

Entre os provedores de nuvem disponíveis, a escolha pela AWS foi fundamentada na ampla gama de serviços maduros e estáveis, uma robusta infraestrutura global e uma riqueza de funcionalidades que suportam workloads complexos de maneira eficiente e segura. Comparada ao Azure ou ao GCP, a AWS oferece vasta experiência de mercado, e um forte ecossistema de suporte e serviços complementares que são aderentes aos requisitos do projeto.

## 3 Amazon EKS + ECR (Kubernetes):

A adoção do Kubernetes é motivada pela sua capacidade de orquestrar contêineres de forma eficiente, promovendo a portabilidade e escalabilidade automática dos aplicativos. A escolha pelo Amazon EKS + ECR ao invés do Fargate ou AWS Lambda, se justifica pela necessidade de controle granular e flexibilidade em gerenciar a infraestrutura do Kubernetes, complementada por um robusto sistema de gerenciamento de contêineres que o ECR proporciona. Essa solução é mais adequada quando comparada ao Fargate, que gerencia contêineres de forma automática, mas não oferece o mesmo nível de controle, e ao Lambda, que é mais indicado para funções serverless de curta duração.

## 4 Banco de dados Relacional – PostgreSQL:

Optar por um banco de dados relacional é crucial devido à necessidade de manter integridade referencial e transações complexas características de aplicativos de

missão crítica. O PostgreSQL foi preferido ao MySQL ou MSSQL devido à sua rica gestão de funcionalidades, conformidade com padrões, e robusta capacidade de extensão. O uso do Amazon RDS for PostgreSQL é preferível sobre hospedar o banco de dados em um pod do Kubernetes, uma vez que o RDS oferece gestão automatizada de backups, segurança aprimorada e escalabilidade sem que haja a sobrecarga operacional de gerir o banco de dados.

## 5 Cache:

A introdução de cache na arquitetura melhora significativamente a resposta do sistema ao reduzir a carga nos bancos de dados através do armazenamento temporário de respostas frequentes. O Amazon ElastiCache é adotado devido à sua integração nativa com a infraestrutura AWS e melhoramento de performance sem a complexidade de gerenciar um Redis internamente num pod Kubernetes.

## 6 API Gateway:

O uso do Amazon API Gateway traz a vantagem de simplificar o processo de criação, publicação, manutenção e segurança das APIs em escala. Diferentemente de hospedar o gateway em um pod, o Amazon API Gateway abstrai complexidades de infraestrutura enquanto fornece features avançados como caching, análise de tráfego e controle monetário.

## 7 Load Balancer:

A inclusão de load balancers na solução assegura que o tráfego de rede seja distribuído eficientemente entre diversas instâncias, garantindo disponibilidade e resiliência do serviço.

## 8 CloudFront:

A utilização do Amazon CloudFront proporciona a distribuição global de conteúdo com baixa latência através de sua rede global habilitada por pontos de presença que garantem que os conteúdos sejam entregues de forma eficiente em qualquer lugar do globo.

## 9 CloudWatch:

Para monitoramento e observabilidade, o AWS CloudWatch é crucial. Ele habilita a coleta, visualização e atuação sob métricas e logs de todo o ecossistema AWS, mantendo saúde e desempenho dos aplicativos.

## 10 Secrets Manager:

Os segredos e credenciais são geridos com segurança através do AWS Secrets Manager, o qual permite rotação automática e proteção de credenciais essenciais, minimizando exposições de segurança.

## 11 KMS:

A gestão de criptografia é tratada com o AWS KMS, provendo uma maneira robusta de manusear chaves criptográficas para proteger dados sensíveis.

## 12 WAF (Web Application Firewall):

No campo da segurança, o Amazon WAF (Web Application Firewall) protege contra ameaças de rede e tráfegos maliciosos, assegurando integridade e desempenho no acesso aos serviços expostos na web.

## 13 Cognito:

O Amazon Cognito foi incorporado para gerenciar autenticação e autorização de forma integrada e segura, simplificando a implementação de identidade e controle de acesso em aplicações críticas.

## 14 CloudFormation:

Para automatizar e controlar infraestruturas, o AWS CloudFormation é empregado para gerenciar stacks de recursos AWS de modo repetível e previsível.

## 15 AWS X-Ray:

Por fim, o AWS X-Ray complementa a capacidade de traçamento de pedidos e debugging complexo, permitindo obter um vislumbre profundo de como os aplicativos interagem e de onde potencial fallback pode ser identificado.

Este conjunto de escolhas e a formação da arquitetura proposta visam atender não apenas às necessidades imediatas, mas também proporcionar flexibilidade e crescer com os desafios futuros que naturalmente surgirão.