

Migrar uma aplicação web legado para a AWS exige um planejamento complexo, estratégico e cuidadoso na execução para garantir um processo suave e sem interrupções para os usuários.

## **1. Planejamento e Auditoria**

### **1.1. Auditoria da Infraestrutura Atual:**

Deve-se Mapear os componentes, identificando todos ligados direta ou indiretamente com o sistema (servidores, bancos de dados, softwares etc.) e as dependências entre eles, inclusive externas.

Ainda, deve-se analisar as dependências de hardware, software e rede da aplicação, bem como a segurança da aplicação e da infraestrutura atual.

Analisar a performance atual do sistema, como latência, memória, uso da CPU, entre outros.

É preciso, também, identificar tecnologias legadas que possam exigir substituição ou refatoração.

### **1.2. Seleção dos Serviços Apropriados**

Mapear os componentes do sistema para os serviços da AWS.

Por exemplo, EC2 para servidores de aplicação, RDS para bancos de dados, S3 para armazenamento etc.

Para a arquitetura escolher entre uma arquitetura lift-and-shift (movendo a aplicação como está) ou refatoração para uma arquitetura baseada em microserviços ou serverless utilizando Lambda.

Avaliar custos e desempenho, utilizando a AWS Pricing Calculator para estimar custos e selecionar as melhores opções de equilíbrio de custo e desempenho.

Para gerenciar DNS e balanceamento de carga, usar Route 53.

### **1.3. Desenvolvimento do Plano de Migração**

Decidir quais componentes serão migrados inicialmente.

Escolher a estratégia de migração (Lift-and-Shift, re-arquitetura ou re-implementação).

Criar um cronograma, estabelecer prazos para cada fase da migração.

Definir e alocar os recursos humanos e financeiros necessários.

Criar um plano de contingência, definindo ações para lidar com problemas previstos ou não.

## **2. Migração e Teste**

### **2.1. Migração Gradual**

A migração dos componentes deve ocorrer em etapas, começando com componentes menos críticos e aumentando gradualmente o escopo.

Cada etapa deve ser testada. É preciso verificar a funcionalidade da aplicação após cada migração, e monitorar o desempenho da aplicação durante a migração.

## **2.2 Minimizar o Tempo de Inatividade**

Utilizar técnicas de downtime zero: Migrar os componentes da aplicação sem interrupções.

Utilizar um balanceador de carga: Redirecionar o tráfego para a nova infraestrutura durante a migração.

## **2.3. Ferramentas de Teste e Monitoramento**

Para uma migração mais eficiente, é preciso criar um ambiente de teste para testar o sistema em um ambiente similar ao da produção.

E para o monitoramento, pode-se utilizar as ferramentas abaixo:

Amazon CloudWatch para monitorar métricas de performance da aplicação e ajuste os recursos conforme necessário.

Amazon X-Ray, para depurar e analisar o desempenho da aplicação.

## **3. Benefícios da Migração e Otimização:**

### **3.1. Benefícios da AWS**

Escalabilidade e Elasticidade. A AWS permite o escalonamento automático de recursos (Auto Scaling), garantindo a adaptação do sistema à demanda variável.

Alta disponibilidade e recuperação de desastres, fazendo uso da infraestrutura global da AWS para garantir alta disponibilidade e implementar uma estratégia eficaz de backup e recuperação.

Integridades e Segurança. Podem-se os serviços de segurança da AWS, como IAM, WAF, e AWS Shield para proteger a aplicação de ameaças.

Custo-efetividade: pagar apenas pelos recursos utilizados.

Agilidade: Implantação de novas funcionalidades de forma rápida e fácil.

### **3.2. Otimização do Desempenho**

CDN com Amazon CloudFront: Para entregar conteúdo com baixa latência aos usuários globais, pode-se utilizar o Amazon CloudFront.

Cache: Usar o ElastiCache para armazenar em cache dados frequentemente acessados e reduzir a carga de trabalho em bancos de dados.

Monitoramento e Ajustes. Usar Amazon CloudWatch para monitorar métricas de performance da aplicação e ajuste os recursos conforme necessário.