

# Plano de Migração de Aplicação Legado para a AWS

## 1. Planejamento e Execução da Migração

### 1.1 Auditoria da Infraestrutura Atual

- Mapeamento de Componentes: Identificação de todos os componentes da aplicação, incluindo servidores, banco de dados, serviços de rede, e dependências externas. Documentação das versões do software, configurações atuais e interdependências.
- Identificação de Pontos Fracos: Avaliação de problemas de escalabilidade, disponibilidade e performance. Identificação de gargalos, como servidores sobrecarregados, falhas em redundância e limites de capacidade.
- Análise de Segurança: Revisão das políticas de segurança, configurações de firewall, e controle de acesso para garantir que essas práticas possam ser replicadas ou melhoradas na AWS.

### 1.2 Seleção de Serviços AWS

- EC2 (Elastic Compute Cloud): Hospedagem de servidores e instâncias de aplicação. Seleção de tipos de instâncias adequados para as necessidades da aplicação (por exemplo, instâncias de uso geral, otimizadas para memória ou CPU). Planejamento de grupos de Auto Scaling para ajuste automático de capacidade com base na demanda.
- RDS (Relational Database Service): Bancos de dados gerenciados. Escolha do tipo de banco de dados gerenciado mais adequado (MySQL, PostgreSQL, etc.). Configuração de réplicas e backups automáticos para garantir alta disponibilidade e recuperação de desastres.
- Elastic Load Balancing (ELB): Implementação de um balanceador de carga para distribuir o tráfego de entrada de forma eficiente entre várias instâncias EC2,

garantindo que a aplicação permaneça disponível mesmo em caso de falha de uma instância.

- S3 (Simple Storage Service): Armazenamento de arquivos estáticos, backups e logs. Aproveitamento do versionamento e controle de acesso granular do S3 para segurança adicional.

- CloudFront: Configuração de CDN para acelerar a entrega de conteúdo estático aos usuários, melhorando a experiência global.

- VPC (Virtual Private Cloud): Design de uma rede segura e isolada na AWS para hospedar os recursos da aplicação, com sub-redes públicas e privadas, controle de tráfego e gateways de internet e NAT.

- IAM (Identity and Access Management): Configuração de papéis e políticas de segurança para garantir que apenas usuários e serviços autorizados tenham acesso aos recursos críticos.

### **1.3 Execução da Migração**

- Desenvolvimento de um Plano de Fases:

- Planejamento de Migração: Criar um plano detalhado com etapas e cronograma.

- Fase 1: Configuração da infraestrutura na AWS, replicando o ambiente atual para testes.

- Fase 2: Migração de dados para o RDS e replicação de dados para minimizar o tempo de corte.

- Fase 3: Migração do código da aplicação e configuração das instâncias EC2.

- Fase 4: Implementação do ELB e configuração do DNS para direcionar o tráfego para a nova infraestrutura.

- Teste de Migração: Criação de um ambiente de pré-produção idêntico para realizar testes de carga, desempenho e segurança. Simulação de cenários de falhas para verificar a resiliência da nova arquitetura.

- Cutover Planejado: Planejamento da janela de migração com comunicação clara aos usuários sobre possíveis períodos de manutenção. Implementação de um plano de rollback caso ocorra algum problema.

## 2. Garantia de Migração sem Interrupções

### 2.1 Métodos e Ferramentas de Teste

- Blue-Green Deployment: Configuração de dois ambientes de produção (Blue e Green) para permitir a mudança do tráfego para o novo ambiente sem downtime. Se problemas forem detectados, o tráfego pode ser revertido rapidamente.
- Canary Releases: Implementação gradual da migração para um pequeno subconjunto de usuários antes de liberar para todos. Monitoramento da resposta e identificação de problemas potenciais em menor escala.
- Testes Automatizados: Uso de ferramentas como AWS CodePipeline e CodeBuild para automação de testes unitários, de integração e de carga. Garantia de que todos os componentes funcionem conforme esperado antes da migração completa.
- Monitoramento Pré-Migração e Contínuo: Configuração do CloudWatch para monitorar métricas de desempenho e logs em tempo real, permitindo a identificação de problemas imediatamente após o início da migração.

### 2.2 Minimização de Downtime

- Sincronização de Dados: Implementação de sincronização contínua entre o banco de dados on-premises e o RDS para garantir que não haja perda de dados durante a mudança.
- Replicação de DNS Gradual: Uso de TTL (Time-to-Live) baixo nas configurações de DNS para acelerar a propagação das mudanças e permitir a reversão rápida se necessário.
- Planos de Contingência: Preparação de um plano de rollback detalhado, incluindo backups completos antes da migração e estratégias para restaurar rapidamente os serviços antigos se necessário.

## 3. Benefícios e Otimização Pós-Migração

### 3.1 Benefícios Esperados

- Escalabilidade: Uso de Auto Scaling para ajustar dinamicamente a capacidade da aplicação com base na demanda, eliminando problemas de sobrecarga durante picos de tráfego.
- Alta Disponibilidade: Implementação de RDS Multi-AZ e ELB para garantir que a aplicação continue funcionando mesmo em caso de falha de um componente. Replicação de dados entre zonas de disponibilidade aumenta a resiliência.
- Desempenho Melhorado: Uso de CloudFront para distribuição global de conteúdo e redução da latência. Alocação eficiente de recursos conforme a necessidade da aplicação, utilizando instâncias otimizadas.
- Segurança: Ferramentas como AWS Identity and Access Management (IAM) para controle de acesso.
- Custo-Efetividade: Pagar apenas pelos recursos utilizados com modelos de preços flexíveis.

### **3.2 Otimização da Aplicação na AWS**

- Cost Optimization: Revisões regulares das instâncias EC2 em uso para garantir que o tipo e o tamanho estão adequados, evitando custos desnecessários. Uso de Reserved Instances ou Savings Plans para reduzir custos a longo prazo.
- Performance Tuning: Ajuste contínuo de parâmetros do RDS, caching utilizando ElastiCache e otimização de consultas para melhorar o tempo de resposta do banco de dados.
- Auto Scaling: Configurar políticas de auto scaling para lidar com picos de tráfego.
- Otimização de Rede: Configurar VPC e CloudFront para melhorar a latência e a entrega de conteúdo.
- Segurança Contínua: Implementação de práticas de segurança recomendadas, como rotinas de patching automatizadas, auditorias de segurança periódicas e criptografia em todos os níveis (em trânsito e em repouso).
- Acompanhamento e Melhoria Contínua: Uso do CloudWatch e AWS X-Ray para monitoramento detalhado da aplicação, análise de desempenho e identificação de gargalos. Implementação de uma estratégia de DevOps para automação contínua de deploys e testes.

## Conclusão

A migração de uma aplicação legado para a AWS exige planejamento detalhado e execução cuidadosa. Seguindo estas etapas, é possível garantir que a migração seja realizada com mínimo impacto aos usuários, enquanto se aproveita ao máximo as capacidades da nuvem para melhorar escalabilidade, disponibilidade e desempenho.