

# Documentação Técnica – Arquitetura Híbrida XPTO

---

## 1. Visão Geral do Projeto

A proposta visa modernizar o sistema de fluxo de caixa da empresa XPTO com uma solução híbrida, integrando a infraestrutura on-premises existente com recursos escaláveis na AWS. O objetivo principal é garantir **alta disponibilidade, escalabilidade, segurança, resiliência e eficiência de custos** com práticas de **FinOps** e **automação via IaC**.

---

## 2. Diagrama de Arquitetura

### Componentes da Solução

- Usuário final (navegador/app)
  - Internet / CDN (CloudFront)
  - ALB (Application Load Balancer)
  - EKS (Kubernetes) – serviços escaláveis
  - Lambda (Serverless) – tarefas assíncronas
  - RDS (Multi-AZ) – banco de dados gerenciado
  - S3 – backups, logs e arquivos estáticos
  - CloudWatch / Prometheus / Grafana – monitoramento e observabilidade
  - VPN / Direct Connect – integração com data center on-premises
  - VMs legadas – serviços de controle e histórico
- 

## 3. Dimensionamento de Recursos

### Serviço de Controle de Lançamentos

- Execução em pods Kubernetes (EKS)
- 2 vCPUs / 4 GB RAM por pod
- Auto Scaling até 5 réplicas

## Serviço de Consolidado Diário

- Executado como função AWS Lambda
  - 1 GB RAM / até 100 execuções simultâneas
  - Tolerância de até 5% de perda em picos de 50 RPS
- 

## 4. Plano de Disaster Recovery (DR)

- **RTO (Recovery Time Objective):** 10 minutos
  - **RPO (Recovery Point Objective):** 5 minutos
  - Backups automáticos no Amazon S3
  - Failover automático com RDS Multi-AZ
  - Testes semestrais de recuperação
- 

## 5. Estratégia FinOps

- Uso de instâncias Spot para cargas não críticas
  - Auto Scaling baseado em métrica de CPU e requisições
  - Tagueamento de recursos para rastreamento de custos por time/projeto
  - Monitoramento com AWS Budgets e CloudWatch
  - Scripts de desligamento de ambientes não produtivos em horário comercial
- 

## 6. Automação com Terraform e Ansible

- Provisionamento da VPC, sub-redes, RDS e EKS com **Terraform**
  - Configuração das VMs legadas on-prem com **Ansible**
  - CI/CD com **GitHub Actions** para validação de infraestrutura
  - Estrutura padronizada de repositório com versionamento
- 

## 7. Monitoramento e Observabilidade

- **Prometheus + Grafana:** métricas dos pods e serviços em EKS
  - **AWS CloudWatch:** métricas de Lambda, ALB e RDS
  - **SNS:** alertas em tempo real para incidentes
  - **Logs Insights:** análise de logs estruturados
-

## 8. Futuras Evoluções

- Adoção de **AWS Fargate** para workloads event-driven
  - Integração com **AWS Step Functions** para workflows complexos
  - Implementação de **GitOps** com ArgoCD
  - Uso de **Service Mesh (Istio)** para controle de tráfego
- 

## 9. Aplicação do Modelo OSI

- **Camadas 3 e 4:** Roteamento seguro com VPN/IPSec e TLS
  - **Camada 7:** Regras no ALB para distribuição por path/serviço
  - Diagnóstico de problemas com `traceroute`, `netstat`, `curl`, `tcpdump`
  - Monitoramento de tempo de resposta e disponibilidade por camada
- 

## 10. Scripts Terraform e Ansible

### Terraform

Provisionamento completo da infraestrutura na AWS, incluindo:

Hcl

```
module "vpc" {  
  source = "terraform-aws-modules/vpc/aws"  
  cidr   = "10.0.0.0/16"  
  azs    = ["us-east-1a", "us-east-1b"]  
  ...  
}
```

### Ansible

Playbook para configurar e preparar as VMs legadas:

yaml

```
- hosts: onprem  
  tasks:  
    - name: Instalar Docker  
      apt:  
        name: docker.io  
        state: present
```

---

## 11. Estrutura do Repositório

Bash

```
infra-hibrida-xpto/  
├── terraform/      # Infraestrutura AWS  
├── ansible/        # Configuração On-Prem  
├── diagrams/       # Diagramas da arquitetura  
├── docs/           # Documentação técnica  
├── .github/        # CI/CD com GitHub Actions  
└── README.md       # Instruções gerais
```

---

## 12. README Técnico do Projeto

Md

# Arquitetura Híbrida - Empresa XPTO

Este projeto propõe uma arquitetura híbrida moderna e resiliente para suportar os serviços de controle financeiro da empresa XPTO. Utiliza infraestrutura on-premises integrada à AWS, garantindo escalabilidade, segurança e economia.

## Tecnologias

- AWS (EKS, Lambda, RDS)
- Terraform, Ansible
- Kubernetes
- GitHub Actions (CI/CD)