DevOps

Deploy Contínuo e Infraestrutura em Nuvem

Estratégias de Deploy

Por que precisamos de estratégias especiais de deploy?

“No mundo real, não podemos simplesmente desligar o sistema para atualizá-lo. Precisamos de métodos para minimizar riscos e impacto.”

Desafios:

- Expectativa de disponibilidade contínua (24/7);
- Complexidade de dependências e integrações;
- Impacto financeiro e reputacional de falhas;
- Necessidades de validação em ambiente real.

Estratégias de Deploy

Métricas de negócio afetadas por problemas em deploys:

- Downtime e custo por minuto;
- Perda de transações;
- Impacto na experiência do usuário e NPS;
- Custo de rollback e recuperação.

Estratégias de Deploy

Estimativa de custo de downtime para e-commerces (Mercado Livre, Amazon, Shopee, Google Shopping...)

Tempo de downtime	Custo estimado
1 minuto	R$ 15.000 - R$ 50.000
10 minutos	R$ 150.000 - R$ 500.000
30 minutos	R$ 450.000 - R$ 1.500.000
1 hora	R$ 900.000 - R$ 3.000.000
24 horas	R$ 21.600.000 - R$ 72.000.000
48 horas	R$ 43.200.000 - R$ 144.000.000

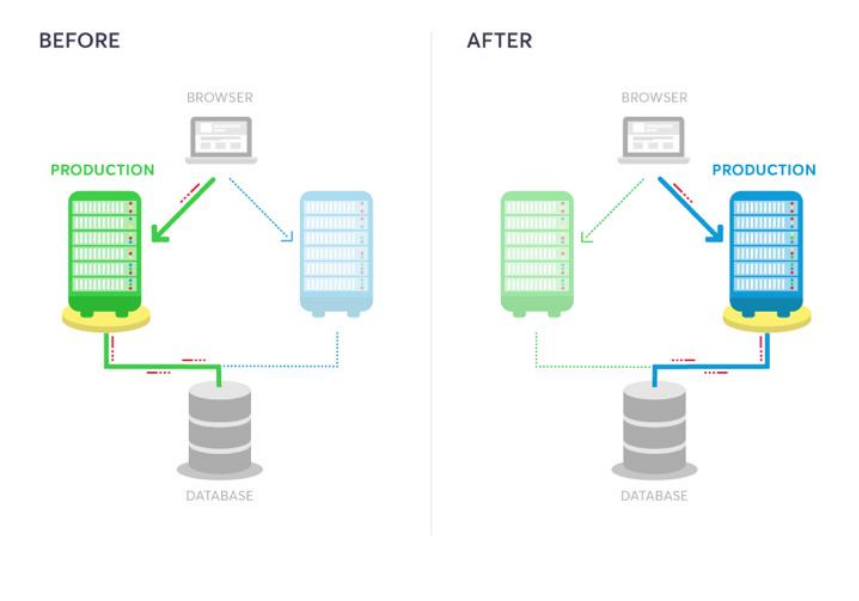
Estratégias de Deploy

Estimativa de custo de downtime para bancos (NuBank, Inter, Itaú, Banco do Brasil, Caixa Econômica...)

Tempo de downtime	Custo estimado
1 minuto	R$ 300.000 - R$ 1.000.000
10 minutos	R$ 3.000.000 - R$ 10.000.000
30 minutos	R$ 9.000.000 - R$ 30.000.000
1 hora	R$ 18.000.000 - R$ 60.000.000
24 horas	R$ 430.000.000 - R$ 1.440.000.000
48 horas	R$ 860.000.000 - R$ 2.880.000.000

Blue-Green Deployment

“Blue-Green é como ter um ambiente reserva pronto para assumir a qualquer momento”



Blue-Green Deployment

Vantagens:

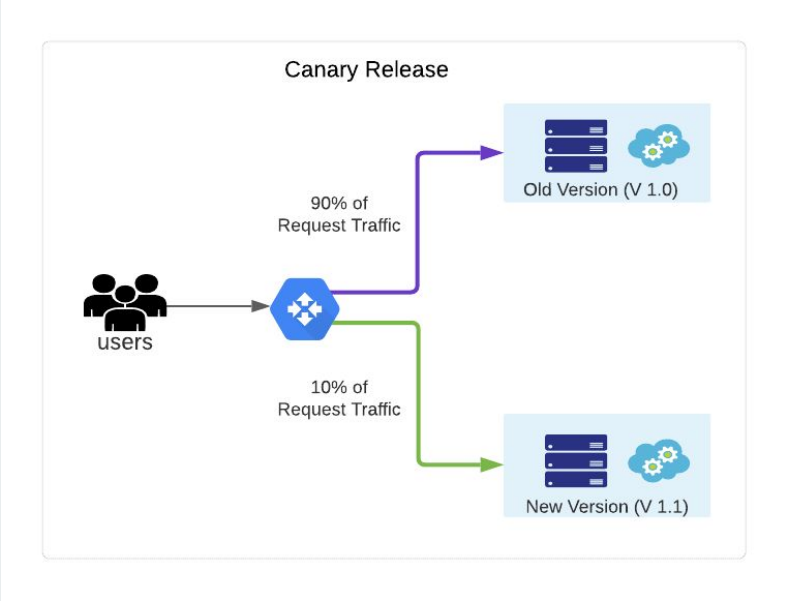
- Zero downtime durante a transição;
- Rollback instantâneo se necessário;
- Possibilidade de testes em ambiente idêntico a produção.

Desafio:

- Duplicação de recursos (custos);
- Sincronização de dados e estado;
- Compatibilidade de banco de dados.

Canary Releases

“Canary Releases são como enviar exploradores antes do exército - um pequeno grupo testa o terreno”



Canary Releases

Métricas críticas para monitorar:

- Taxa de erro;
- Latência de resposta;
- Uso de recursos;
- Métricas de negócio (conversões, etc).

Ferramentas que facilitam:

- Spinnaker;
- Argo Rollouts;
- AWS AppMesh;
- Google Cloud Traffic Director.

Feature Flags

“Feature Flags transforman deployments em eventos sem cerimonia, separando deploy de release”

```
// Javascript exemplo

if (featureFlags.isEnabled('new-checkout-flow', user)) {

  // Novo fluxo de checkout

  showNewCheckout();

} else {

  // Fluxo antigo

  showOldCheckout();

}
```

Feature Flags

Tipos de feature flags:

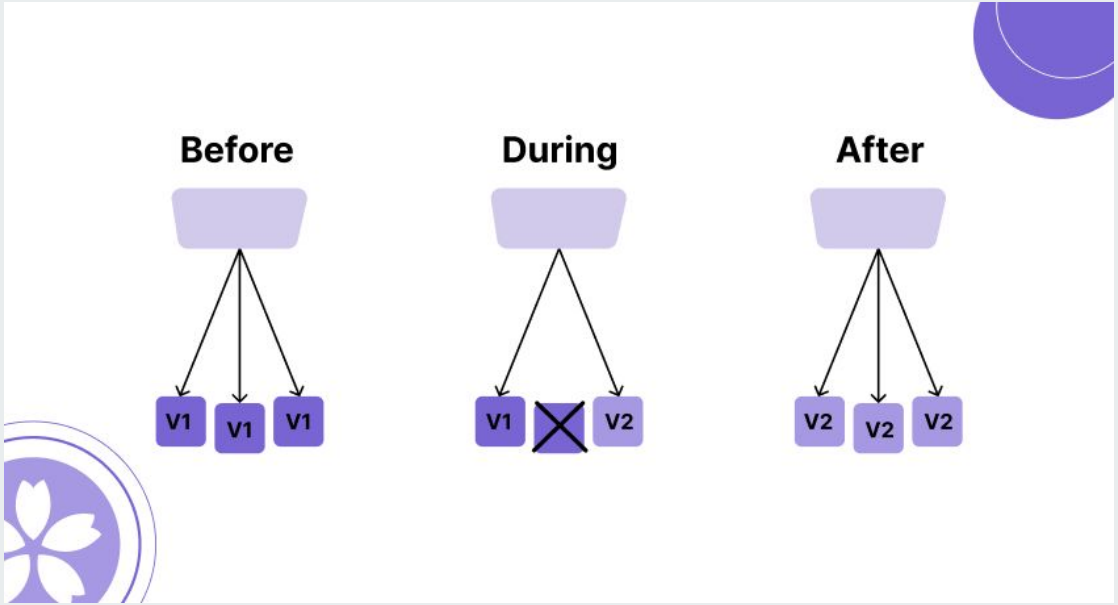
- Release flags: habilitar/desabilitar features;
- Ops flags: controles operacionais (throttling, circuit breakers);
- Permissioning flags: controle de acesso;
- Experiment flags: testes A/B.

Ferramentas que facilitam:

- LaunchDarkly;
- Split.io;
- Flagsmith.

Rolling Updates

“Rolling updates são como trocar as rodas do carro enquanto ele está em movimento”

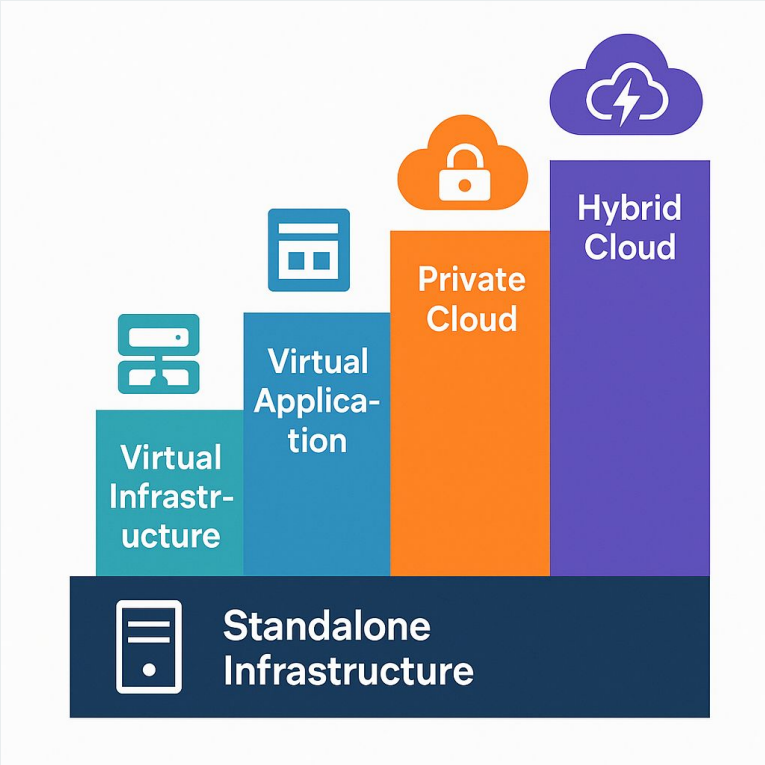


Rolling Updates

Considerações importantes:

- Health checks robustos;
- Compatibilidade entre versões;
- Monitoramento durante rollout.

Infraestrutura em Nuvem



Standalone Infrastructure (Infraestrutura Isolada)

O que é:
É o modelo tradicional de TI, em que servidores, redes, armazenamento e outros recursos físicos estão instalados **localmente** (on-premises), geralmente dentro da própria empresa.

Características:

- Altos custos com compra e manutenção de hardware;
- Baixa escalabilidade (crescimento lento e caro);
- Gerenciamento centralizado e totalmente local;
- Riscos maiores de downtime por falhas físicas.

Exemplo:
Um servidor físico em um data center da empresa rodando um sistema ERP.

Virtual Infrastructure (Infraestrutura Virtualizada)

O que é:
É o uso de **virtualização** para criar múltiplas máquinas virtuais (VMs) em um único servidor físico, utilizando softwares como VMware, Hyper-V ou KVM.

Características:

- Melhor aproveitamento do hardware;
- Isolamento entre aplicações e ambientes;
- Facilidade de gerenciamento e backup;
- Ainda depende de data centers locais, mas com mais flexibilidade.

Exemplo:
Um servidor rodando várias VMs para diferentes departamentos da empresa.

Virtual Application (Aplicações Virtualizadas)

O que é:
Neste estágio, não se virtualiza apenas o servidor, mas a própria aplicação. Isso pode incluir uso de **containers (ex: Docker)** ou **aplicações em ambientes virtuais gerenciados**.

Características:

- Redução de dependência da infraestrutura subjacente;
- Mais portabilidade (roda em qualquer ambiente com suporte ao container);
- Melhor uso de recursos, comparado a VMs;
- Mais agilidade em desenvolvimento e entrega (DevOps).

Exemplo:
Uma aplicação web empacotada em um container Docker e orquestrada por Kubernetes.

Private Cloud (Nuvem Privada)

O que é:
Uma **nuvem construída para uso exclusivo de uma organização**, podendo estar hospedada localmente ou em provedores terceiros. Usa conceitos de nuvem (provisionamento automático, elasticidade, autosserviço), mas com controle privado.

Características:

- Alto nível de controle e segurança;
- Ideal para empresas com requisitos regulatórios.;
- Pode usar ferramentas como OpenStack ou VMware vCloud;
- Mais caro que nuvens públicas.

Exemplo:
Um banco que gerencia seu próprio data center com tecnologias de nuvem para atender às normas de segurança.

Hybrid Cloud (Nuvem Híbrida)

O que é:
Combinação de nuvem privada com nuvens públicas (ex: AWS, Azure, Google Cloud), permitindo que workloads sejam distribuídos entre os ambientes conforme a necessidade.

Características:

- Equilíbrio entre segurança, performance e escalabilidade;
- Permite mover dados e aplicações entre ambientes;
- Ideal para empresas que precisam atender a diferentes exigências de compliance e performance;
- Desafios com integração, gerenciamento e segurança.

Exemplo:
Uma empresa roda aplicações críticas em sua nuvem privada, mas usa a nuvem pública para lidar com picos de demanda.

Infraestrutura em Nuvem

Limitações:

- Capacidade fixa (sub ou superdimensionada);
- Alto custo inicial;
- Longo tempo para expansão
- Manutenção física e substituição de hardware

Transição:

- Elasticidade;
- Redução de custos;
- Foco em desenvolvimento x gerenciamento de infraestrutura.

Modelos de serviço em nuvem

- Iaas - Infrastructure as a Service
  - é um modelo de computação em nuvem onde você **aluga recursos de infraestrutura de TI** — como servidores, redes, armazenamento e sistemas operacionais — de um provedor de nuvem.
- Paas - Platform as a Service
  - é um modelo de computação em nuvem que fornece uma **plataforma completa para desenvolver, executar e gerenciar aplicações** — sem que você precise se preocupar com a infraestrutura subjacente (servidores, redes, armazenamento, sistema operacional, etc.).
- Saas - Software as a Service
  - é um modelo de computação em nuvem em que o usuário acessa **um software pronto, via internet**, sem precisar instalar, manter ou atualizar nada.
- Faas - Function as a Service
  - é um modelo de computação em nuvem que permite executar **funções específicas sob demanda**, **sem precisar gerenciar servidores ou infraestrutura**.

Iaas - Infrastructure as a Service

"IaaS é como ter seus servidores sem precisar gerenciar o hardware físico."

Principais características:

- Controle total sobre sistema operacional e software;
- Responsabilidade sobre patches, atualizações, segurança;
- Flexibilidade máxima com maior responsabilidade.

Casos de uso ideais:

- Migração lift-and-shift;
- Aplicações que exigem configurações específicas de SO;
- Quando se precisa de controle total sobre o ambiente.

Exemplos: AWS EC2, Azure VMs, Google Compute Engine

Paas - Platform as a Service

"PaaS é como um playground pronto para seus desenvolvedores, onde você só se preocupa com seu código."

Principais características:

- Plataforma pré-configurada para desenvolvimento;
- Runtime, middleware e OS gerenciados pelo provedor;
- Foco no código da aplicação.

Casos de uso ideais:

- Startups e times com recursos limitados;
- Desenvolvimento rápido de aplicações;
- Quando equipes de desenvolvimento precisam de agilidade.

Exemplos: Heroku, Google App Engine, Azure App Service, AWS Elastic Beanstalk

Saas - Software as a Service

"SaaS permite que você use ferramentas sofisticadas sem a complexidade de instalação e manutenção."

Principais características:

- Software completo entregue via web;
- Sem instalação local ou gerenciamento de servidor;
- Tipicamente baseado em assinatura.

Faas - Function as a Service

"Serverless leva a abstração ao extremo - você só paga pelo tempo de execução real do seu código."

Principais características:

- Execução de código sem gerenciar servidores;
- Escalabilidade automática e transparente;
- Cobrança por tempo de execução (pay-per-use).

Casos de uso ideais:

- Processamento assíncrono;
- APIs com tráfego variável;
- Automações e integrações;
- ETL e processamento de dados.

Exemplos: AWS Lambda, Azure Functions, Google Cloud Functions

Provedores de nuvem

aws

Azure

Google Cloud

IBM Cloud

ORACLE
Cloud Infrastructure

HETZNER
SERVER · CLOUD · HOSTING

DigitalOcean

Provedores de nuvem

Categoria	AWS	Azure	GCP
Computação (VMs)	EC2 (Elastic Compute Cloud)	Azure Virtual Machines	Compute Engine
Execução de containers	ECS / EKS (Kubernetes)	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)
Funções (FaaS)	AWS Lambda	Azure Functions	Cloud Functions
Armazenamento de Objetos	S3 (Simple Storage Service)	Azure Blob Storage	Cloud Storage
Banco de Dados Relacional (SQL)	Amazon RDS	Azure SQL Database	Cloud SQL
Banco de Dados Não Relacional (NoSQL)	DynamoDB	Azure Cosmos DB	Firestore / Datastore
Cache (Redis/Memcached)	ElastiCache	Azure Cache for Redis	Memorystore

Provedores de nuvem

Foco estratégico

- AWS
  - Maior variedade de serviços, pioneira em cloud, robustez e escalabilidade
- Azure
  - Integração nativa com Microsoft
- GCP
  - Big Data, IA/ML e performance em containers com Kubernetes (GKE)

Provedores de nuvem

Quando escolher cada um?

- **AWS**: quando se precisa de **ampla gama de serviços** e flexibilidade total.
- **Azure**: ideal se sua empresa já usa **tecnologias Microsoft**.
- **GCP**: ótimo para projetos com foco em **dados, IA e containers**.