**🧩 AWS CloudFormation — Implementando sua Primeira Stack**

**🧠 Conceitos-Chave**

* **CloudFormation** = Infraestrutura como Código (IaC) da AWS.
* Permite criar, atualizar e excluir recursos AWS automaticamente via **template** (YAML/JSON).
* **Stack** = Conjunto de recursos criados e gerenciados juntos a partir de um template.

**⚙️ Etapas da Implementação**

1. **Criar o Template**
   * Define os recursos (ex: EC2, S3, VPC).
   * Exemplo simples:
   * Resources:
   * MeuBucketS3:
   * Type: AWS::S3::Bucket
   * Properties:
   * BucketName: meu-primeiro-bucket
2. **Fazer Upload no CloudFormation**
   * Console → *Create Stack* → *Upload a template file*.
3. **Configurar e Criar a Stack**
   * Nomear Stack, definir permissões e confirmar.
4. **Acompanhar o Processo**
   * Status: *CREATE\_IN\_PROGRESS* → *CREATE\_COMPLETE*.
   * Ver logs se ocorrer falha.

**💡 Insights Importantes**

* **Automação Total**: elimina configurações manuais.
* **Consistência**: ambientes iguais em diferentes regiões/contas.
* **Reutilização**: templates podem ser replicados.
* **Rollback automático** em caso de falha.
* **Idempotência**: recriação garante ambiente consistente.

**🪄 Boas Práticas**

* Validar o template (validate-template) antes de criar.
* Usar **parâmetros e outputs** para deixar o template dinâmico.
* Armazenar templates em **controle de versão (Git)**.
* Nomear recursos de forma padronizada.
* Evitar credenciais no template (use Secrets Manager).

**⚠️ Erros Comuns**

* Nomes duplicados (principalmente buckets S3).
* Falta de permissões IAM.
* Sintaxe incorreta no YAML/JSON.

**🚀 Resumo Final**

* CloudFormation simplifica a criação e manutenção da infraestrutura.
* Uma Stack representa **a base de um ambiente completo**.
* Dominar o processo é essencial para práticas **DevOps e automação na AWS**.