# Aula de aws-cloudformation Anotações

**Anotações realizadas durante a aula**

* **CloudFormation** permite criar e gerenciar recursos da AWS de forma declarativa, usando **templates** em **YAML** ou **JSON**.
* Uma **stack** é a instância de um template, ou seja, o conjunto de recursos que são criados, atualizados ou excluídos juntos.
* É possível versionar e reutilizar templates para garantir consistência na infraestrutura.
* As alterações em uma stack seguem o conceito de **Change Set**, que mostra o que será modificado antes de aplicar.
* Erros em uma criação ou atualização podem ser revertidos automaticamente, evitando que a infraestrutura fique em um estado inconsistente.

**Insights de implementação que entendi**

* **Modularização**: dividir a infraestrutura em múltiplos templates (nested stacks) facilita manutenção e reuso.
* **Parâmetros**: permitem flexibilizar o template, tornando-o adaptável a diferentes ambientes (dev, homologação, produção).
* **Saídas (Outputs)**: úteis para expor informações de recursos criados (ex: URL de um Load Balancer ou ID de uma VPC).
* **Controle de dependências**: o CloudFormation entende a ordem de criação de recursos com base nas referências definidas no template.
* **Boas práticas de segurança**: usar **IAM Roles** específicos para stacks e limitar permissões ao mínimo necessário.
* **Integração com CI/CD**: CloudFormation pode ser usado junto com AWS CodePipeline e CodeBuild para provisionamento automático da infraestrutura.
* **Rollback**: sempre planejar rollback automático em caso de falha na criação da stack, reduzindo riscos de custos inesperados.
* **Stack Policies**: ajudam a proteger recursos críticos de atualizações acidentais.