**Plataforma AWS**

A solução aws destaca-se como as melhores práticas de qualidade em seus produtos, tendo constantes atualizações de segurança, o que é um ponto primordial para empresas que se preocupam com a segurança digital em suas operações.

Sendo assim, a Amazon Web Service proporciona um suporte 24 hs junto a especialistas e, além disso, também carrega consigo um firewall que limita acesso e aumenta o controle de fluxo, disponibiliza autenticação com múltiplos fatores de segurança e armazenamento de dados criptografados.

Serviços utilizados:

* AWS Codedeploy
* AWS CodePipeline
* Amazon Aurora MySql Compatible

**Etapas para Deploy**

* Para o processo de deploy, você deverá estar logado em sua conta na AWS e ter configurado o “Load Balancing” (um tipo de máquina virtual que permite aumentar a disponibilidade e desempenho evitando assim, gargalos de tráfego além de eliminar a possibilidade de um único ponto de falha).
* Feito isso, configurar o “Auto Scaling” . Clicar em: “Create Launch Configuration” (processo em que criará as instâncias, lembrando que para utilização do GITHUB será necessário utilizar EC2), escolher um nome e ainda nesta etapa, é necessário registrar o script com as dependências necessárias para execução da aplicação.
* Após a finalizar a criação da “Launch Configuration”, será necessário a criação da “Auto Scaling Group”. Optar por “Launch Configuration” , selecionar a “Launch” criada no processo anterior e marcar o uso de “Load Balancing" selecionando a do primeiro processo. Assim, todas as instâncias criadas neste grupo, ficarão no mesmo “Load Balancing". Escolher um nome para o “Auto Scaling Group” e clicar em create. Após isso, será criado a primeira instância.
* Para iniciarmos o Deploy, utilizaremos o “CODEDEPLOY”. Este aplicativo baseia-se no arquivo “appspec.yaml” que serve para mostrar “da onde eu quero copiar” e “para onde eu quero levar”.
* Na plataforma AWS, na busca, procurar por “codedeploy” , após isso, clicar em “criar um aplicativo”, escolher um nome e após isso, escolher a plataforma (clicar em EC2 que são as instâncias). Dentro deste aplicativo, criar um grupo de implantação (que são os servidores), escolher um nome e escolher a função “codedeploy” , na parte de “configuração de ambiente”, selecionar a opção “Grupos de Auto Scaling do Amazon EC2” e clicar em “Criar Grupo de Implantação”.
* Feito esses processos, integraremos o Github ao deploy e para isso, utilizaremos o Pipeline, ferramenta responsável pela automação do processo de deploy quando alterado o código da aplicação.
* Nesta etapa, clicar em Pipeline (CodePipeline), clicar em “Criar novo Pipeline”, escolher um nome, clicar em próximo e escolher a origem, neste caso, Github. Ao selecionar esta opção, será necessário digitar seu usuário e senha (caso ainda não possua conta, será necessário criar através do site: [www.github.com](http://www.github.com)). Após se logar com o Github, escolher o repositório que será utilizado, escolher a ramificação (master ou outra), selecionar a opção “Webhooks do Github” (o Github comunica quando houver alteração), clicar em próximo, a parte de compilação, pode-se clicar em “Ignorar etapa de compilação”, após isso, escolher o provedor de implantação, neste caso, optamos por “AWS CodeDeploy” , selecionar o nome do aplicativo criado anteriormente e também o nome do grupo de implantação, selecionar "próximo" e clicar em “Criar Pipeline”.
* Feito isso, a aplicação já estará rodando e qualquer alteração que seja realizada no código, após os “push” de commit, as alterações serão automáticas na aplicação.