

# Desafio - Machine Learning Platform Engineer

O desafio consiste em implementar uma arquitetura completa que consome a [Punk Api](https://api.punkapi.com/v2/beers/random) no endpoint `https://api.punkapi.com/v2/beers/random` e ingere em um `Kinesis Stream` que terá 2 consumidores. Para isso você precisará configurar:

1. Um `CloudWatch Event` que dispara a cada 5 minutos uma função `Lambda` para alimentar o `Kinesis Stream` que terá como saída:
  - Um `Firehose` agregando todas as entradas para guardar em um bucket `S3` com o nome de `raw`.
  - Outro `Firehose` com um `Data Transformation` que pega somente os `id`, `name`, `abv`, `ibu`, `target_fg`, `target_og`, `ebc`, `srm` e `ph` das cervejas e guarda em um outro bucket `S3` com o nome de `cleaned` em formato `csv`.
2. Crie uma tabela com os dados do bucket `cleaned`.
3. Com base nos dados da tabela `cleaned`, treine um modelo de machine learning que classifique as cervejas em seus respectivos ibus.

## Execução do Desafio

Siga os passos abaixo para a entrega do desafio, entregue tantos itens quanto puder.

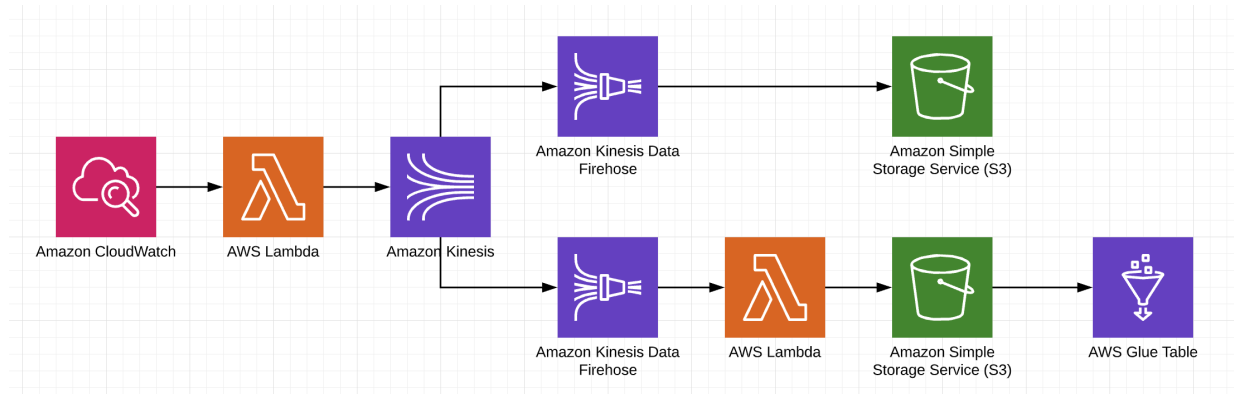
1. Criar uma conta gratuita na `AWS`.
2. Você deve utilizar `Terraform` para construir a arquitetura de uma maneira reproduzível em outras contas.
3. Todas as funções `Lambdas` devem ser desenvolvidas em `Python` assim como o modelo de machine learning.
4. O modelo de machine learning deve ser apresentado em um `Jupyter Notebook`, local ou remoto. O arquivo do notebook deve estar no repositório do `github`!
5. Bônus (não obrigatório): Integre o modelo de machine learning em sua arquitetura. Esta etapa é livre, fique a vontade para adorar a solução que preferir! Mas não esqueça de que precisará justificá-la!

## Entrega

A entrega deverá ocorrer em 2 etapas:

1. Entrega do código via `Github`.
2. Uma vídeo chamada para explicar e defender a solução feita.

## Desenho da Arquitetura



## Observações:

- Você terá 10 dias corridos a partir do recebimento deste email para fazer a entrega final via [Github](#), em um repositório público e o link do repositório deverá ser enviado para o email [data.engineering@picpay.com](mailto:data.engineering@picpay.com). Use o título da sua vaga no **assunto** do email.
- Durante todo o período o time estará disponível para dúvidas no email [data.engineering@picpay.com](mailto:data.engineering@picpay.com).
- O foco do teste é descobrir como você lida ao aprender tecnologias que nunca viu/trabalhou antes para fazer uma entrega.
- Caso não consiga terminar 100% do proposto, recomendamos que faça as entregas mesmo assim para que o time possa avaliar seu desempenho.
- Não esqueça de **desligar a sua infraestrutura** para não gerar *cobrança*.

## Referências:

- <https://registry.terraform.io/browse?provider=aws>
- <https://serverless.com/examples/aws-java-simple-http-endpoint/>
- <https://www.youtube.com/watch?v=wgzgVm7Sqlk>
- <https://www.youtube.com/watch?v=5WykrpB7qS4>
- <https://www.youtube.com/watch?v=7mBo6pT09RM>