

Ambiente Exploratório e Operacional de Ciência de Dados

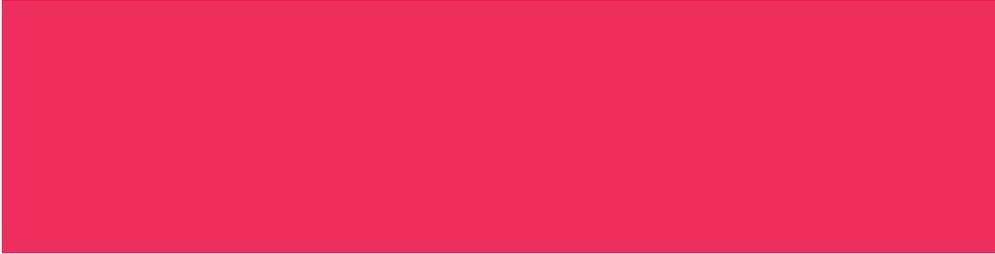


1. Check in

Nome:

Área:

Expectativa:



2. Contexto

Este material foi separado em **2 partes**, para que ficasse bem claro cada item e o pipeline de cada ambiente: **exploratório** e **operacional**.

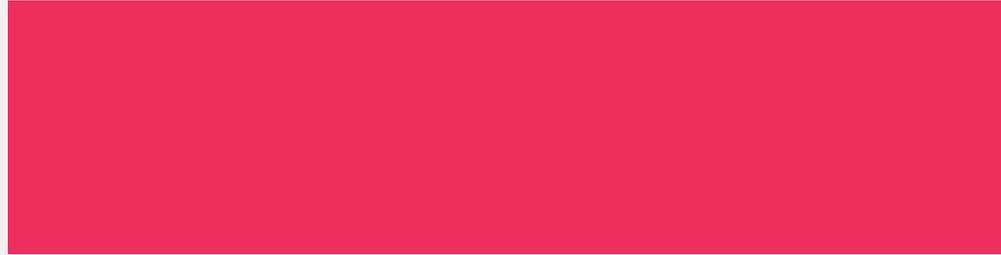


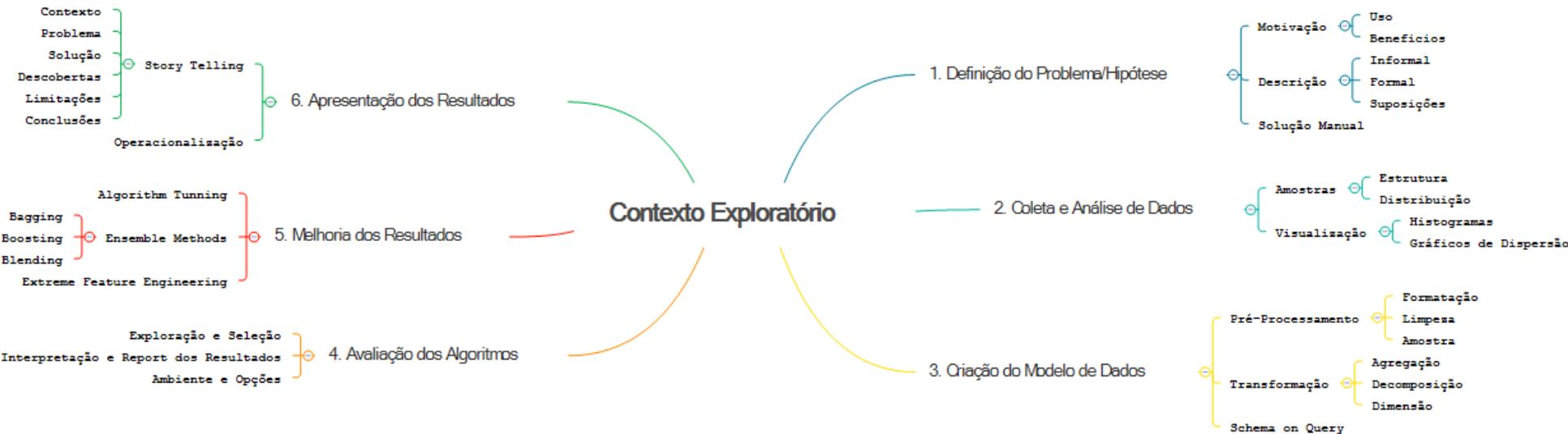
Premissas

- Todos o ambiente foi baseado em serviços e recursos disponíveis na região **South America(São Paulo) sa-east-1**, da AWS;
- Necessidade **mínima** de engenharia de dados;
- Ambiente baseado em Python;
- Primeiramente concebido em um cluster compartilhado



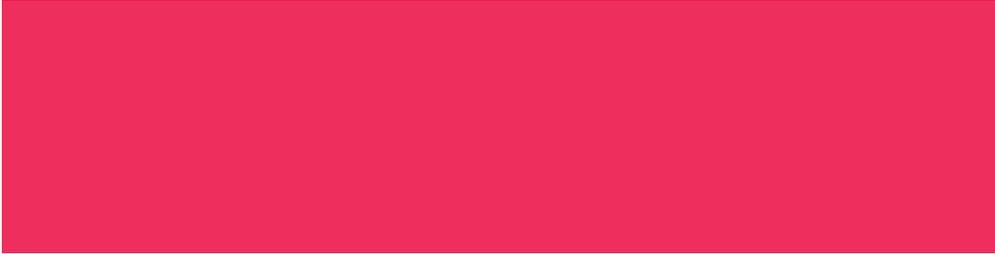
3. Exploração

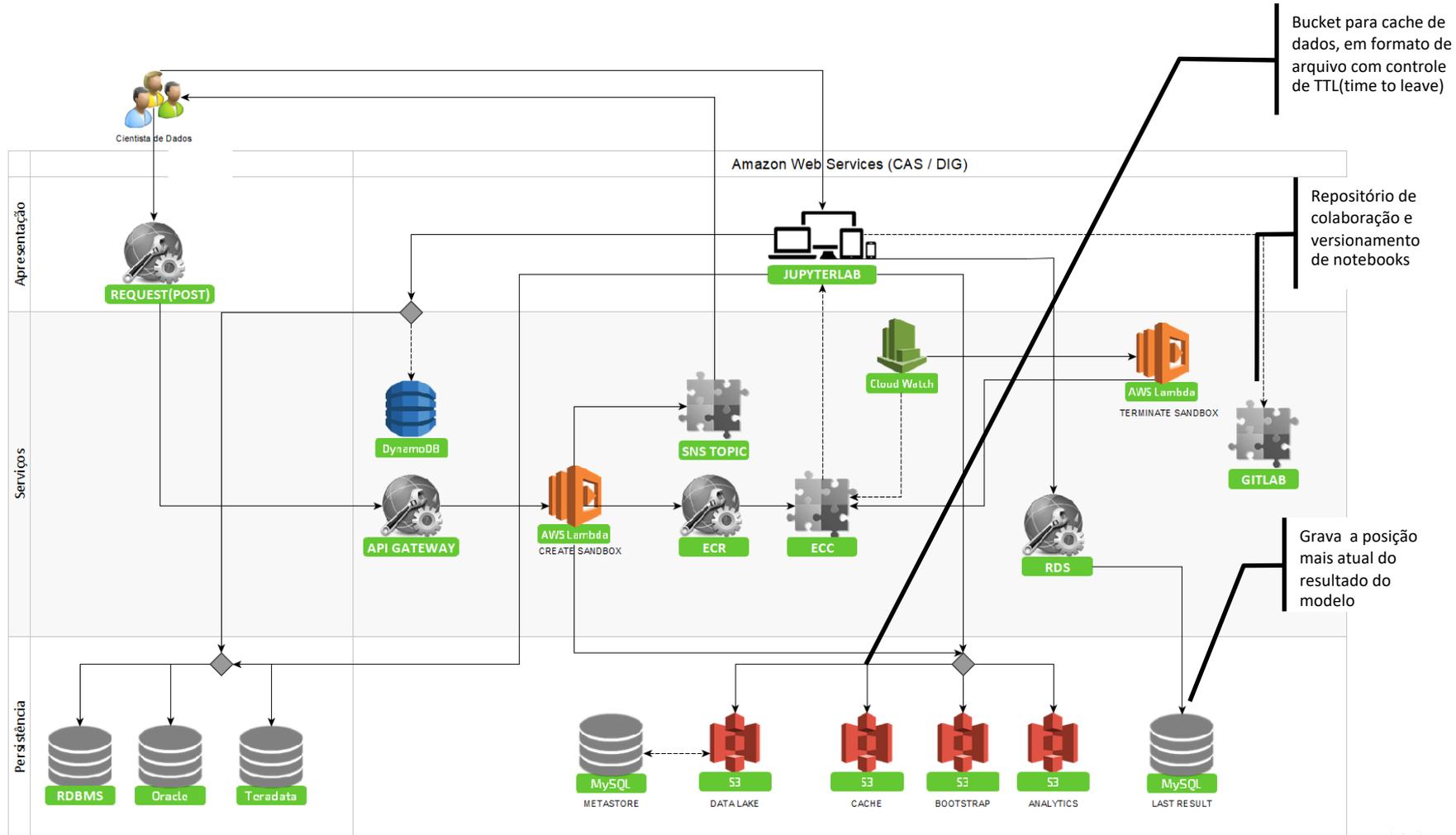






4. Blueprint da Solução



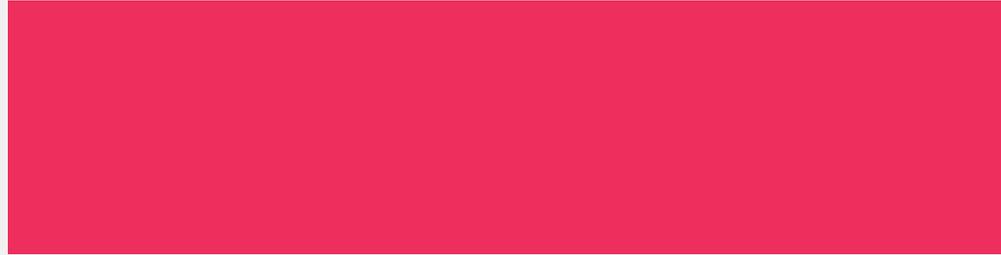


Propósito e Benefícios

- Armazenados em seu formato natural
- Fácil adaptação a mudanças
- Acessível para os diferentes perfis de usuários: heavy e soft users, cientista de dados, etc.
- Insights rápidos
- Alta disponibilidade e escalabilidade
- Entrega de valor desde o início
- Mobiliza a curiosidade constante
- Promove um ambiente único para: exploração, experimentação e colaboração de modelos;
- Habilita a utilização de diversas fontes de dados. Exemplo: Oracle, MySQL, MSSQLServer, S3, etc.
- Flexibiliza o uso de várias linguagens, no mesmo notebook. Exemplo: Python, SQL, Shell, etc.
- Aumenta a eficiência operacional;
- Diminui a necessidade de engenharia de dados;
- Documentação do modelo em um formato de storytelling.



5. Operação



NOTEBOOK

<CODE>

BUILD

REPO

```

Zeppelin Notebook - Job
...s/UPSELL_CART_ADICIONAL_DEPLOY

%python
%spark.pyspark

import gc

from cql import CQL
cql = CQL.CQLContext(spark=spark, sqlContext=sqlContext, zeppelin = z, region_name="sa-east-1")

def consultaz(query):
    cql.query(query)
    gc.collect()

def consulta(query):
    var = cql.query(query, False)
    gc.collect()
    return var
  
```

```

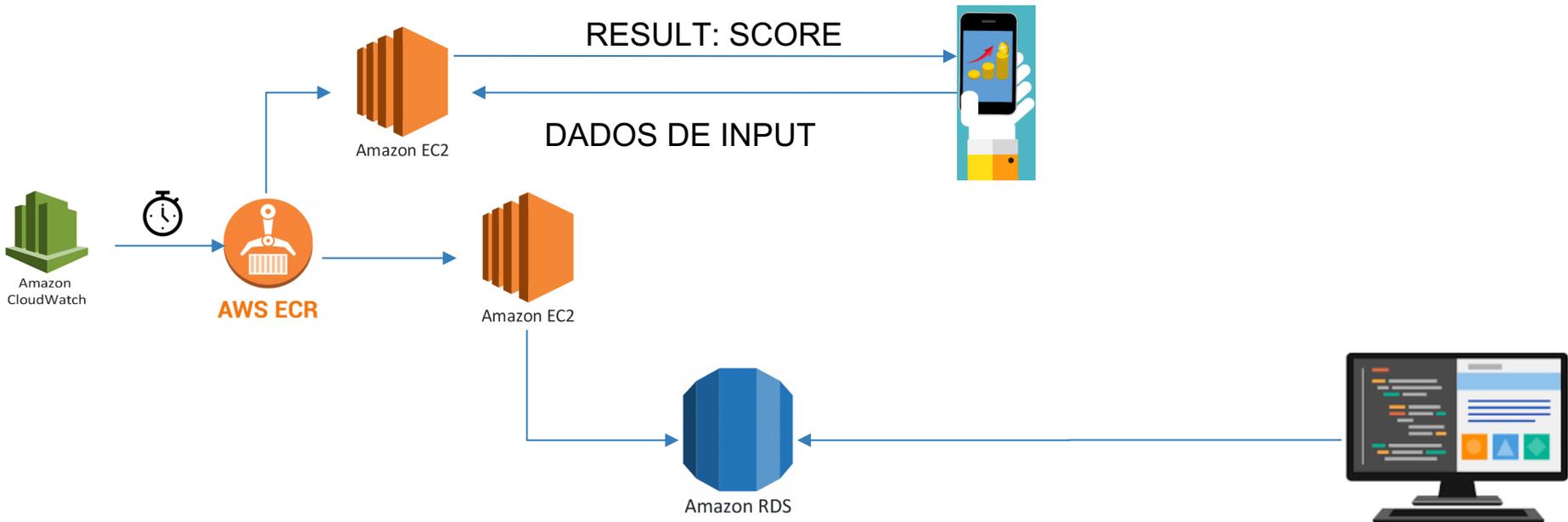
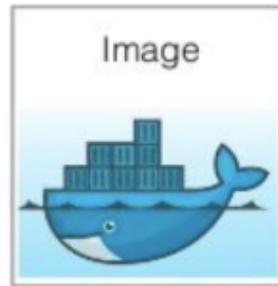
python
%spark.pyspark

import gc

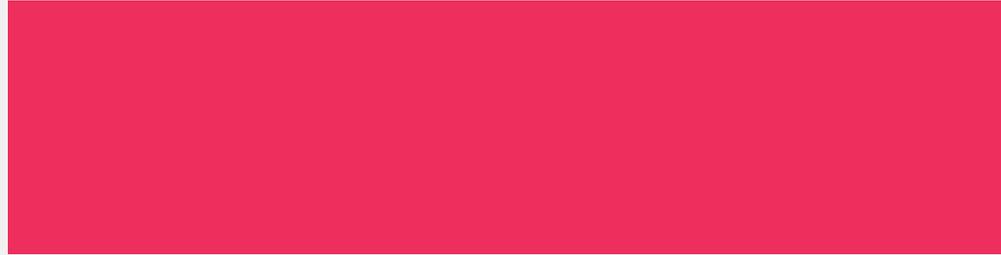
from cql import CQL
cql = CQL.CQLContext(spark=spark, sqlContext=sqlContext, zeppelin = z, region_name="sa-east-1")

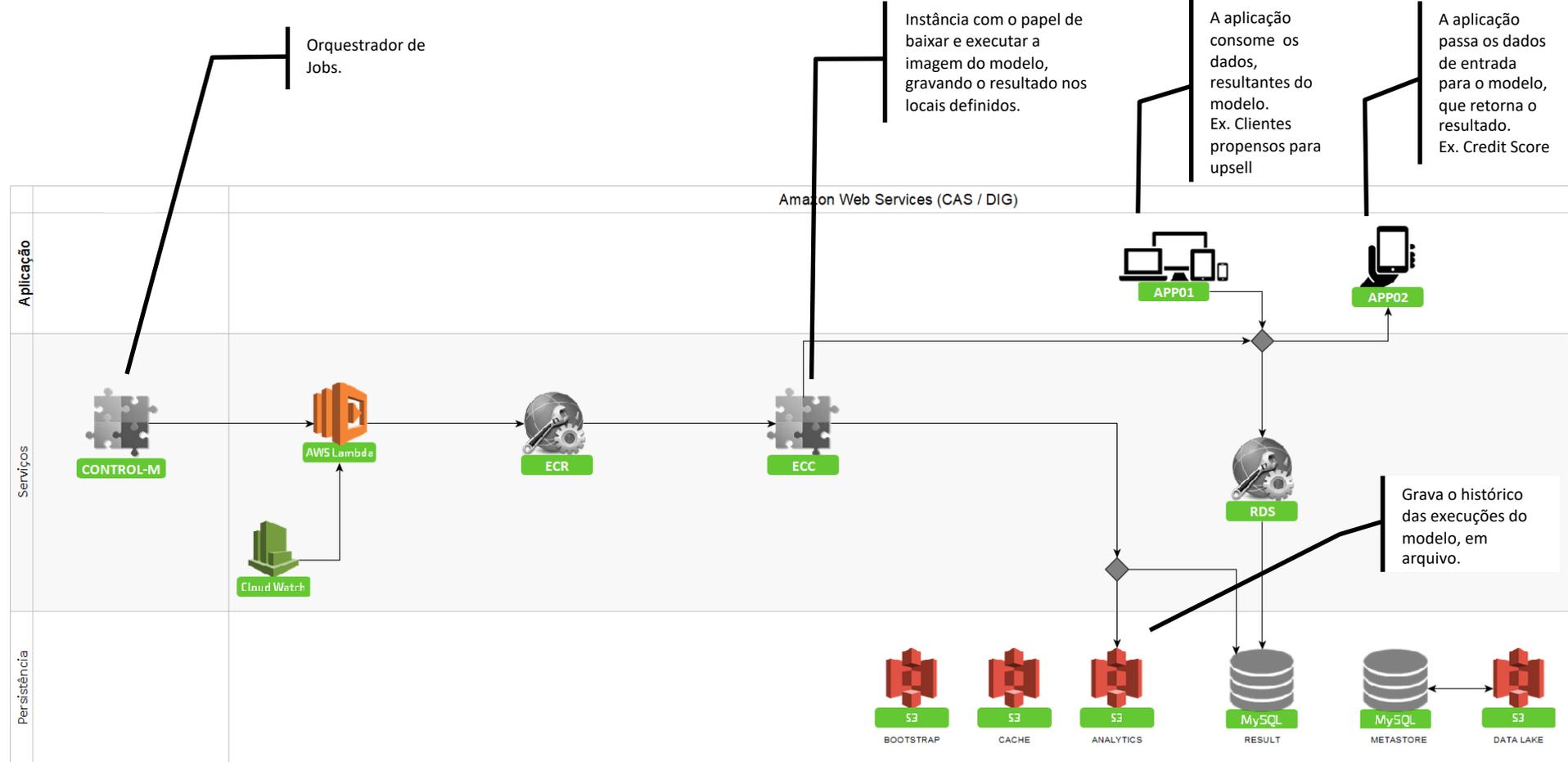
def consultaz(query):
    cql.query(query)
    gc.collect()

def consulta(query):
    var = cql.query(query, False)
    gc.collect()
    return var
  
```



6. Blueprint da Solução

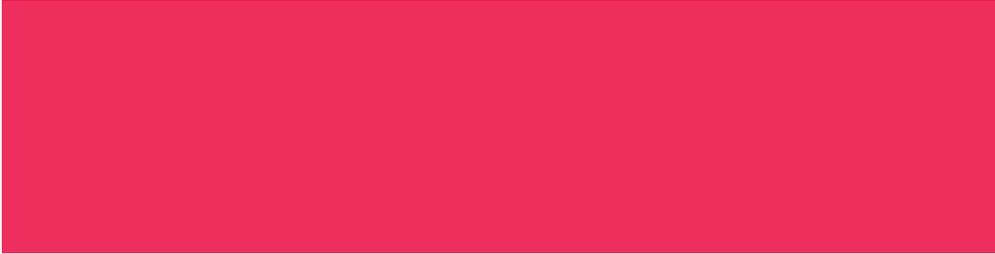




Propósito e Benefícios

-
- Controle de versionamento, ciclo de vida e deploy dos modelos;
 - Escalabilidade;
 - Redução de custos;
 - Integração entre os serviços aws;
 - Fácil consumo do resultado pelas aplicações;
 - Novas oportunidades de negocio





7. Ongoing

- A **avaliação de desempenho** será feita após a conclusão dos modelos migrados;
- O fato de **atualizar os modelos** (reescrita) já trouxe um grande ganho para o negocio, gerando novas oportunidades.
Exemplo: A criação do modelo de upsell do Produto XPTO, criou a oportunidade da criação de outro modelo para cross sell de outros produtos, não identificados antes
- A **avaliação do resultado \$\$\$**, será após 2-4 meses de operação de cada modelo e o acompanhamento é feito pelo time em um painel da area;
- **Benefícios** (Cloud x SAS):
 - Diversas opções de algoritmos e tipos de modelagem
 - Transformações e Pré-Processamentos são mais flexíveis
 - Comunidade Ativa
- Com a apresentação deste trabalho, pelo time que utiliza o ambiente, foi perguntando como foi feito, gerando uma novidade e novas frentes e clientes internos para se beneficiar do ambiente e o que ele habilita. Com isso, será elencamos varias temas para algumas POCs com estes times. o
- Projeto implantado em 6 meses, gerando resultado desde o primeiro mês.
- Tivemos um total de 8 Modelos (finalizados ou em andamento):
 - 4 Modelos construídos no novo ambiente (Python)
 - 4 Modelos reescritos (SAS -> Python)



Conclusões

- Qual onda sua empresa/negócio está surfando?
- Identificar o nível de maturidade para este tema;
- Consigo adaptar o conteúdo visto a minha realidade?
- Qual a 1ª ação deve ser tomada? Quais são os próximos passos?
- Transformem o que falamos em ações praticas, ou em um plano de curto-médio prazo, comece a gerar valor logo de início com pequenas quick wins!!!

Depois me contem o resultado 😊





Obrigado!

Ronaldo Bernardi
Head of Business Intelligence na Warren Brasil

Contatos:

E-mail: ronaldo.bernardi@gmail.com

Linkedin: www.linkedin.com/in/ronaldo-bernardi

Café? Só chamar ☺

warrenbrasil.com.br