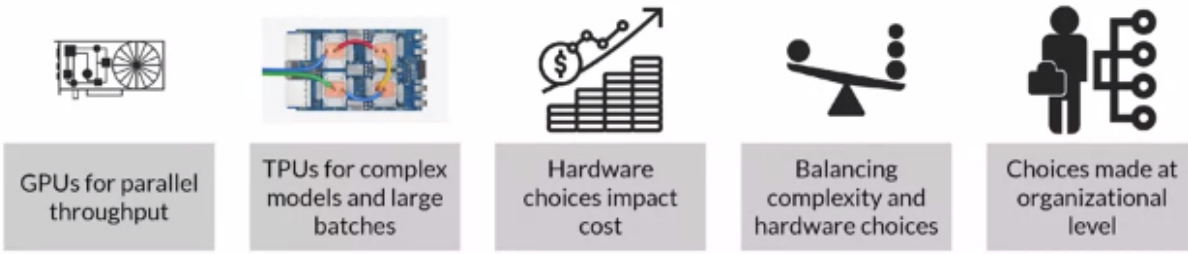
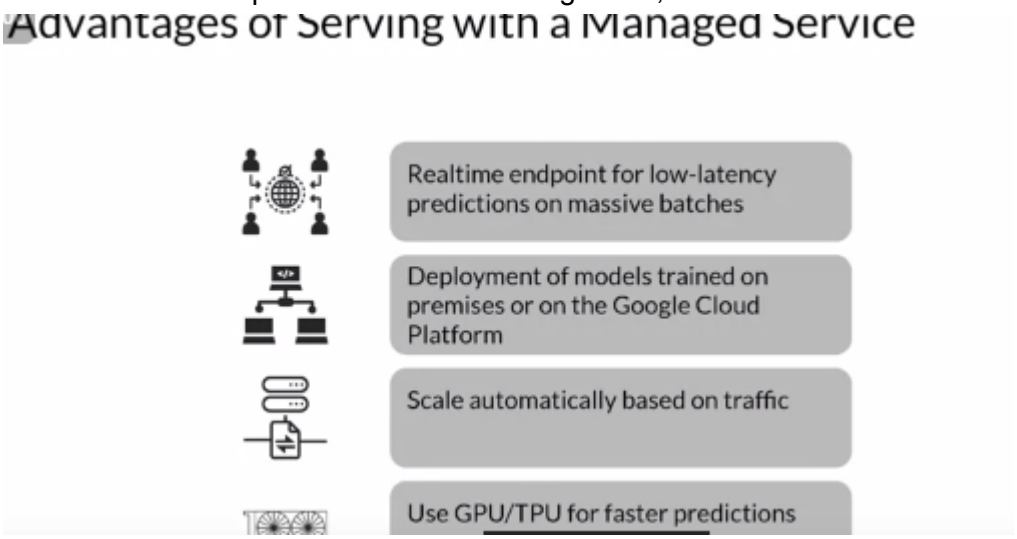
1. **Semana 1**
    - **Model serving** é utilizar um servidor para fornecer dados para um modelo, isso pode ser online (com o mdelo sendo atualizado a quase todo instante) ou offline, deve se considerar coisa como delay entre ações do usuário, latência, throughput, etc. Ainda mais em casos como carros autonomos ou recomendações em sites, latência é bastante importante, deve-se prestar atenção no maximizar throughput x minimizar latência
    - A infraestrutura deve ter um balanço entre o custo e a velocidade, levando em consideração coisas como acuracia, mas sem esquecer de latência e coisas o tipo

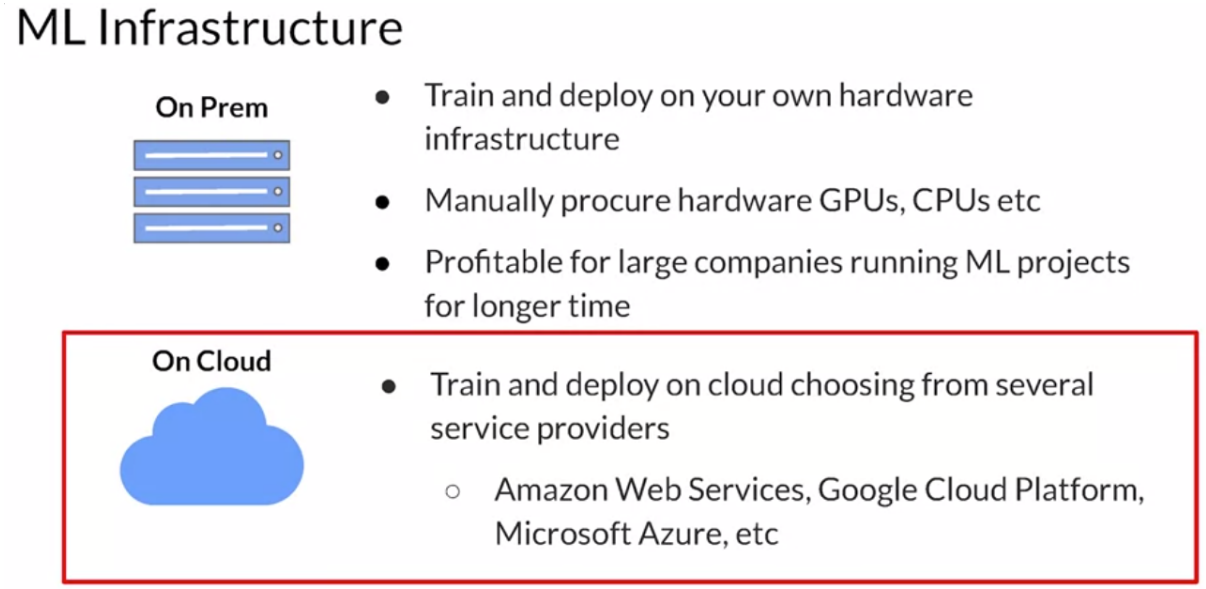


    - Há diversas formas de se dar deploy no modelo, seja em centros de dados ou no próprio dispositivo do usuário (deve-se considerar coisas como tamanho, uso de cpu, etc, usuários não vao intalar algo pesado), existem diversas ferramentas para lidar com isso, como clipper, tensorflow serving. Essas ferramentas permitem uma série de ganhos, como:

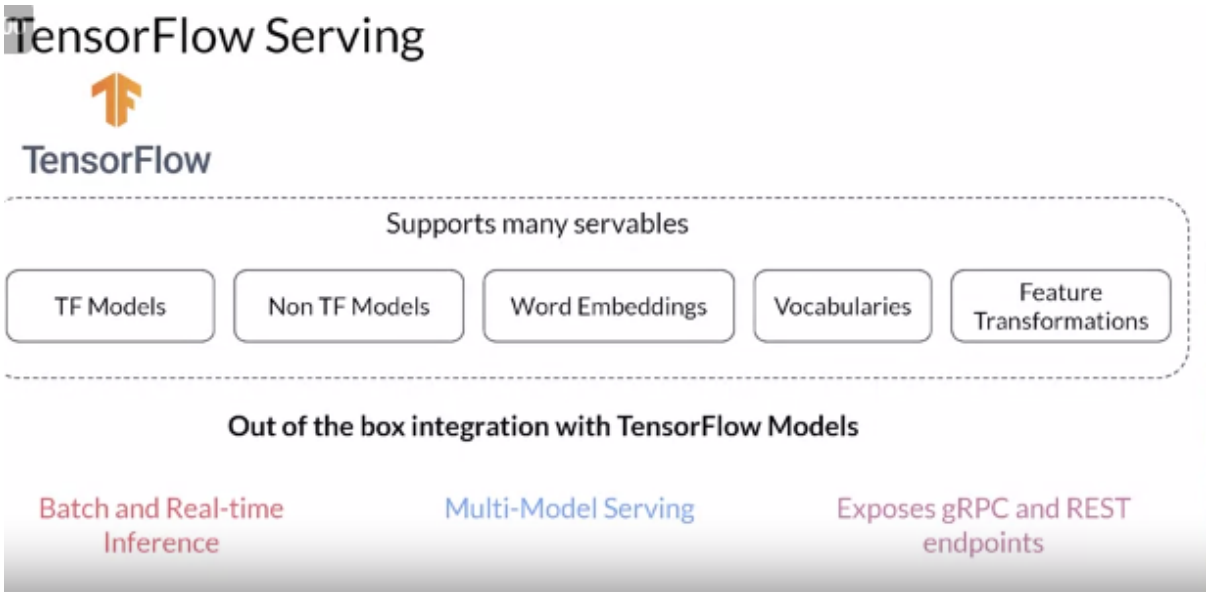


    -
2. **Semana 2**

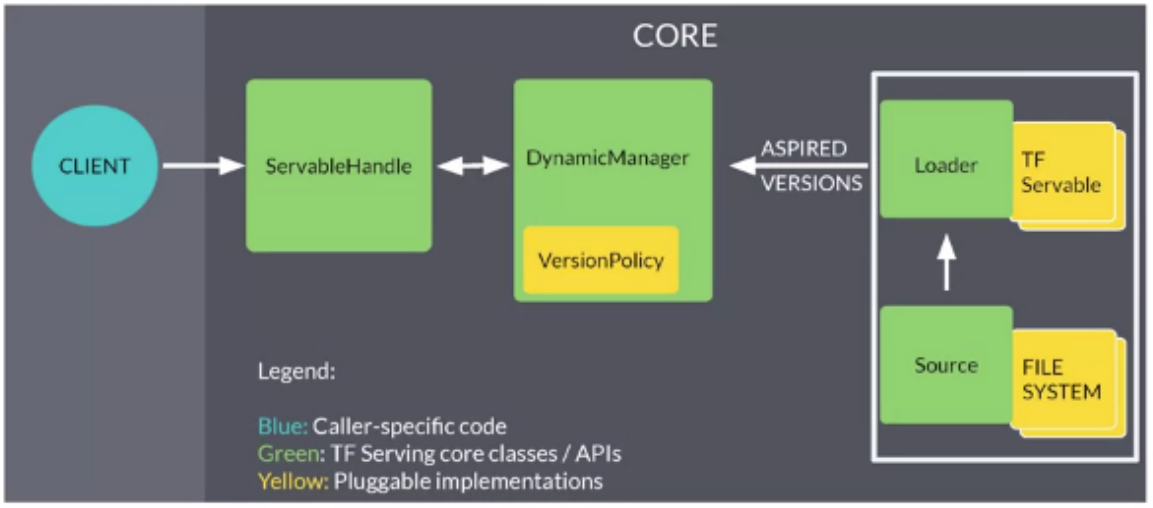
- ML infraestructure:



- Alguns servidores bastante utilizados>



TensorFlow Serving Architecture



NVIDIA Triton Inference Server

- Simplifies deployment of AI models at scale in production.
- Open source inference serving software
- Deploy trained models from any framework:
  - TensorFlow, TensorRT, PyTorch, ONNX Runtime, or a custom framework
- Models can be stored on:
  - Local storage, AWS S3, GCP, Any CPU-GPU Architecture (cloud, data centre or edge)

NVIDIA
TRITON INFERENCE SERVER

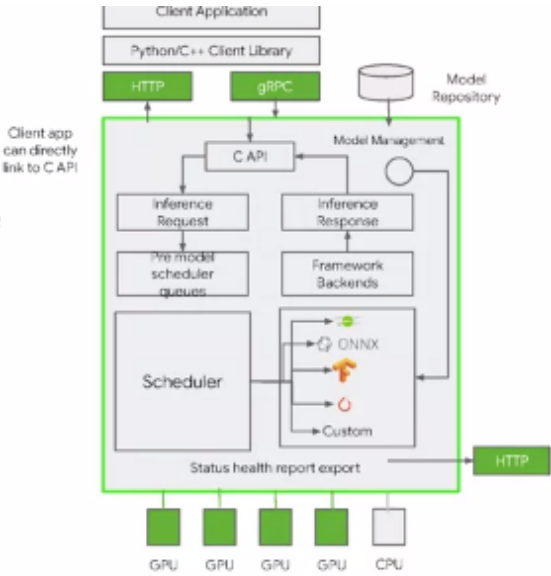
HTTP REST or gRPC endpoints are supported.

Architecture

Triton Inference Server Architecture supports:

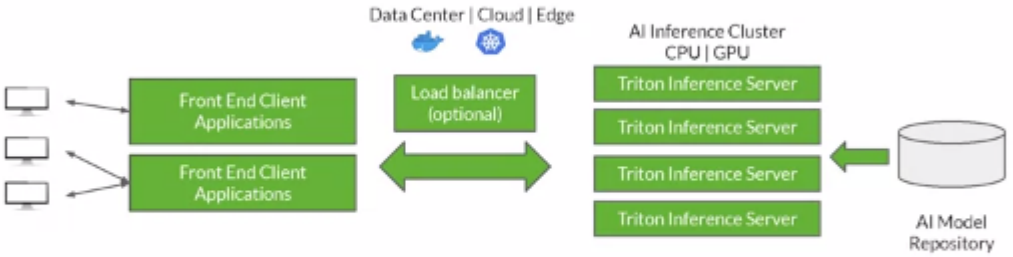
- Single GPU for multiple models from same or different frameworks
- Multi-GPU for same model
  - Can run instances of model on multiple GPUs for increased inference performance.

Supports model ensembles.



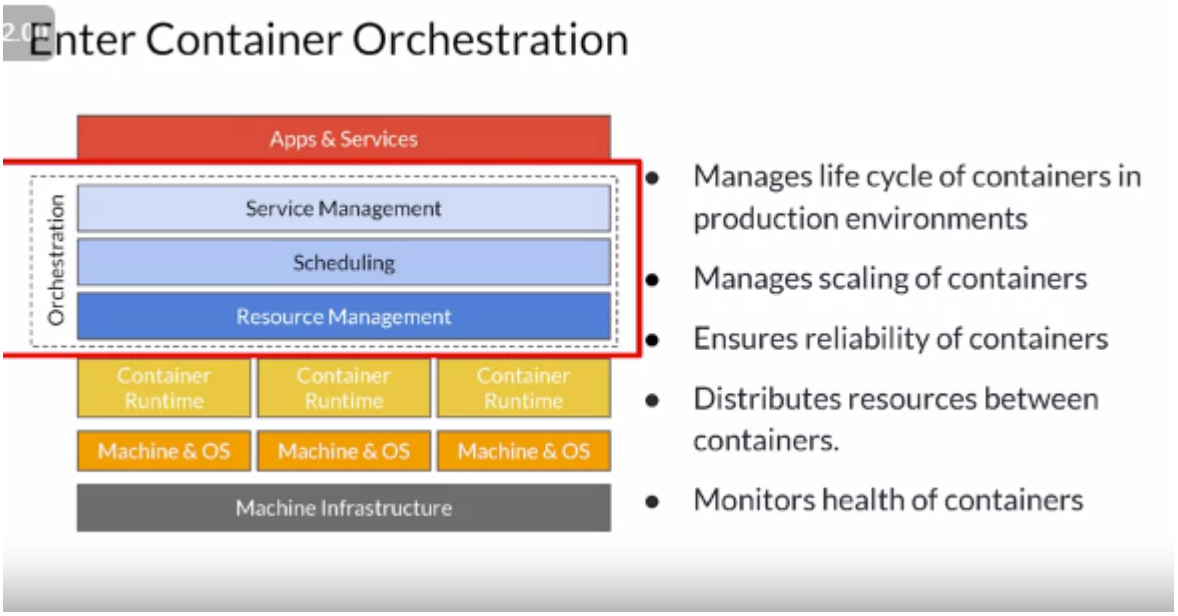
-

Designed for Scalability



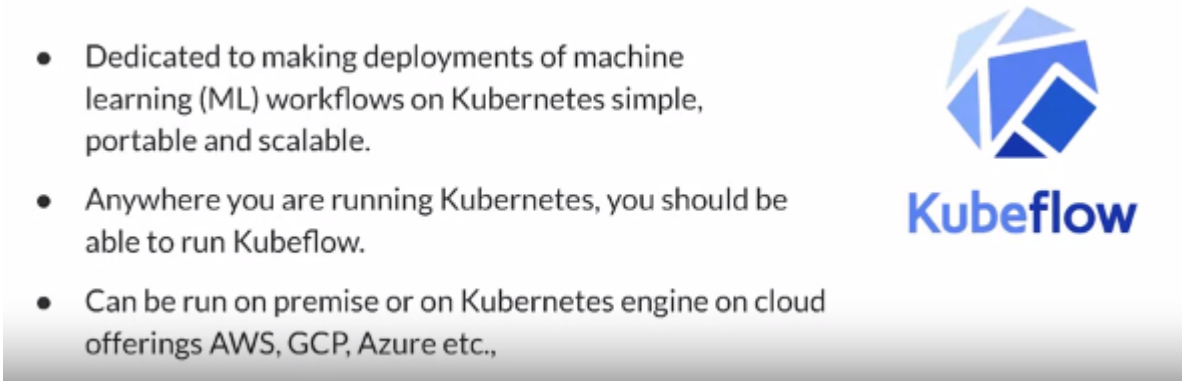
Can integrate with KubeFlow pipelines for end to end AI workflow

-
-
- **<u>Escalabilidade:</u>**



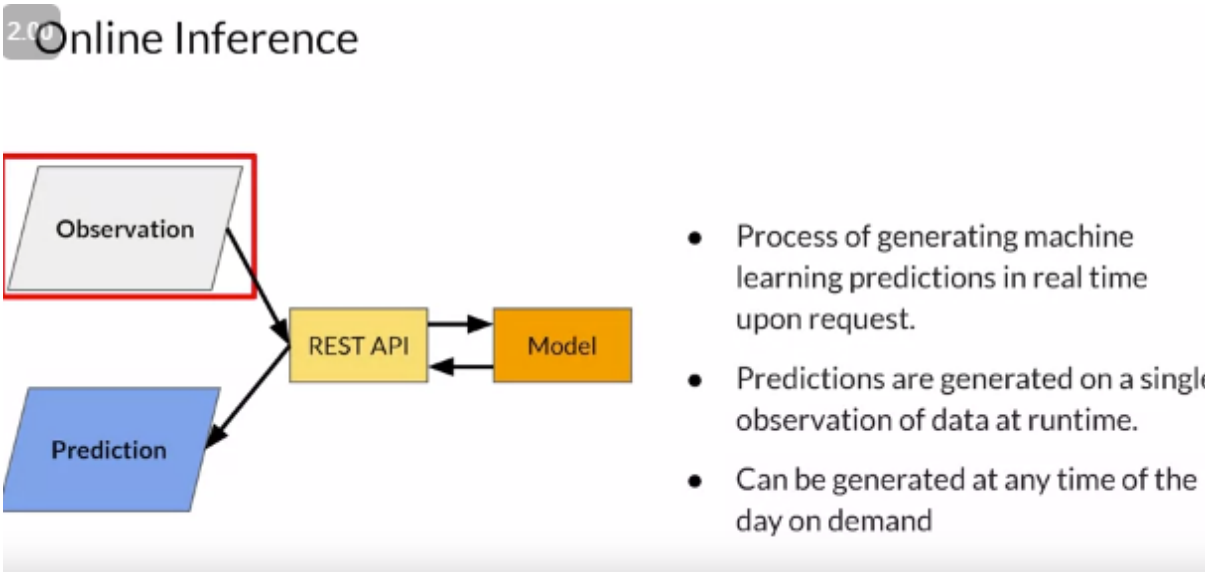
-

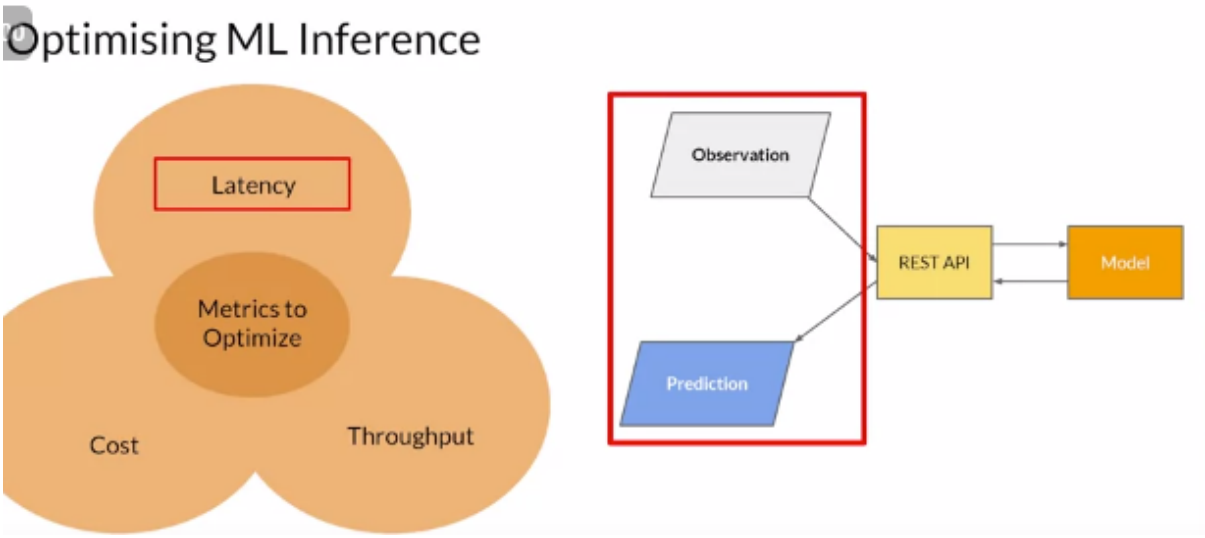
- **Kubernets:** um sistema de código aberto para automação, implantação, dimensionamento e gerenciamento de aplicativos em contêineres. Os contêineres de um grupo que compõem um aplicativo em unidades lógicas para fácil gerenciamento e descoberta.
- **Kuberflow:**



- **Online inference**: processo de gerar predições em tempo real dado uma requisiçao, predições são geradas em uma unica observação de dados no

runtime, são gerados em qualquer hora de demanda do dia





-
- Antes de fazer uma inferência pelo modelo, geralmente se precisa pré processar esses dados e também pos processar as vezes

Preprocessing Operations Needed Before Inference

Data Cleansing
- Correcting invalid values in incoming data

Feature Tuning
- Normalization
- Clipping outliers
- Imputing Missing Values

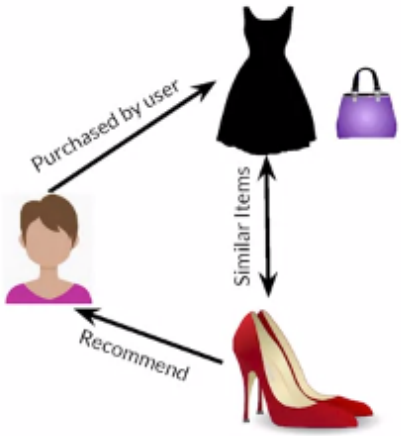
Feature Construction
- Combine inputs
- Feature crossing
- Polynomial expansion

Representation Transformation
- Change data format for the model
- One-hot encoding
- Vectorization

Feature Selection
- Apply same feature selection done during training on incoming data and features fetched from cache

-
- Batch inference gera predições a partir de um batch de predições, são geralmente recorrentes , reduz o custo do sistema, pode ter problemas em sistemas real time,

Use Case - Product Recommendations

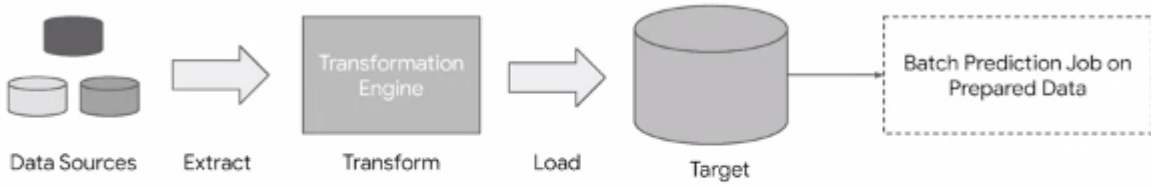


- E-commerce sites: new recommendations on a recurring schedule
- Cache these for easy retrieval
- Enables use of more predictors to train more complex models.
  - Helps personalization to a greater degree, but with delayed data

Data Processing - Batch and Streaming

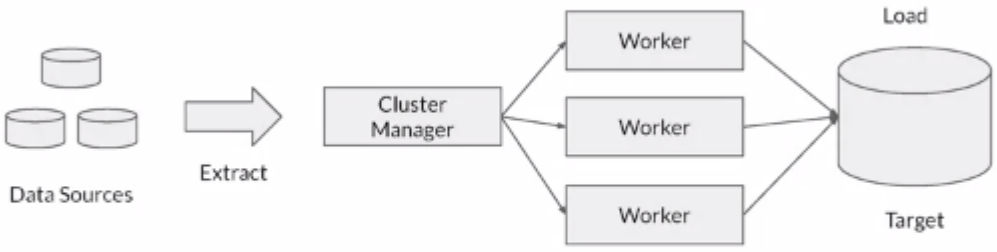
- Data can be of different types based on the source.
- **Batch Data**
  - Batch processing can be done on data available in huge volumes in data lakes, from csv files, log files etc.,
- **Streaming Data**
  - Real-time streaming data, like data from sensors.

ETL Pipelines

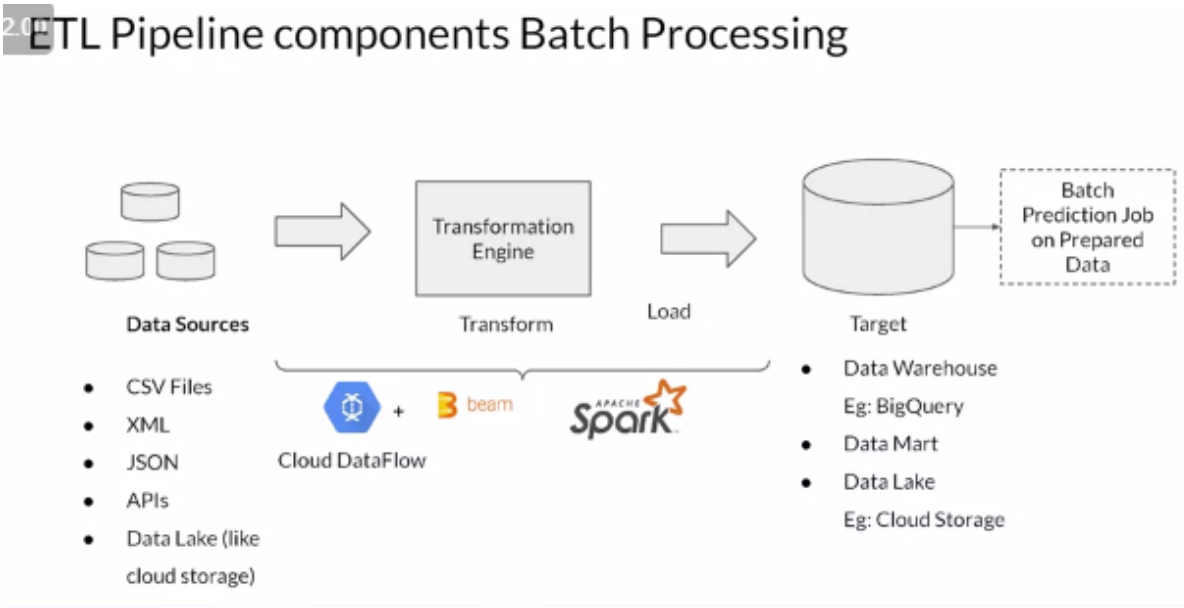


- Set of processes for
  - extracting data from data sources
  - Transforming data
  - Loading into an output destination like data warehouse
- From there data can be consumed for training or making predictions using ML models,
- ETL - Extrack track load

Distributed Processing



- ETL can be performed huge volumes of data in distributed manner.
- Data is split into chunks and parallely processed by multiple workers.
- The results of the ETL workflow are stored in a database.
- Results in lower latency and higher throughput of data processing.



-

3. **<u>Semana 3</u>**
    - Experiment tracking: fazer o rastreio dos experimentos é algo muito importante, seja por consumo, seja por tempo, o importante é entender tudo que vai dentro de cada experimento , os hyperparametros, etc.
    - notebooks são feramentas interessantes, entretanto não podem ser tão util para produção, existem diversas ferramentas que transformam / convertem um notebook para um .py, por exemplo
    - Deve se pensar em um código modular, ainda mais em time, além de usar git.
    - Usar config é interessante, além de command line, etc
    - Além de controlar versionamento do modelo, também deve se controlar o versionamento do dados, como neptune, pachyderm, delta lake, git lfs (imagems, videos)
    - Tensorboard é bem interessante para controlar os parâmetros e o treinamento do modelo conforme os experimentos acontecem.
    -