TARENTO

MLOps: *Deploying ML Models as Microservices with Seldon Core*

Dileep Gadiraju

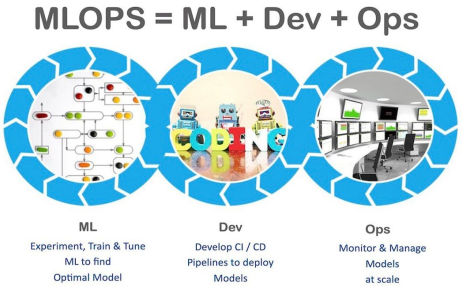
GitHub: https://github.com/dileep-gadiraju/try-seldon-core

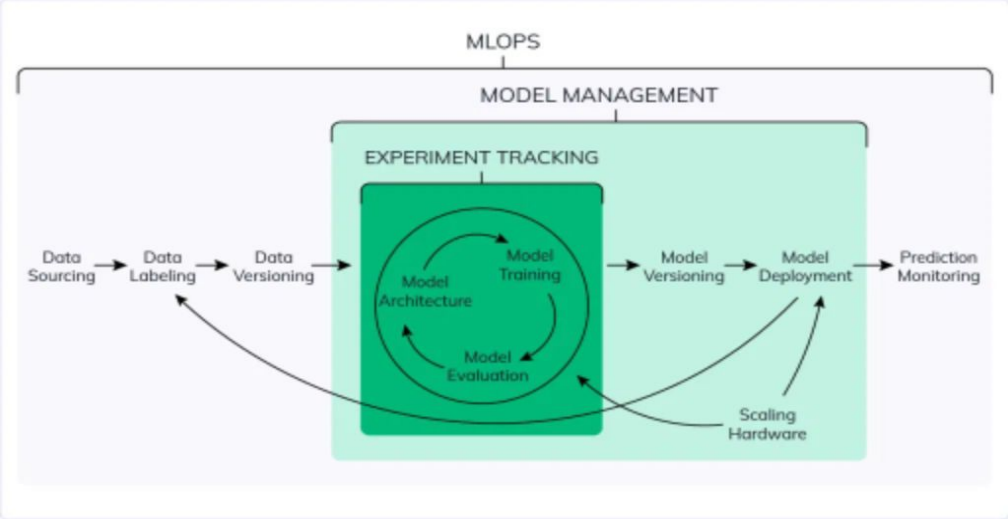
Topics

- What is MLOps ?
- Kubernetes for Automation
- Let us understand CRDs and Operators in K8S
- Ambassador - API Gateway
- Istio - Service Mesh
- NVIDIA Triton Inference Server
- Introduction to Seldon Core
    - What is Seldon Core?
    - Seldon Ecosystem
    - Seldon Core Architecture
    - Seldon Deployment CRD
    - Seldon Core Architecture
    - Scaling ML model APIs
- Overall Landscape
- Demo

What is MLOps ?

“MLOps or ML Ops is a set of practices that aims to deploy and maintain machine learning models in production reliably and efficiently.”





Kubernetes for Automation

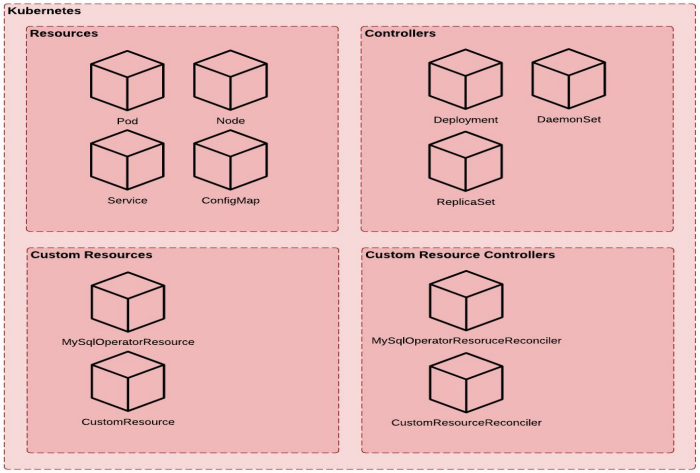
- Kubernetes a.k.a **K8s** is an open-source container-centric application management software for automating deployments, scaling.
- Built-in automation for deploying and running workloads.
- K8S distributions:
  - Openshift
  - VMware Tanzu
  - Mirantis Kubernetes Engine
  - Rancher Kubernetes Engine
  - Docker Kubernetes Engine(DKE)

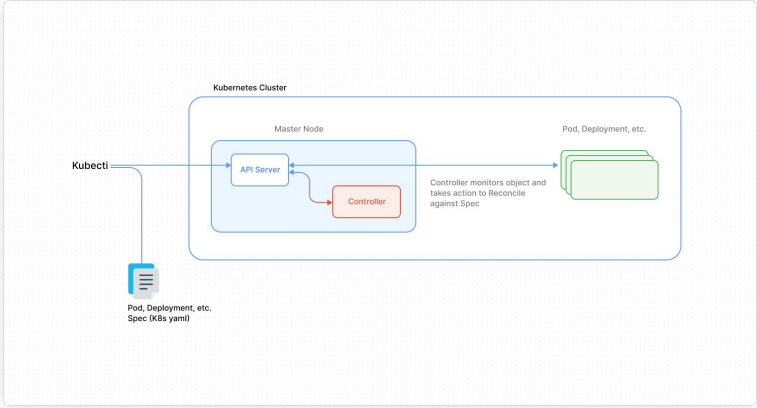
kubernetes

Kubernetes Objects

CRDs and Operator in K8s

- ***Custom Resources*** are extensions of the Kubernetes API
- *A **resource** is an endpoint in the Kubernetes API that stores a collection of API objects of a certain kind*
- *Custom resources can appear and disappear in a running cluster through dynamic registration*
- ***Kubernetes controllers** are the powerful means by which the platform regulates itself to ensure it maintains the desired conditions.*
- ***Custom Controllers** is controller component for custom resources to monitor and maintain.*
- ***Operator** extends K8S cluster behavior without modifying K8S code by linking **customer resources** and **controllers**.*



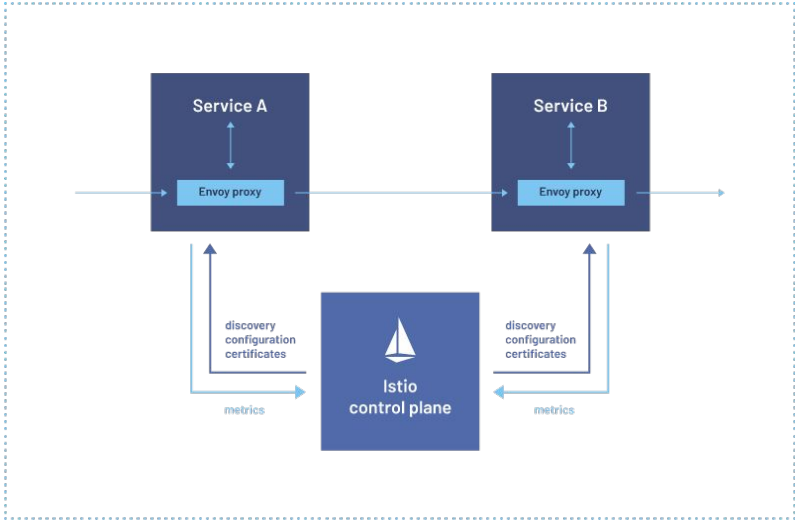


Istio - A Service Mesh

“As proxy sidecar that can help with below features in Microservices architecture. Managers communication between microservices”

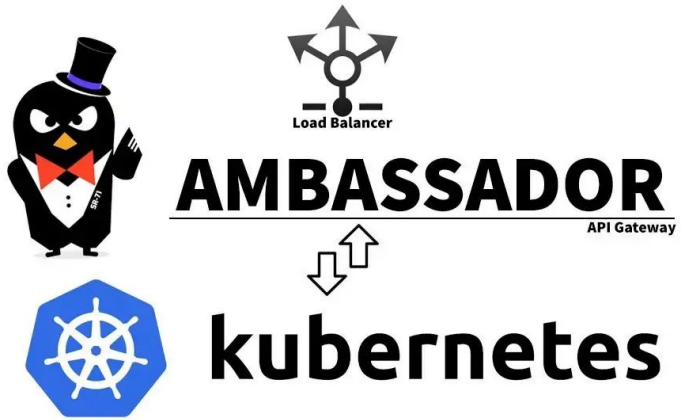
Core capabilities of Istio:

- Traffic management
  - Virtual Services
  - Destination Rules
  - Gateways
  - Service Entries
  - Side Cars
- Observability
  - Access logs
  - Metrics
  - Distributed Tracing - profiling
- Security Capabilities
  - Authentication & Authorization policies
  - Secure Naming Information
- Extensibility



Ambassador - API Gateway

“API gateway is a service that accepts incoming API requests from clients, directs the request to the appropriate application service, processes that service's response and relays that response to the requesting client.”



API Gateway vs Service Mesh

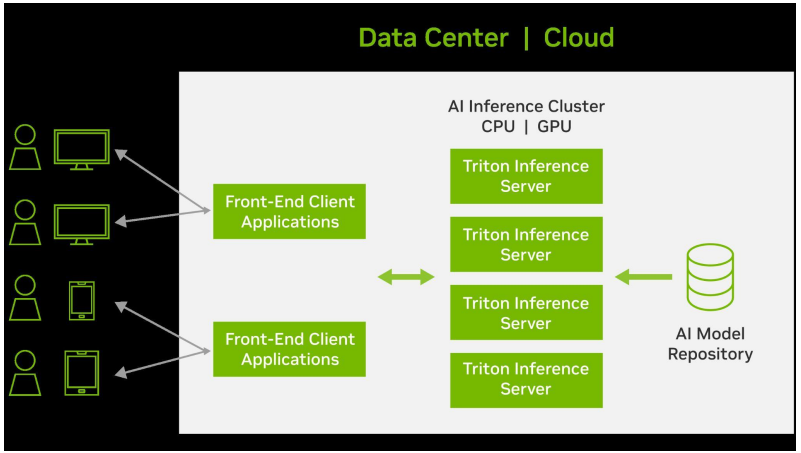
- Abstraction
- Decoupling
- Edge Routing
- Edge Security

- Endpoints , Hosts, Ports
- Traffic Routing
- Security
- Observability
  - Metrics Collection
  - Access logs
  - Distributed tracing - profiling
  -

NVIDIA Inference Server

“A open-source inference serving software that helps standardize model deployment and execution and delivers fast and scalable AI in production.”

- Support for multiple frameworks.
- High-performance inference.
- Designed for DevOps and MLOps.
- An integral part of NVIDIA AI.



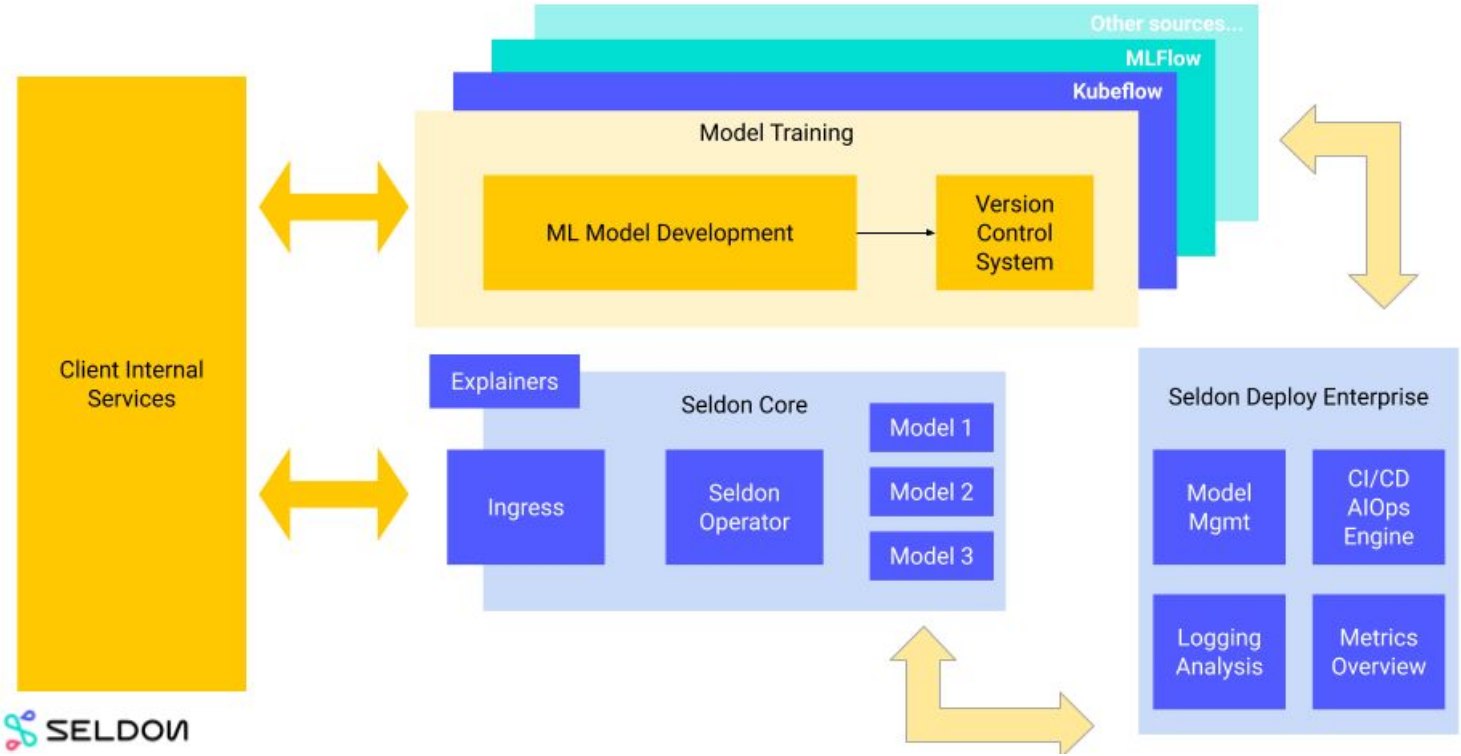
[Model Repository Examples]

Introduction to Seldon Core

“**Seldon core** open source framework to convert ML models (Tensorflow, Pytorch, H2o, etc.) or language wrappers (Python, Java, etc.) into production REST/GRPC microservices.”

- **Runs anywhere** - Built on Docker and Kubernetes, runs on your local machine, on any cloud and on premises
- **Agnostic and independent** - Framework agnostic, supports top ML libraries, toolkits and languages **(**eg**.** Kubeflow**)**
- **Runtime inference graphs** - Advanced deployments with experiments, ensembles and transformers
- Seamlessly integrates with **NVIDIA Triton Inference server**

Seldon Ecosystem - End-to-end Architectural Overview



Seldon Core Architecture

1. Containerise
2. Deploy
3. Monitor

From model binary

Or language wrapper

Into fully fledged microservice

Simple Seldon Core Inference Graph

Model A → API (REST, gRPC)

Complex Seldon Core Inference Graph

Model B, Model A, Model C → Multi Armed Bandit → Feature Transformation → Outlier Detection → Explanation → API (REST, gRPC)

Multi Armed Bandit: Direct traffic to the most optimal model

Outlier Detection: Key features to identify outlier anomalies (*Fraud, KYC*)

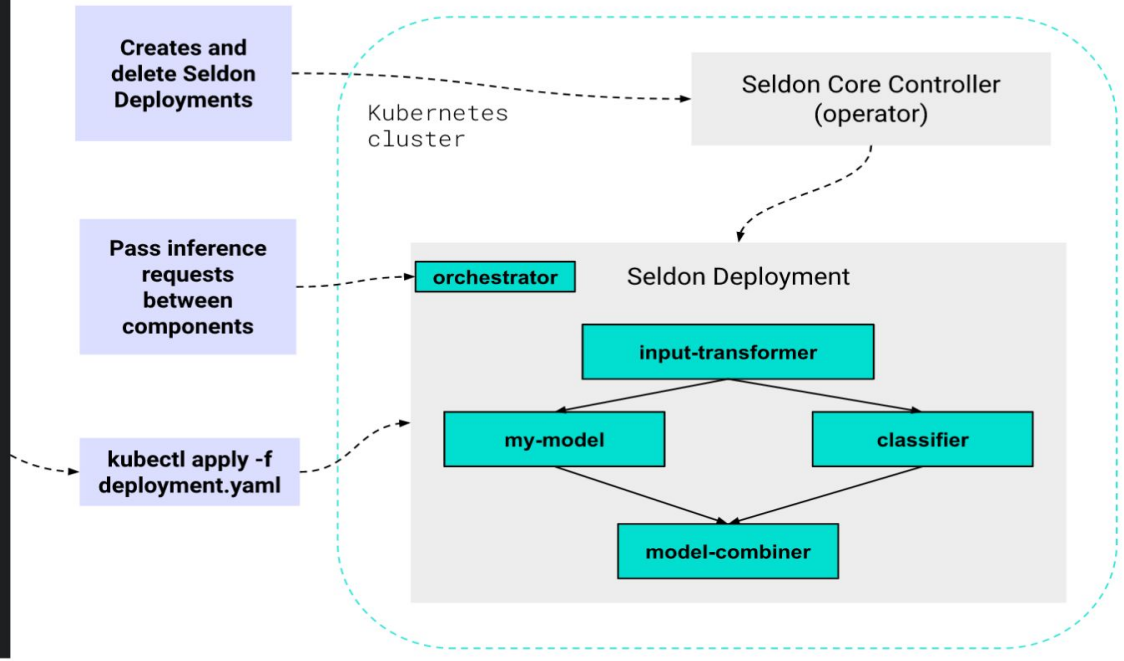
Explanation: Why is the model doing what it's doing?

Seldon Deployment CRD

[Seldon Inference Graph]

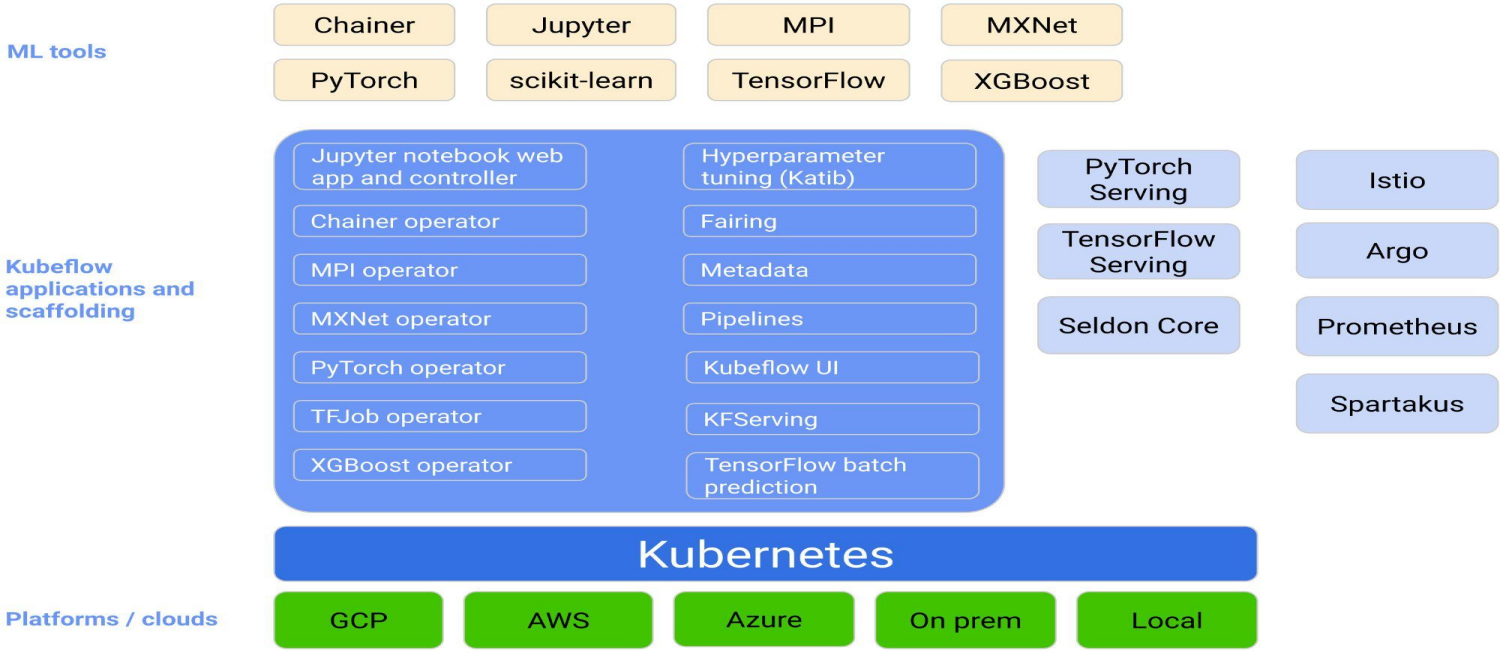
deployment.yaml

```yaml
apiVersion: machinelearning.seldon.io/v1
kind: SeldonDeployment
metadata:
  name: example-model
spec:
  name: example
  predictors:
  - componentSpecs:
    - spec:
        containers:
        - image: model:0.1
          name: my-model
        - image: transformer:0.1
          name: input-transformer
        - image: combiner:0.1
          name: model-combiner
    graph:
      name: input-transformer
      type: TRANSFORMER
      children:
        - name: model-combiner
          type: COMBINER
          children:
          - name: my-model
            type: MODEL
          - name: classifier
            implementation: SKLEARN_SERVER
            modelUri: gs://seldon-models/sklearn/iris
    name: default
    replicas: 1
```



Overall Landscape

"A open-source orchestration toolkit/platform for machine learning on Kubernetes"



Demo

- Kind Cluster and Istio Setup
- MINIO - A Multi Cloud Storage setup
- CRD examples in Seldon Environment
- Seldon Deployment CRD
- Seldon Protocol Examples -> **protocol_examples.ipynb**
- Seldon Graph Examples -> **graph-examples.ipynb**
- Triton container demo
- Model Repository with Triton,Seldon,MINIO -> **triton_minio_model_store.ipynb**
- Scaling Seldon Deployments -> **scale_examples.ipynb**
- Seldon Monitoring with Prometheus -> **metrics_prometheus.ipynb**

Thank you!