Time to Merge Tool

Podman Community Call - Jan 19.2023

Aakanksha Duggal
ET Data Science Team

Red Hat

AI4CI: Open Source AIOps toolkit

Problem

- Need for **AIOps** - Automated monitoring, analysis, alerting for Ops (CI/CD, development processes)
- **Open Source data** originating from real world production systems is a rarity for public datasets.
- Lack of AI driven metrics for open source community health.

Opportunity

- **Open operations data** made available by running open source software and applications in production.
- Data includes CI/CD data, code, telemetry, logs, operational dashboards.
- Eg: Kubernetes testing infrastructure, Fedora make their testing data available open source.

Solution

- Collection of intelligent and open source **data science tools** to collect and analyze the CI/CD data.
- **AI models** like Github time-to-merge service, optimal stopping time prediction, build log classifier
- KPI and Metric dashboards
- Goal is to foster an open source AIOps community with open ops data, AI tools and services.

Red Hat

AI4CI supports CI/CD and software dev processes

What is AI4CI?

Collection of **Open Source AIOps tools** including scripts, notebooks, pipelines, dashboards and data sources.

github.com/aicoe-aiops/ocp-ci-analysis

Data collection

Collection of open operations data from Kubernetes testing platforms eg: Testgrid, Github, and Prow.

Metrics

Collects metrics and **KPIs** and visualization dashboards.

ML Services

ML services which can support CI/CD processes.

Open source AIOps template

Resource for open source AIOps communities (notebooks, scripts, automated ML pipelines, dashboards, services tools)

Red Hat

Time to Merge Model

A new PR is opened on a Git Repository

Predict the time that it will take to merge this open PR, and add a label on the PR

Learn from the actual time to merge and create a feedback loop

Time to merge prediction service for community health

- Identify **bottlenecks** in development process
- Leverage the rich **historical data** of consisting of Issues, Commits, PRs
- Give **new contributors** of an estimate of when their PR will be reacted upon
- Similar models like **time to review** can encourage reviewers

Toolification Process

Collect data → **Features** → **Model Building** → **Training Action** → **Predictions**

Use the Github API to collect **data** from code repositories

Features like **size of the PR, types of files in the PR description of the PR**

Train repository or org specific model

Github Action Workflow that trains a model on repo of interest and offers predictions on PRs

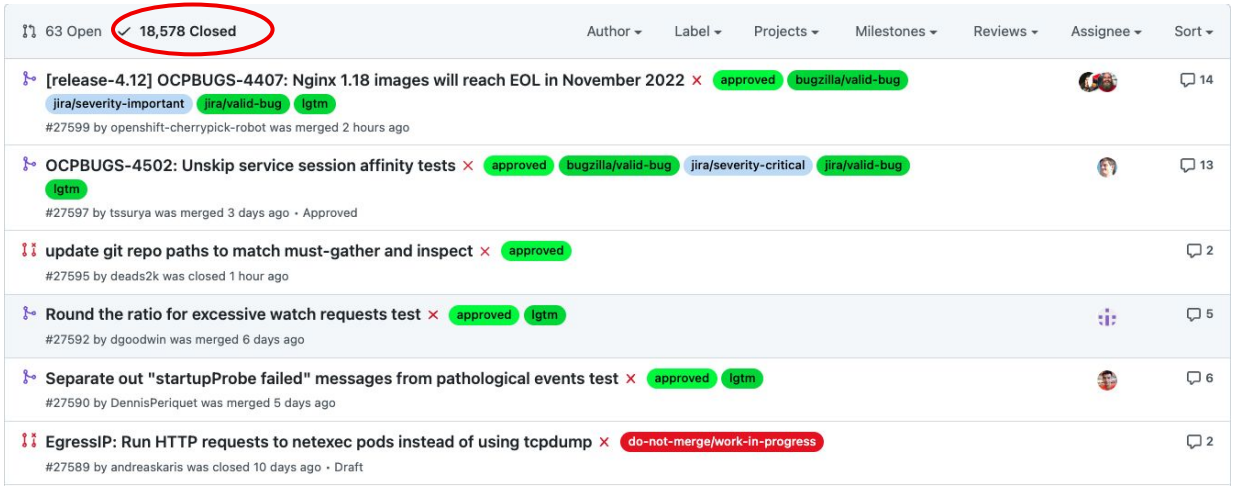
Action configured on repo to offer **live time estimate prediction** on newly open PRs

PRs from repos → **Discover project specific behavior** → **Model returns merge time estimate** → **Train custom model using github action** → **Offer live prediction using github action**

ML Service: Github Time to Merge Model

COLLECT DATA

openshift / origin



ML Service: Github Time to Merge Model

GITHUB PR DATA

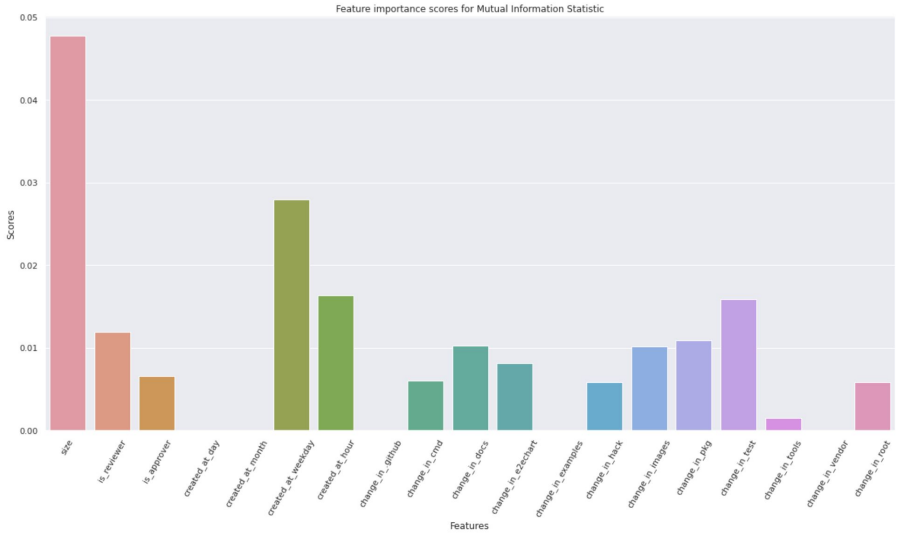
openshift / origin

FEATURE ENGINEERING

size of the PR,

types of files in the PR

description of the PR



ML Service: Github Time to Merge Model

GITHUB PR DATA → FEATURE ENGINEERING → MODEL TRAINING

openshift / origin

size of the PR,

types of files in the PR

description of the PR

Naive Bayes

XGBoost

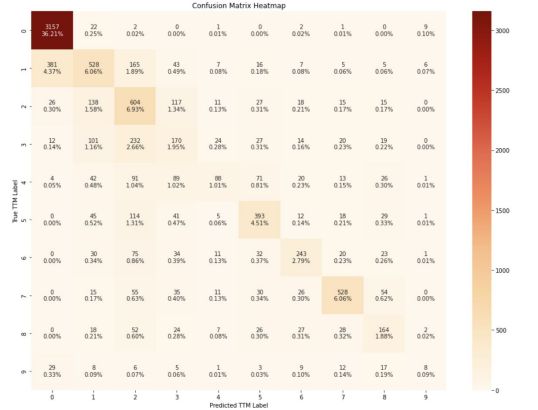
Random Forest

SVM

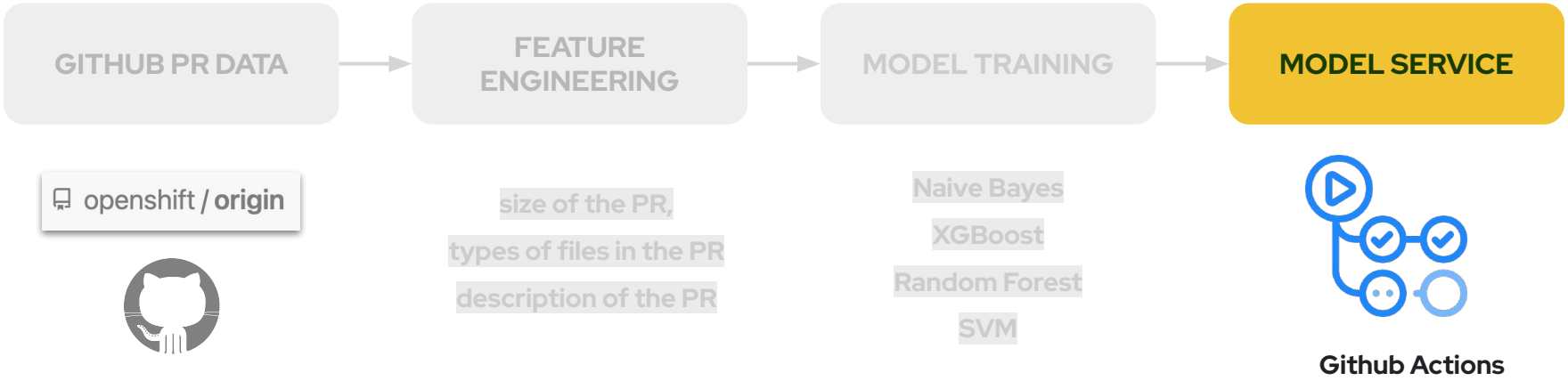
XG Boost

```
train_evaluate_classifier(xgbc, X_train_scaled, y_train, X_test_scaled, y_test)
```

	precision	recall	f1-score	support
0	0.87	0.99	0.93	3194
1	0.56	0.45	0.50	1163
2	0.43	0.62	0.51	971
3	0.30	0.27	0.29	619
4	0.53	0.20	0.29	445
5	0.63	0.60	0.61	658
6	0.64	0.52	0.57	469
7	0.88	0.70	0.75	754
8	0.47	0.47	0.47	348
9	0.29	0.08	0.13	98
accuracy			0.67	8719
macro avg	0.55	0.49	0.50	8719
weighted avg	0.66	0.67	0.66	8719

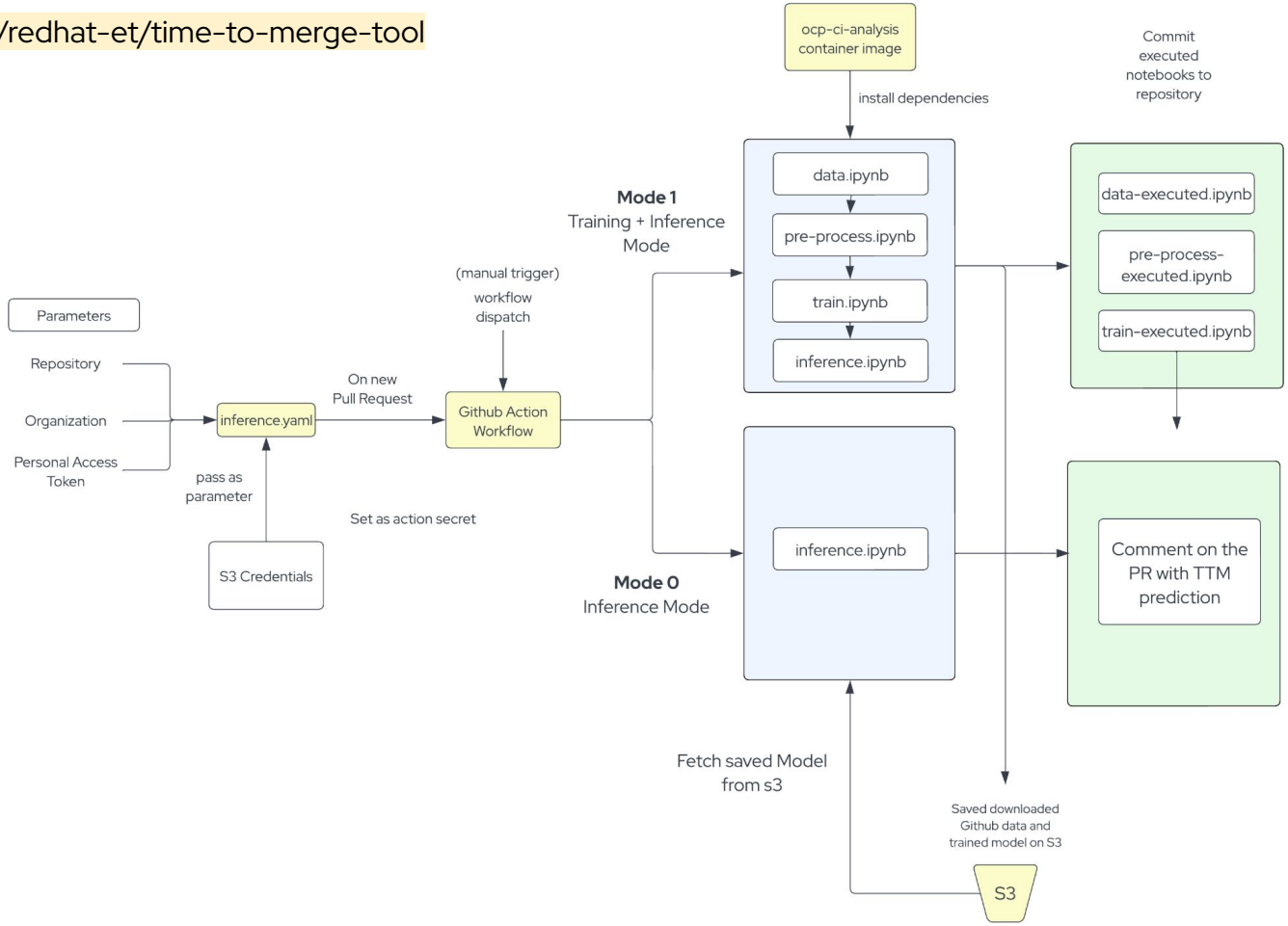


ML Service: Github Time to Merge Model

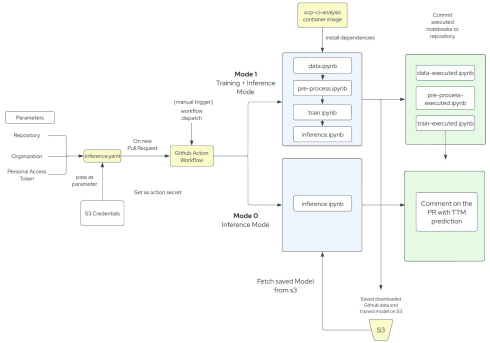


Time to Merge Model: Github Action Workflow

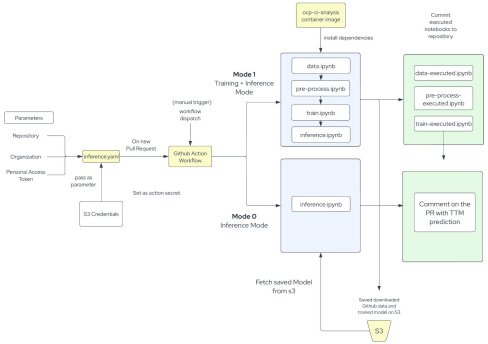
github.com/redhat-et/time-to-merge-tool



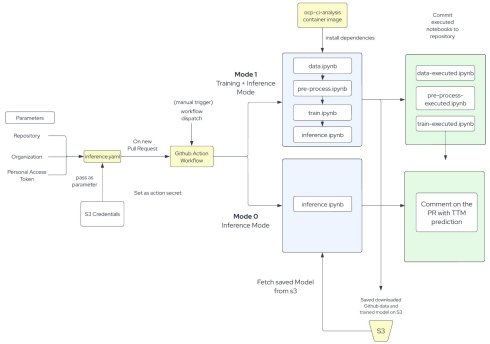
Train the model for each different repository or organization



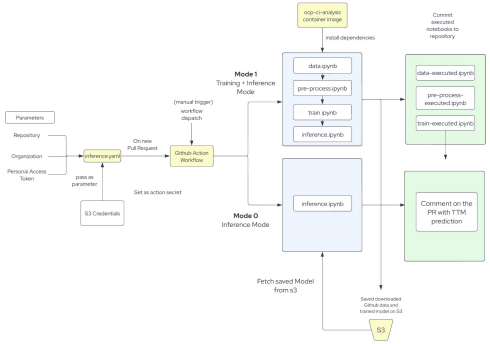
aicoe-aiops/ocp-ci-analysis



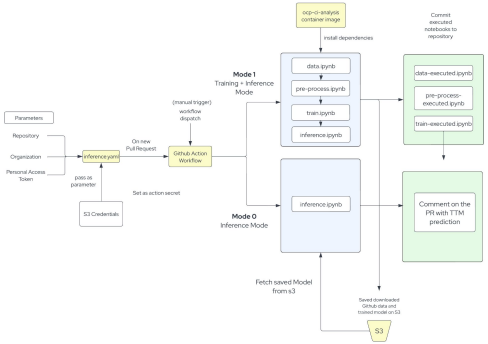
openshift/origin



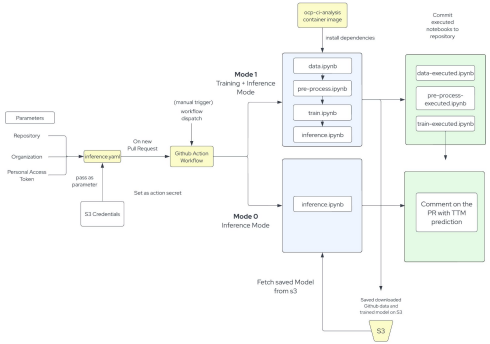
ansible/ansible



thoth-station/support



operate-first/support



redhat-et/copilot-ops

Key Components

action.yml

17 lines (14 sloc) | 427 Bytes

```yaml
name: Time To Merge Tool - Model Inference Test
description: 'This is the github action to predict time to merge for a new pull request'
author: 'redhat-et'

inputs:
  MODE:
    description: "0 : Inference Mode and 1: Training and Inference Mode"
    required: true
    default: 1

outputs:
  prediction:
    description: 'Provides a prediction of the PRs time to merge'

runs:
  using: 'docker'
  image: 'Dockerfile'
```

Input

Modes:
1: Training & Inference
0: Only Inference

Output

Training & Prediction
Or only Prediction

Dockerfile

10 lines (7 sloc) | 316 Bytes

```
1  # Container image that runs your code
2  FROM quay.io/aicoe/ocp-ci-analysis
3
4  # Copies your code file from your action repository to the filesystem path `/` of the container
5  COPY . /
6
7  RUN chmod +x /entrypoint.sh
8
9  # Code file to execute when the docker container starts up (`entrypoint.sh`)
10 ENTRYPOINT ["/entrypoint.sh"]
```

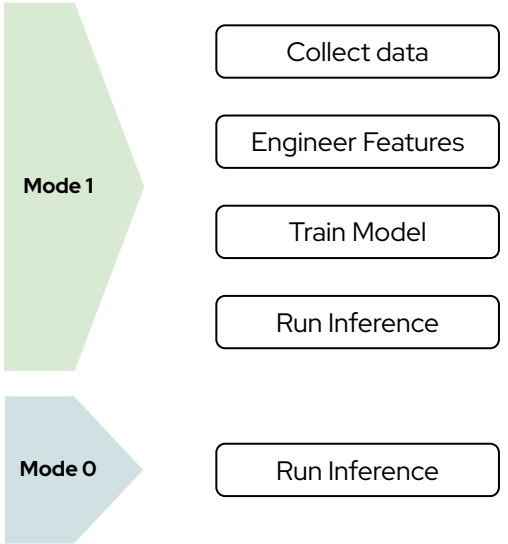
← Base image that this is running on

← Script to execute when container starts up

entrypoint.sh

27 lines (22 sloc) | 1.1 KB

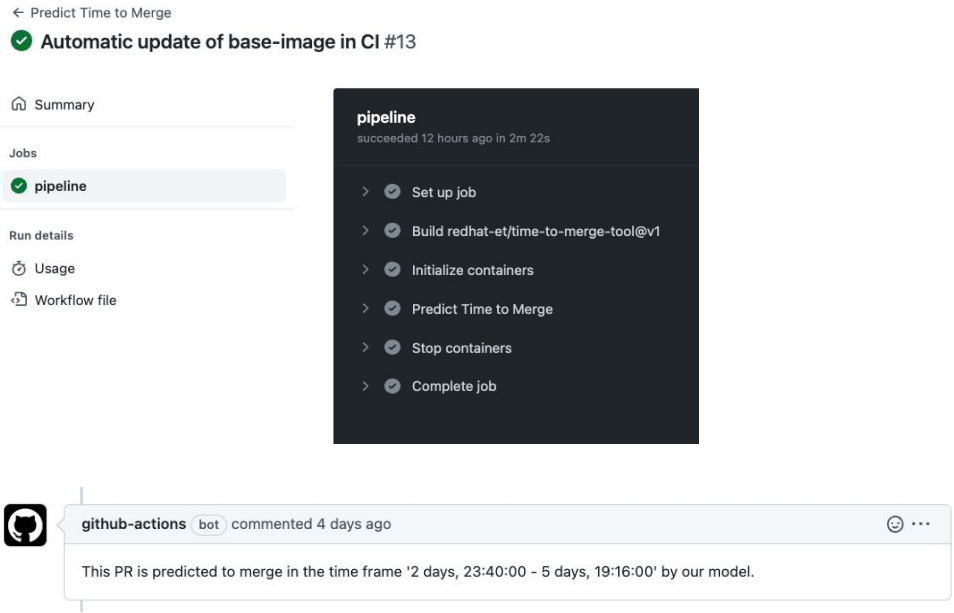
```
#!/bin/sh
# If mode is 1, run training + inference mode, otherwise just run the inference

if (( $MODE==1 ))
then
 echo "Training Mode"
 # Data collection
 python3 /01_data_collection.py

 # Feature Engineering
 jupyter nbconvert --to notebook --execute /02_feature_engineering.ipynb --TemplateExporter.exclude_input=True \
 --ExecutePreprocessor.kernel_name='python3' --output 02_notebook_executed

 # Model Training
 jupyter nbconvert --to notebook --execute /03_model_training.ipynb --TemplateExporter.exclude_input=True \
 --ExecutePreprocessor.kernel_name='python3' --output 03_notebook_executed

 # Model Inference
  jupyter nbconvert --to notebook --execute /04_model_inference.ipynb  --TemplateExporter.exclude_input=True \
 --ExecutePreprocessor.kernel_name='python3' --output 04_model_inference_executed

else
  echo "Inference Mode"
  # Since the mode wasn't specified just run the model inference on new pull request
  jupyter nbconvert --to notebook --execute /04_model_inference.ipynb  --TemplateExporter.exclude_input=True \
 --ExecutePreprocessor.kernel_name='python3' --output 04_model_inference_executed
fi
```



Example Usage

```yaml
jobs:
  pipeline:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest
    container:
      image: quay.io/aicoe/ocp-ci-analysis

    steps:
      - name: Predict Time to Merge
        uses: redhat-et/time-to-merge-tool@v1
        env:
```

Result



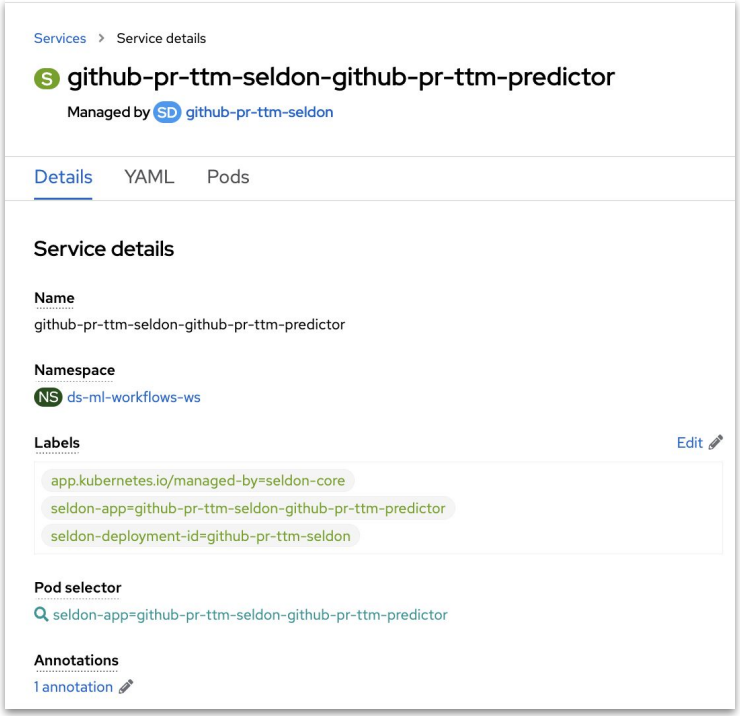
Demo on how to set this up for a new repository

Next Steps

- Extend time to merge model to organization level models
- Handle large number of PRs in github workflows by dynamically distributing jobs
- Extend Time to Merge to Time to Review (Ansible community has interest in this)
- Explore alternative deployment model (self-hosted)
- Add Model Monitoring and Summary Statistics

Thank you!

ML Service: Alternate Deployment method



MODEL DEPLOYMENT

SELDON

RED HAT® OPENSHIFT Container Platform