



A MLOPS JOURNEY

Jacob Bags

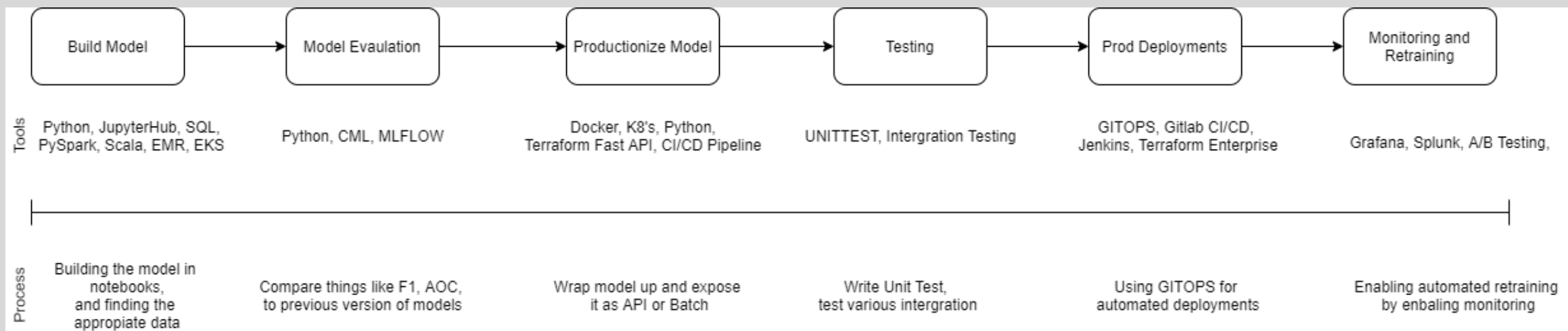
What is MLOPS

- MLOPS is the process of taking an experimental Machine Learning model into a production system¹
- Goals of dev ops¹:
 - Streamline deployments and automation
 - Reproducible models
 - Scalability
 - Monitoring and management
 - Governance and regulatory compliance
 - Collaboration

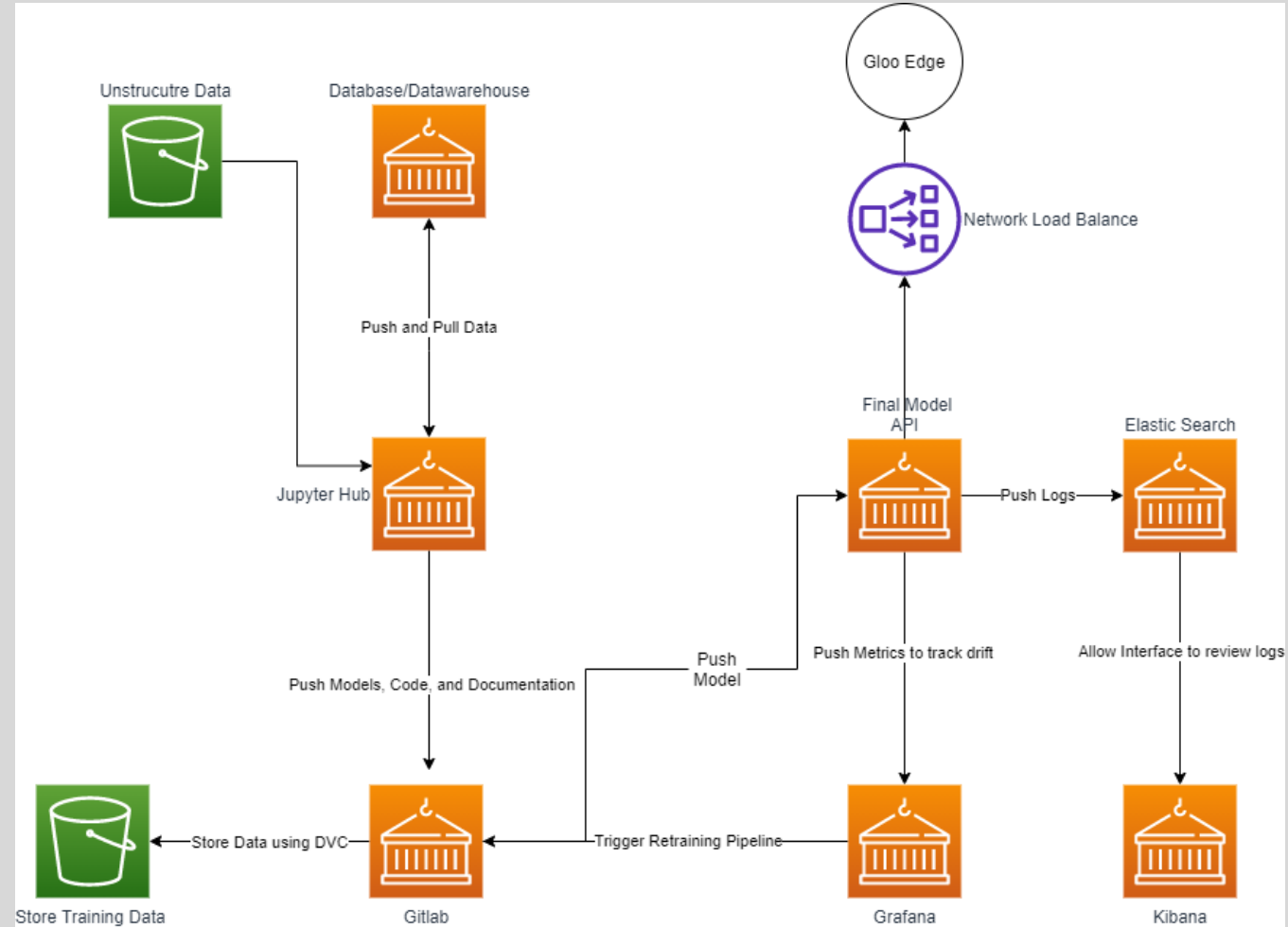
Example Problem

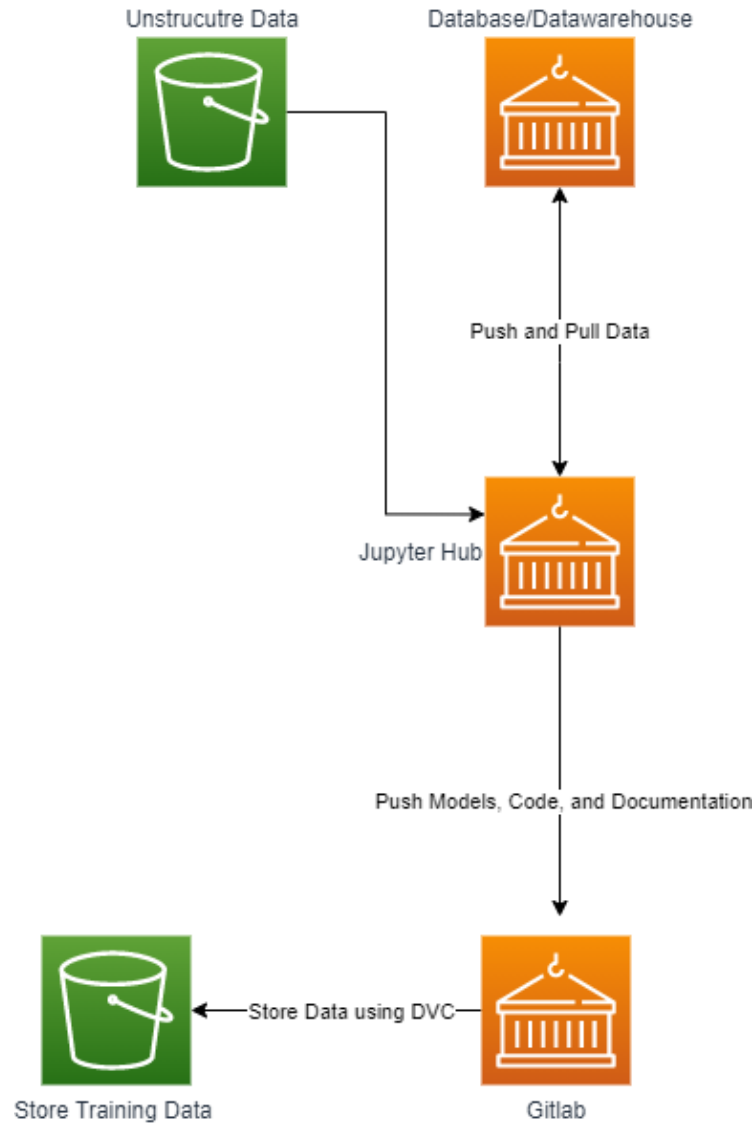
- Model Type: Binary Classification Model
- Model Service Type: API
- Volume: 1000's transaction a second

MLOps Journey



Example Architect- AWS

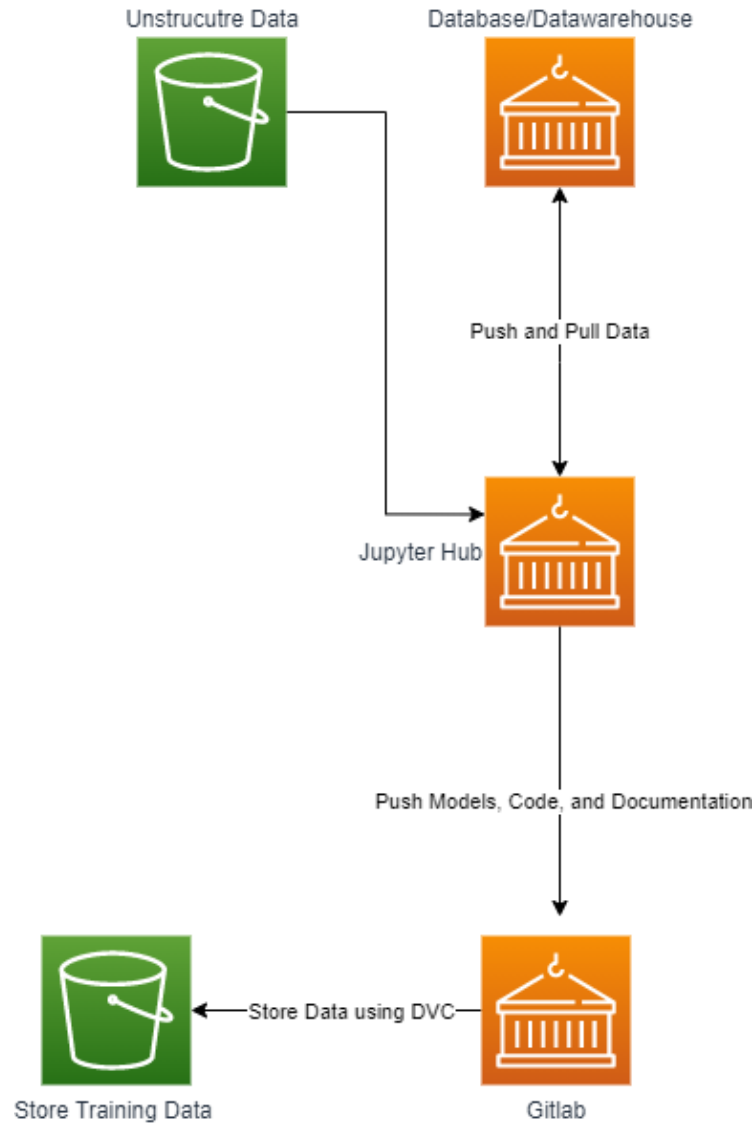




Building the Model

Tools:

- Database (Postgres, MySQL, Hadoop) or Data Warehouse
- Gitlab
- Git
- S3/ any type of blob storage
- Jupyter Hub
- Data Version Control (DVC)



Building the Model

Process:

- Evaluate Use Case
- Find the data
- Construct model
- Test the model
- Review model with SME
- Review model's ethics

Overview 2 Commits 5 Pipelines 5 Changes 8

Request to merge trying-stuff into master

Pipeline #300678764 passed for 2001d548 on trying-stuff 3 weeks ago

Approval is optional

View eligible approvers

Merged by jacob baggs 3 weeks ago Revert Cherry-pick

The changes were merged into master with e34b1f34

The source branch has been deleted

Pipeline #301287327 passed for e34b1f34 on master 3 weeks ago

0 0 Oldest first Show all activity

jacob baggs @jbaggs62 added 1 commit 3 weeks ago

- 9132bcc1 - stop tracking data

Compare with previous version

jacob baggs @jbaggs62 commented on commit 9132bcc1 3 weeks ago Maintainer

| Path | Metric | Old | New | Change |
|--------------|-----------|---------|---------|----------|
| metrics.json | accuracy | 0.87236 | 0.86772 | -0.00464 |
| metrics.json | precision | 0.86419 | 0.86691 | 0.00273 |
| metrics.json | recall | 0.88568 | 0.87098 | -0.0147 |

loss

step

rev

- master
- workspace

Reply...

Model Evaluation

Tools:

- DVC
- Continuous Machine Learning Library
- Gitlab CI/CD
- S3

24 lines (19 sloc) | 520 Bytes

```
1  # .gitlab-ci.yml
2  stages:
3    - cml_run
4
5  cml:
6    stage: cml_run
7    image: dvcorg/cml-py3:latest
8    script:
9      - dvc pull data
10
11      - pip install -r requirements.txt
12      - dvc repro
13
14      # Compare metrics to master
15      - git fetch --prune
16      - dvc metrics diff --show-md master >> report.md
17      - echo >> report.md
18
19      # Visualize loss function diff
20      - dvc plots diff
21        --target loss.csv --show-vega master > vega.json
22      - vl2png vega.json | cml-publish --md >> report.md
23      - cml-send-comment report.md
24
```

Model Evaluation

Tools:

- DVC
- Continuous Machine Learning Library
- Gitlab CI/CD
- S3

- Terraform
 - Infrastructure as Code
 - Allow easy control of infrastructure as code at scale
- Docker
 - Containerized code
 - Bring code anywhere
- Gitops
 - Allows for more control over coding going into prod
 - Uses development and config repos to allow separation of development code and prod code
 - Allows for more security
- Pipelines
 - YAML or JSON files to configure multiple steps
- Kubernetes
 - Orchestration tool
 - Allows for YAML configured ingress, service, and deployment
 - Highly Scalable and cloud agnostic

Putting the Model In Prod

Tools:

- Docker
- Kubernetes
- GitOps
- Gitlab CI/CD
- Terraform
- FastAPI
- API Gateways
- Security Scans



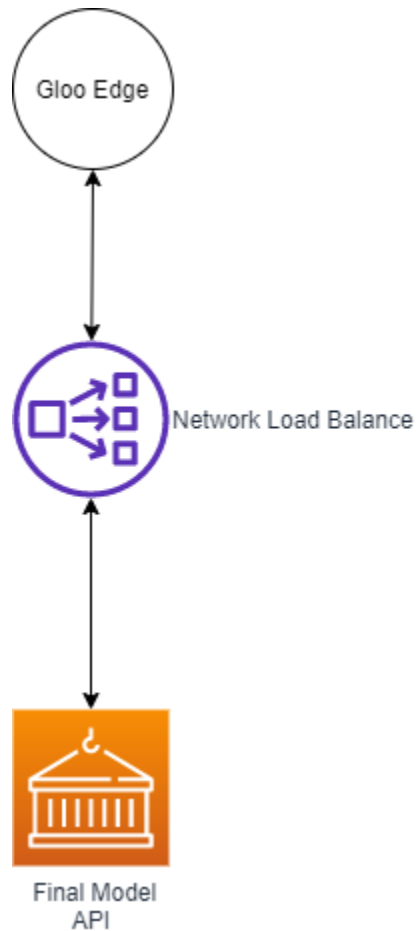
Putting the Model In Prod

Tools:

- Docker
- Kubernetes
- GitOps
- Gitlab CI/CD

Process:

- Write Code in development Repo
- Create Tag and push code into Config Repo
- Build Code in config repo and deploy



Model In Production

Tools:

- API Gateway
 - Allows for authentication
 - Secure private IP's behind public IP
- Network Load Balancer
 - Balance and coordinate traffic between different pods with tools like NGINX
- API
 - FASTAPI based on model deployed on Kubernetes

53 lines (41 sloc) | 1.82 KB

Raw Blame



```
1 #this is still a WIP
2
3 import unittest
4 from unittest import result
5 import pandas as pd
6 from nose.tools import assert_true
7 import requests
8
9 from app import clean_final_dataset, create_final_dataset, query_example
10
11
12 class test_clean_final_dataset(unittest.TestCase):
13     def ftesting_cleaning_dataset(self):
14         """
15         test cleaning final dataset. I created an ugly data frame and cleaned one and compared the two to mak sure the cleaning was working correctly
16         """
17         d_ugly = {"x12": ["($12,000)"], "x62": ["67%"]}
18         d_clean = {"x12": [12000.0], "x62": [67.0]}
19         df_cleaned = pd.DataFrame(data=d_clean)
20         df = pd.DataFrame(data=d_ugly)
21         df_not_ugly = clean_final_dataset(df)
22         result = df_not_ugly.equals(df_cleaned)
23         self.assertTrue(result)
24
25
26 class test_create_final_dataset(unittest.TestCase):
27     def testing_create_dataset(self):
28         """
29         test cleaning final dataset. I created this unit test to verify the dummy variables were being created by taking a sample dataset run the function
30         and verifying it against a correct response
31         """
32         df_test = pd.read_csv("../test/create_test_db.csv")
33         df_right_answer = pd.read_csv("../test/df_final_test.csv")
34         df_test_final = clean_final_dataset(df_test)
35         result = df_test_final.equals(df_right_answer)
36         self.assertTrue(result)
37
38
39 class test_query_example(unittest.TestCase):
40     def test_request_response(self):
41         """
42         run main function and verify the response is okay
43         """
44
45         # Send a request to the API server and store the response.
46         response = requests.get("http://jsonplaceholder.typicode.com/todos")
47
48         # Confirm that the request-response cycle completed successfully.
49         self.assert_true(response.ok)
50
51
52 if __name__ == "__main__":
53     unittest.main()
```

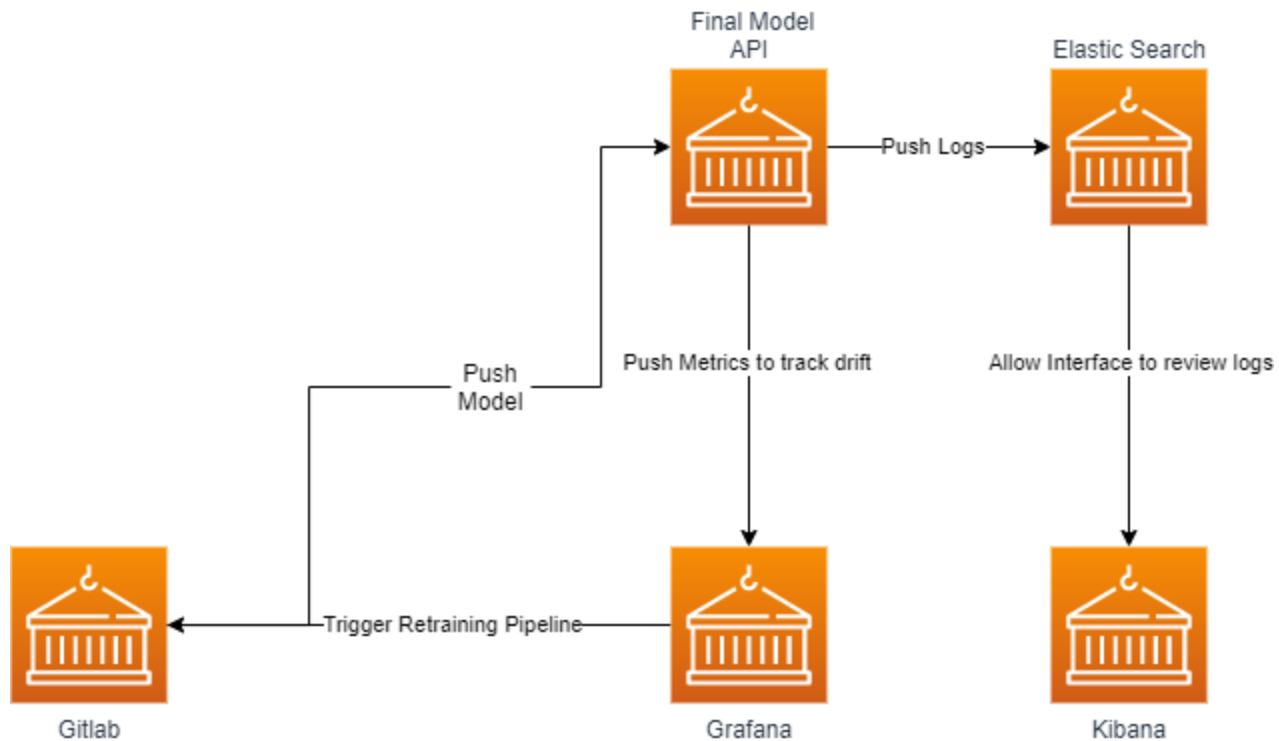
Testing

Tools:

- Unit Test
- Python
- Boto3
- Mock

• Types of Testing

- Unit Testing
- Functional Testing
- Integration Testing



Monitoring and Retraining

- Tools
 - ELK
 - Grafana
 - Prometheus
 - Gitlab API
 - GitOps



Q&A

Citations

1. MLOps. (2021, May 25). Retrieved from <https://en.wikipedia.org/wiki/MLOps>