MATLAB EXPO

DevOps with MATLAB: A Predictive Maintenance System for Streaming Data

Christine Bolliger, PhD
She/Her

Nicole Bonfatti
She/Her

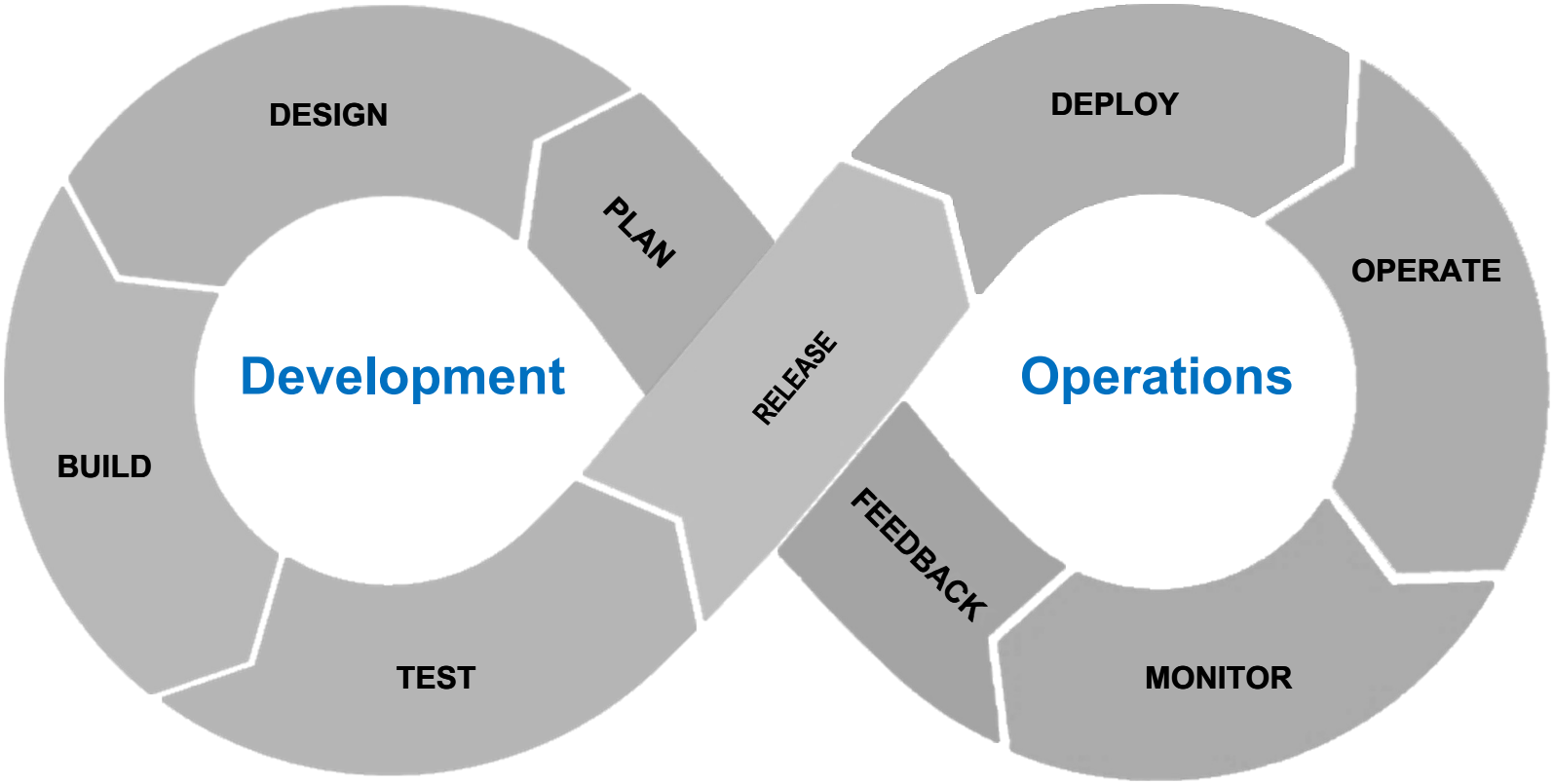
Seth DeLand
He/Him

MathWorks®

Key Takeaways

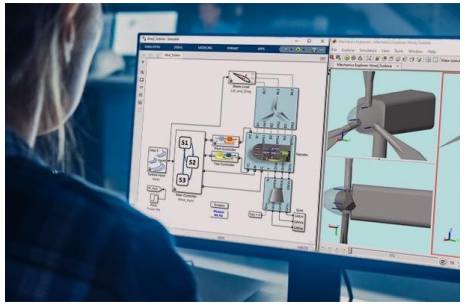
- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations
- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers
- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD

DevOps: Develop and Operate Production Software



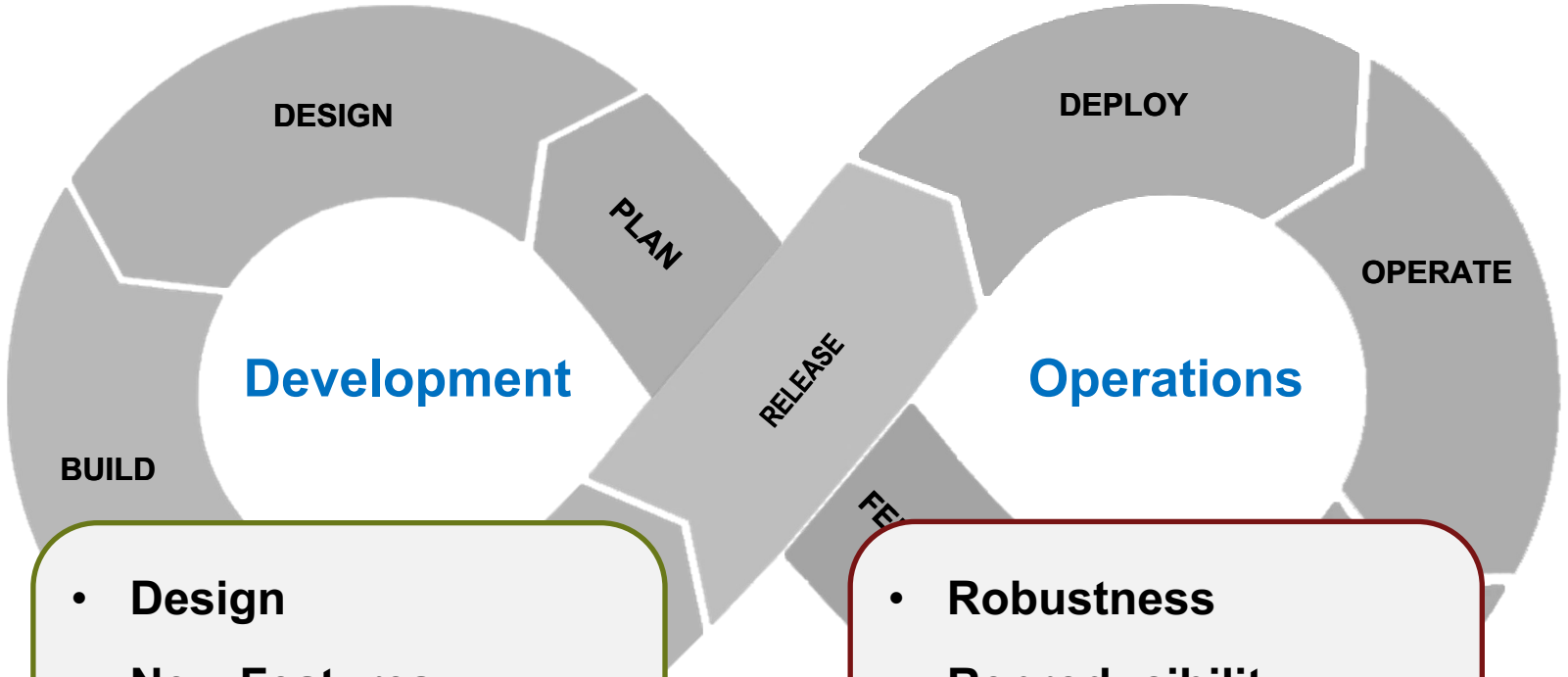
Predictive Maintenance

Financial Modeling



Embedded Controls

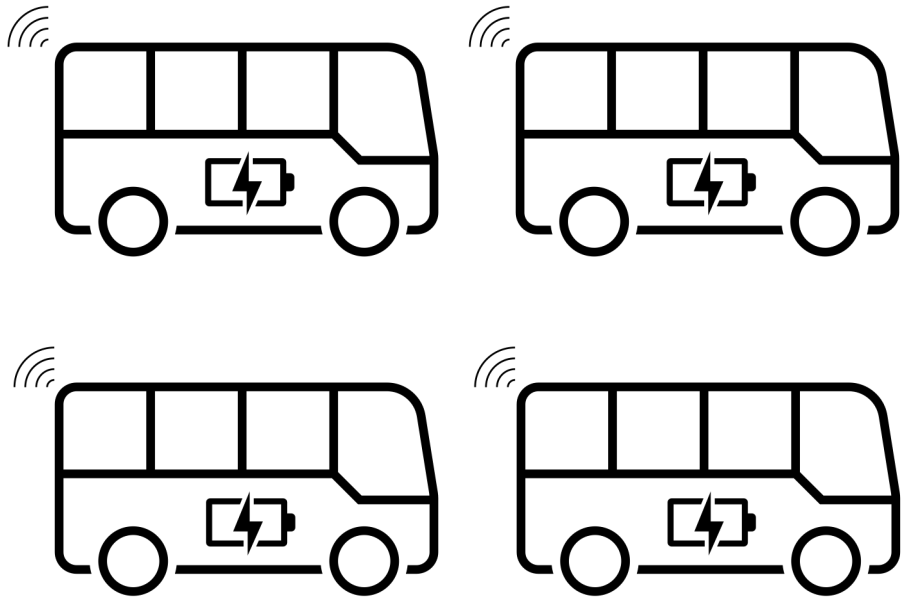
Not necessarily a conflict of interests, but certainly different interests



- Design
- New Features
- Performance
- Testing

- Robustness
- Reproducibility
- Scalability
- Monitoring

Example:
Predicting Battery State-of-Health



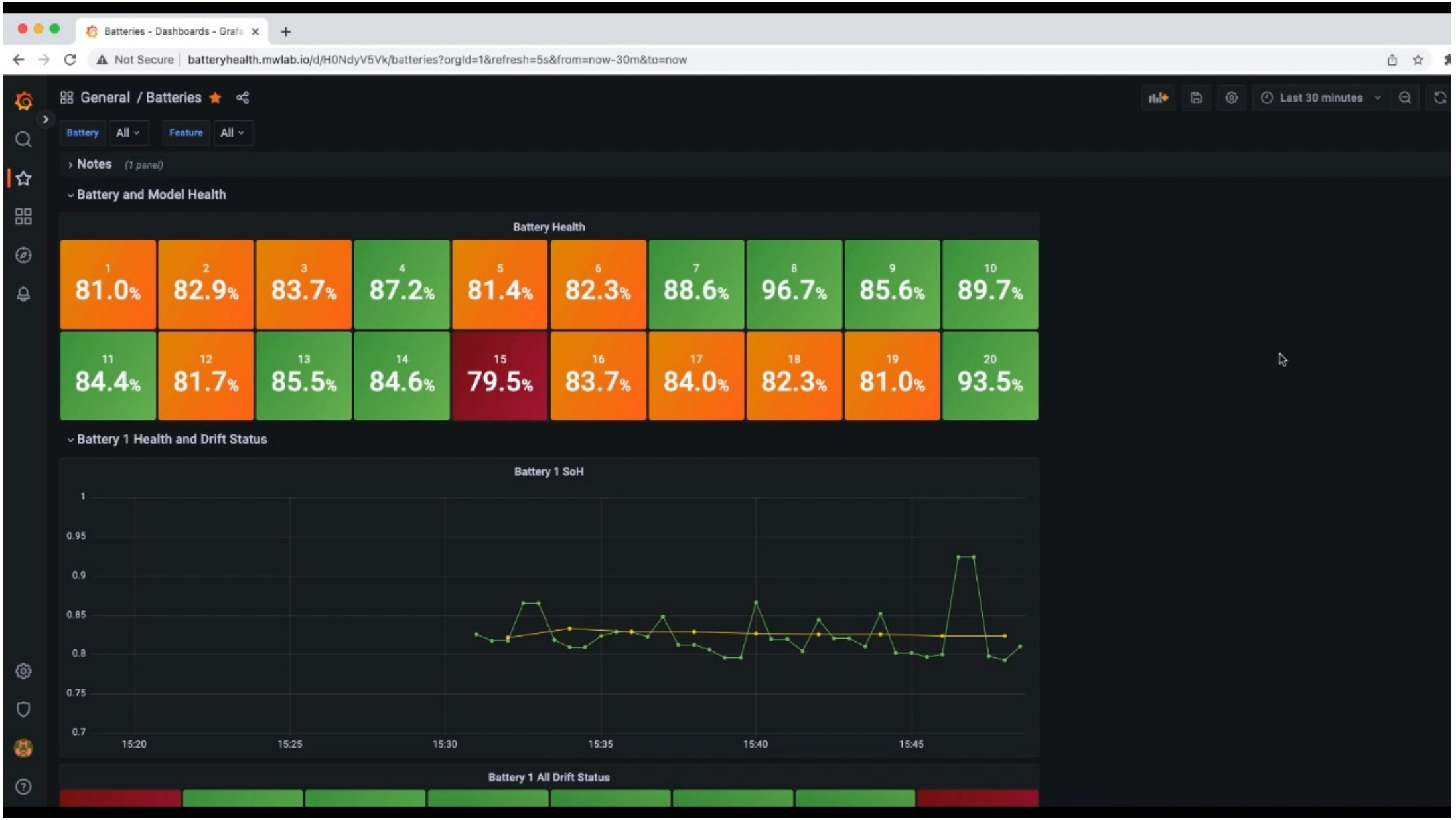
- Fleet of electric buses
- Maintenance is expensive. **Could we do a better job predicting when batteries need replacing?**
- Started gathering telemetry data on batteries

Variables – observations

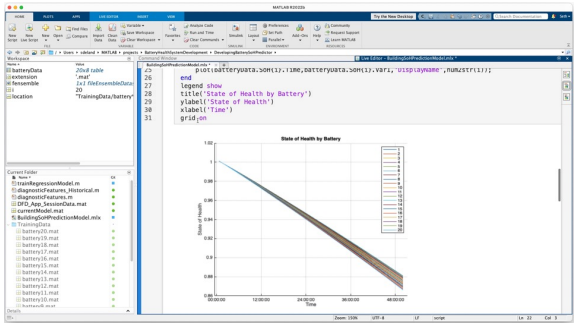
observations

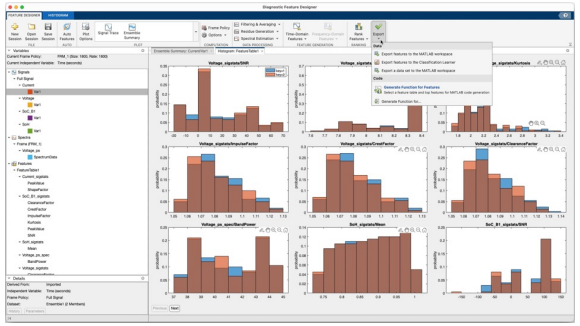
1800x7 timetable

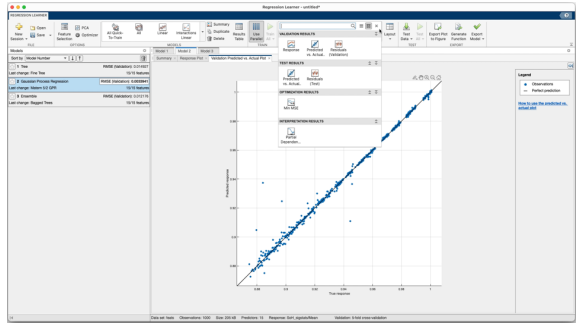
	timestamp	1 Current	2 Voltage	3 Temperature1	4 Temperature2	5 SoC_B1	6 SoC_B2	7 BatteryID
1	01-Nov-2021 00:...	2.6869	7.4436	333.1463	332.7619	0.4995	0.4995	1
2	01-Nov-2021 00:...	2.6872	7.4426	333.1317	332.3924	0.4990	0.4990	1
3	01-Nov-2021 00:...	2.6876	7.4417	333.1073	332.0405	0.4985	0.4985	1
4	01-Nov-2021 00:...	2.6879	7.4408	333.0740	331.7048	0.4980	0.4980	1
5	01-Nov-2021 00:...	2.6882	7.4399	333.0327	331.3844	0.4975	0.4975	1
6	01-Nov-2021 00:...	2.6885	7.4390	332.9843	331.0783	0.4970	0.4970	1
7	01-Nov-2021 00:...	2.6889	7.4381	332.9295	330.7857	0.4965	0.4965	1


Batteries - Dashboards - Grafa
Not Secure
batteryhealth.mwlab.io/d/H0NdyV5Vk/batteries?orgId=1&refresh=5s&from=now-30m&to=now
General / Batteries
Last 30 minutes
Battery
All
Feature
All
Notes (1 panel)
Battery and Model Health
Battery Health
1 81.0%
2 82.9%
3 83.7%
4 87.2%
5 81.4%
6 82.3%
7 88.6%
8 96.7%
9 85.6%
10 89.7%
11 84.4%
12 81.7%
13 85.5%
14 84.6%
15 79.5%
16 83.7%
17 84.0%
18 82.3%
19 81.0%
20 93.5%
Battery 1 Health and Drift Status
Battery 1 SoH
1
0.95
0.9
0.85
0.8
0.75
0.7
15:20
15:25
15:30
15:35
15:40
15:45
Battery 1 All Drift Status

Create a SoH prediction function using domain-specific tools for engineering data and predictive maintenance

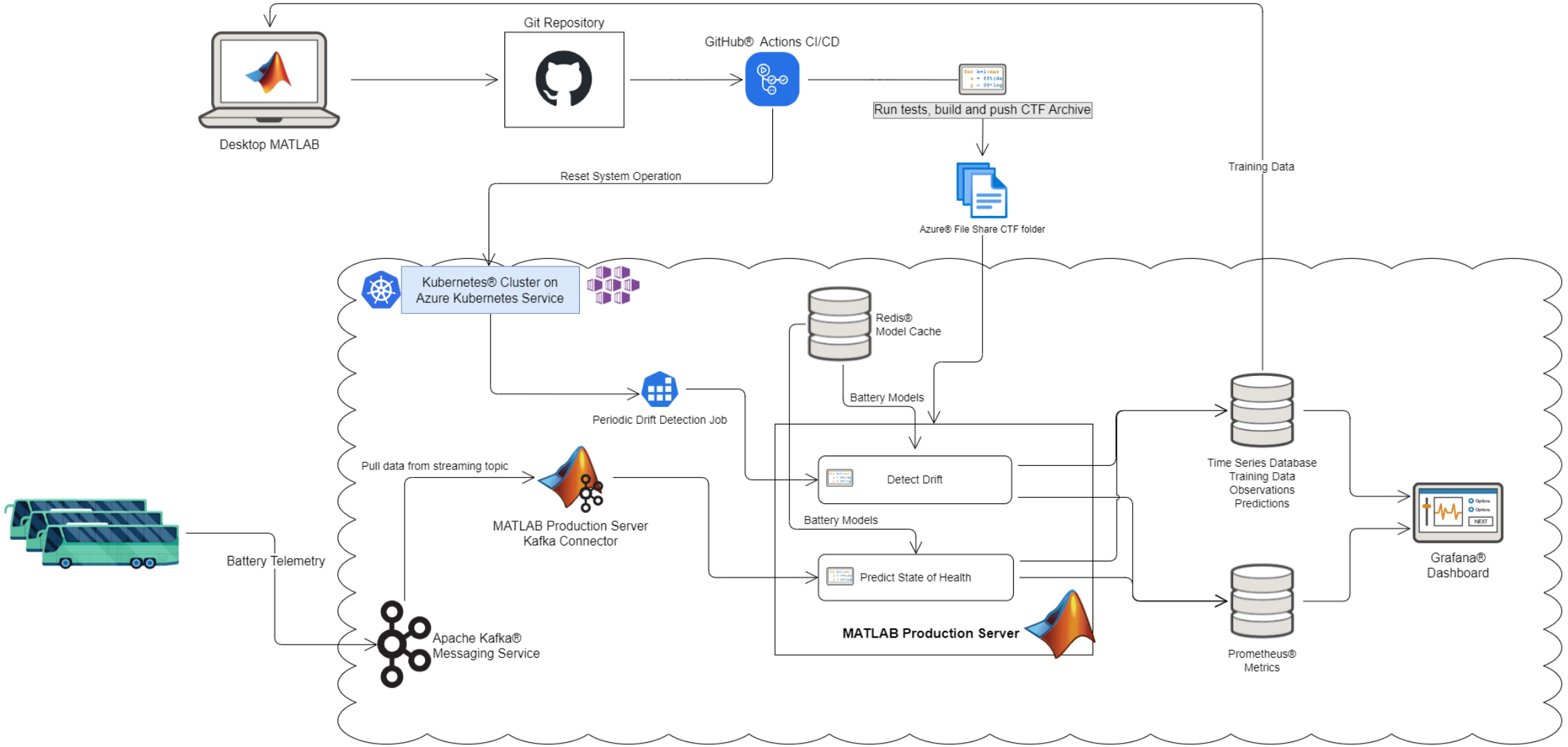






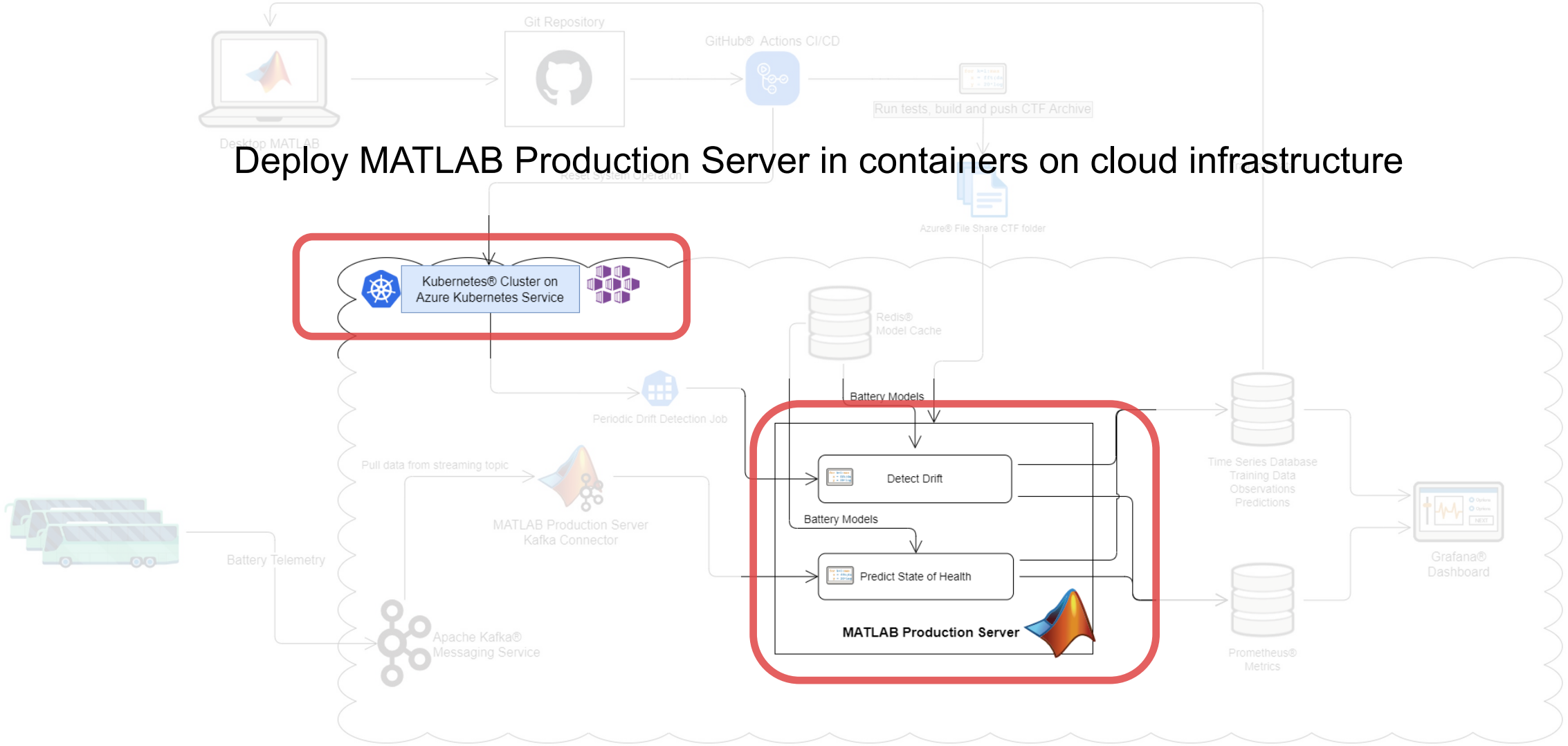
Data Exploration → **Feature Extraction** → **Machine Learning**

A scalable production system running on the cloud, using industry standard tools



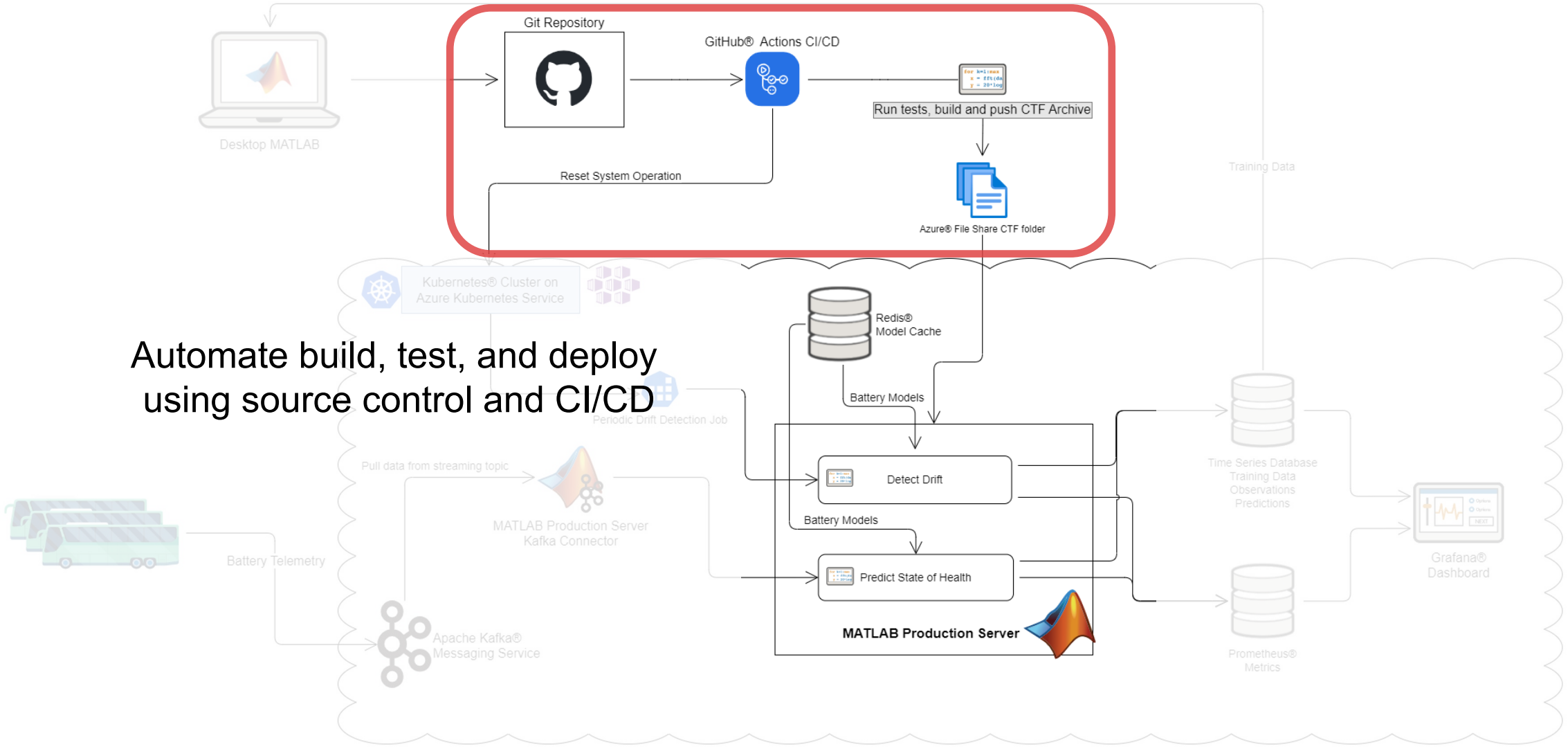
A scalable production system running on the cloud, using industry standard tools

Deploy MATLAB Production Server in containers on cloud infrastructure

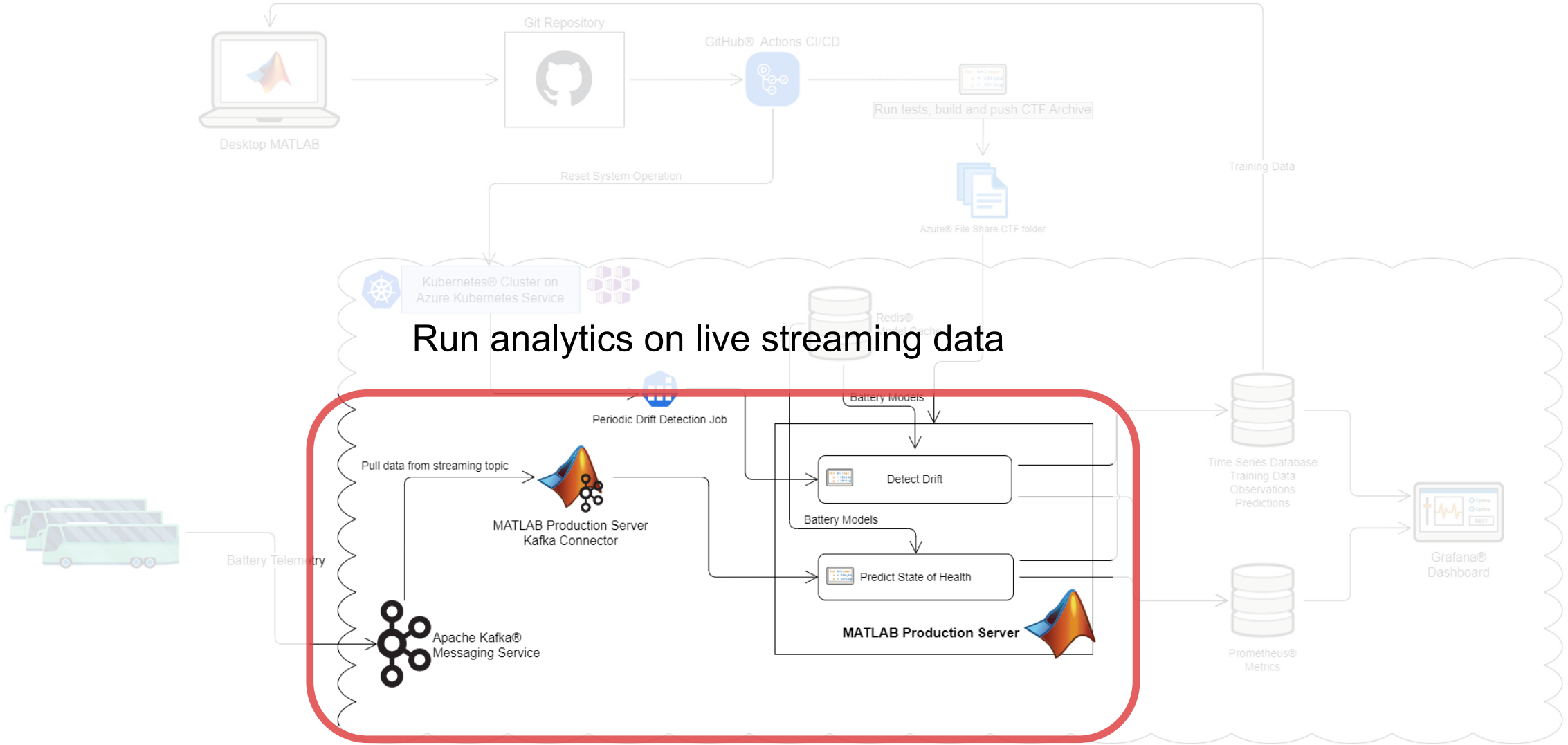


A scalable production system running on the cloud, using industry standard tools

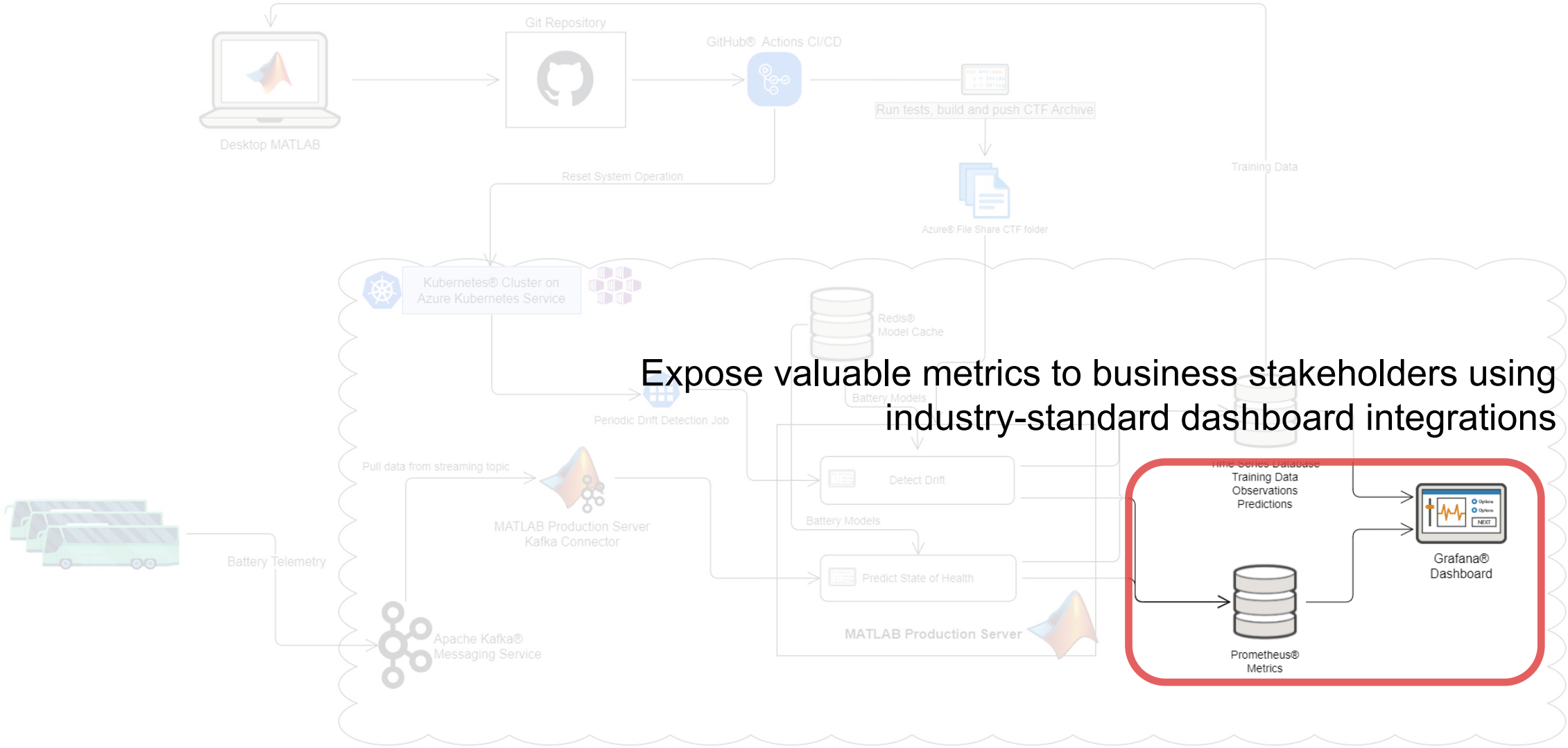
Automate build, test, and deploy using source control and CI/CD



A scalable production system running on the cloud, using industry standard tools



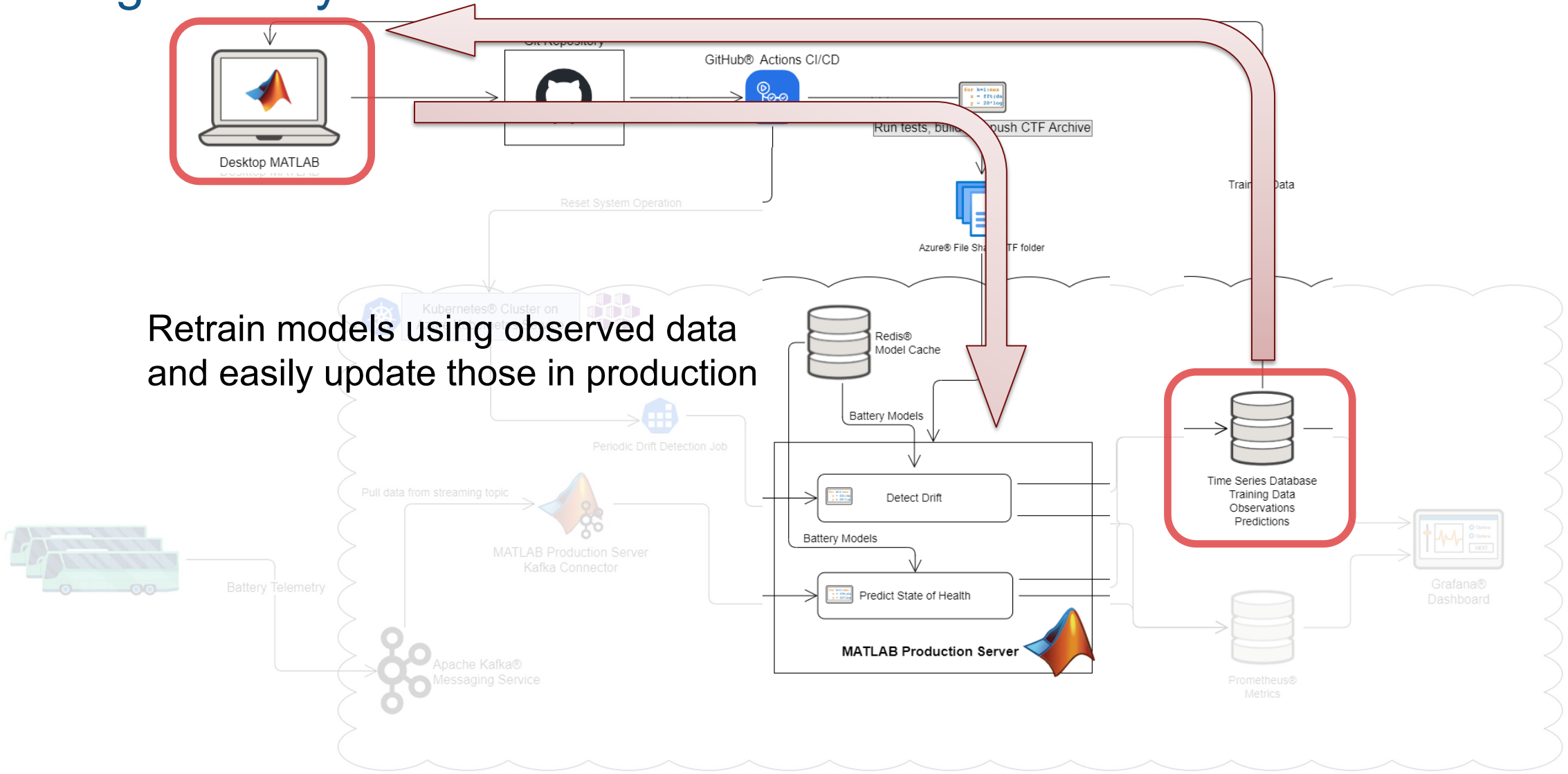
A scalable production system running on the cloud, using industry standard tools



Expose valuable metrics to business stakeholders using industry-standard dashboard integrations

A scalable production system running on the cloud, using industry standard tools

Retrain models using observed data and easily update those in production



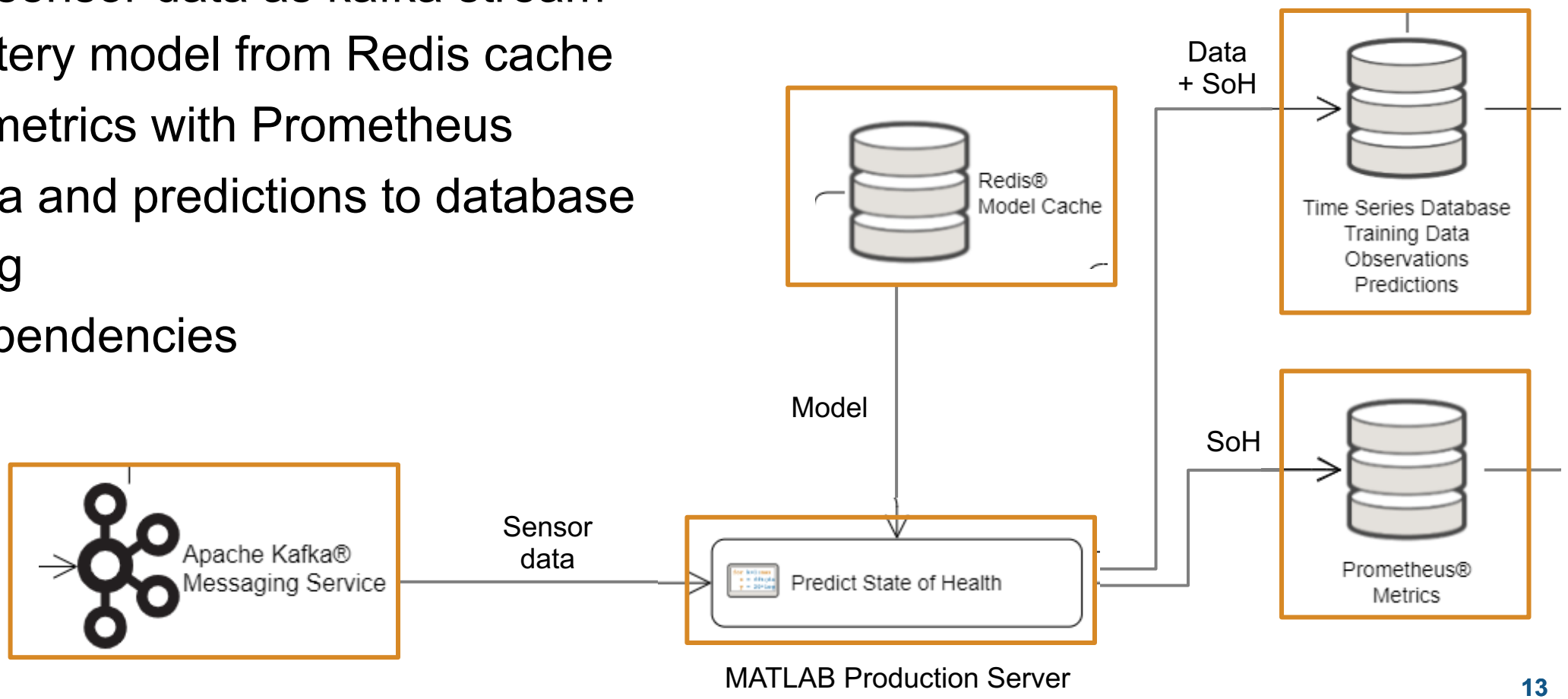
State of health algorithm in production

Production System

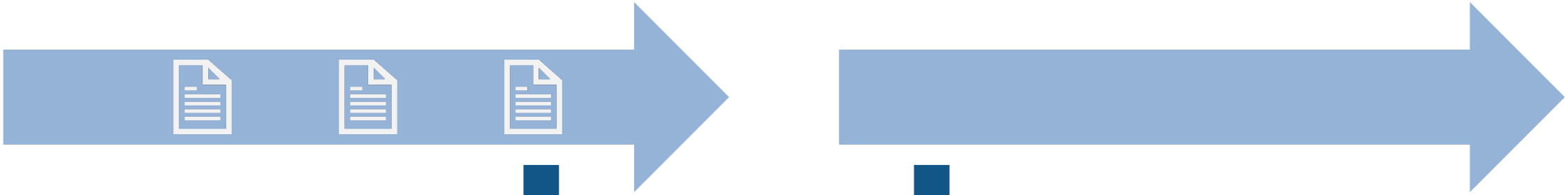
- Receive sensor data as kafka stream
- Load battery model from Redis cache
- Expose metrics with Prometheus
- Save data and predictions to database

Local testing

- Mock dependencies

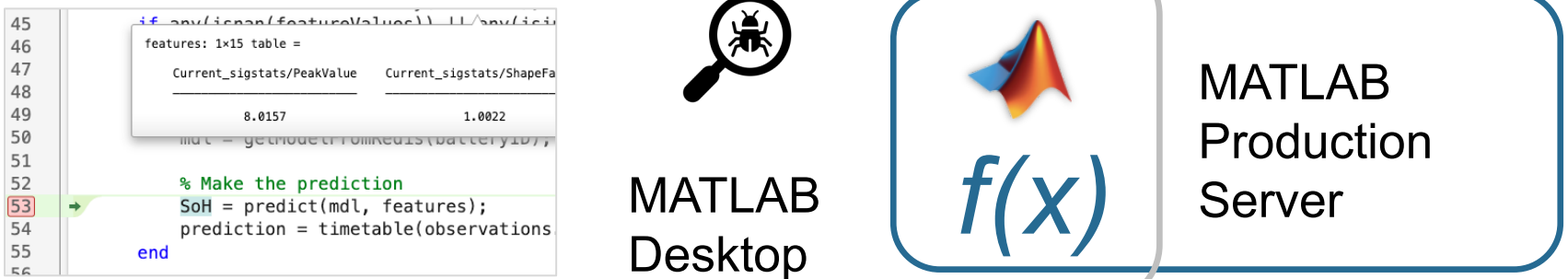


Write SoH prediction function to use kafka streams



Simulate streams with
- `inMemoryStream`
- `testStream`

Use production Kafka streams with `kafkaStream`



Debug locally, then deploy the same MATLAB code to production.

Automatically build, test, package, and deploy MATLAB code

esteinerMW / Battery-Health-Estimation-Streaming-Demo Private

forked from mathworks/Battery-Health-Estimation-Streaming-Demo

Watch 0 Fork 2 Star 1

Code Pull requests Actions Projects Wiki

Build and upload deployable archive (CTF) to MATLAB Production Server

update battery dashboard #19

Summary

Jobs

build

Reset Demo Operations

Run details

Usage

Workflow file

build

succeeded 12 hours ago in 3m

Set up job

Run actions/checkout@v3 3s

Setup MATLAB Support Packages 0s

Setup MATLAB

Run MATLAB buildtool

1 ▶ Run matlab-actions/run-build@v1

8 ▶ Generate script

10 ▶ Run command

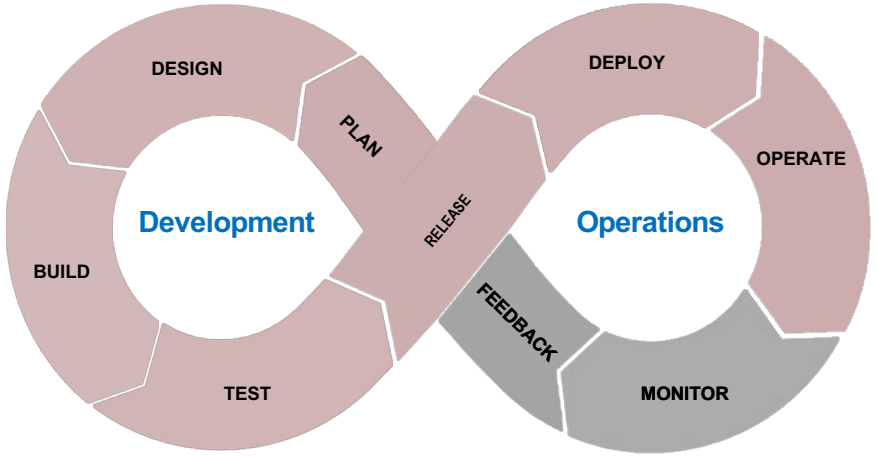
Azure login

Azure CLI script - upload CTF to az file-share

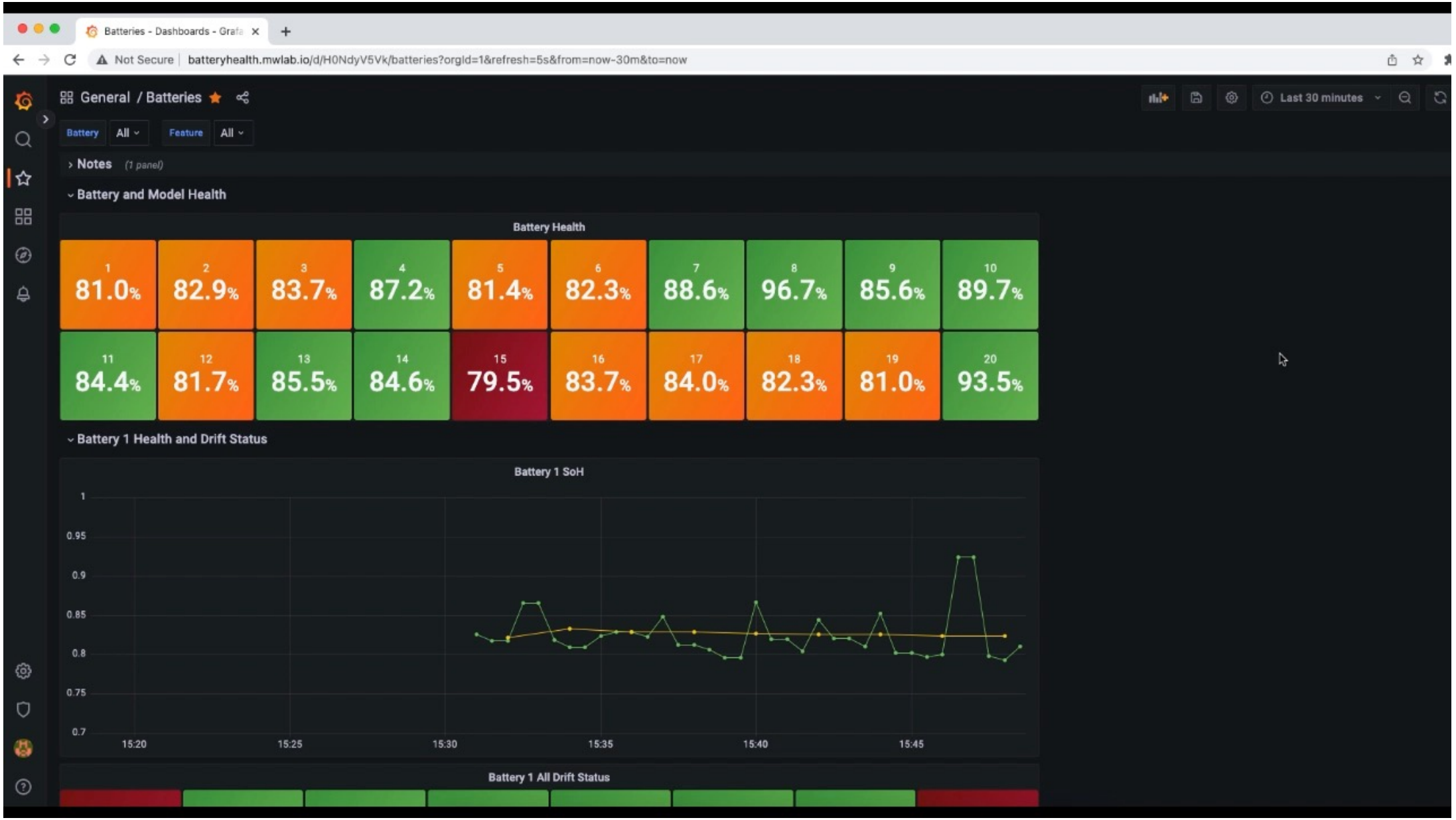
Post Run actions/checkout@v3

Complete job

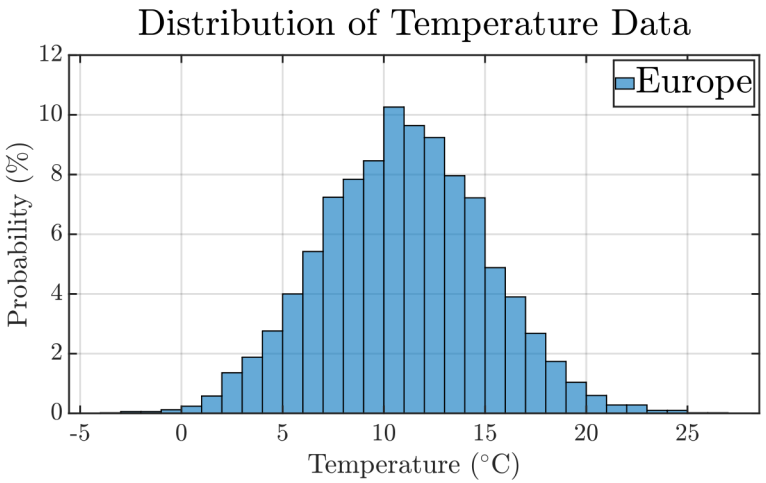
```
- name: Run MATLAB buildtool
uses: matlab-actions/run-
build@v1
with:
tasks: packageDriftDetection
packageSoHPrediction
```

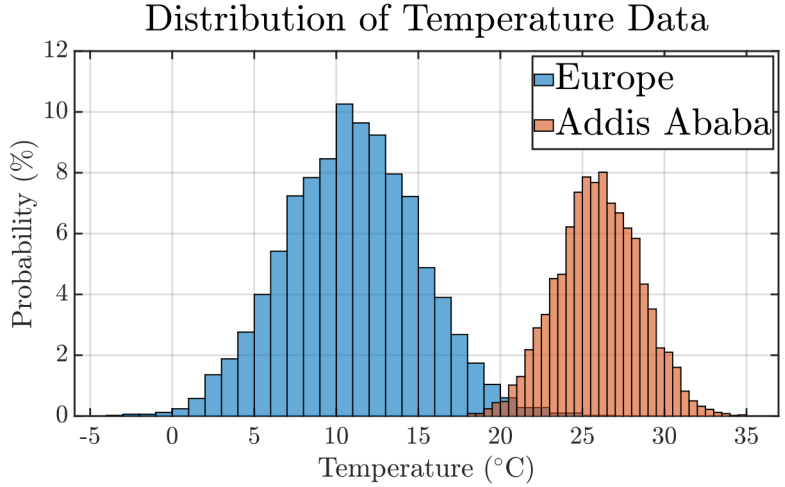


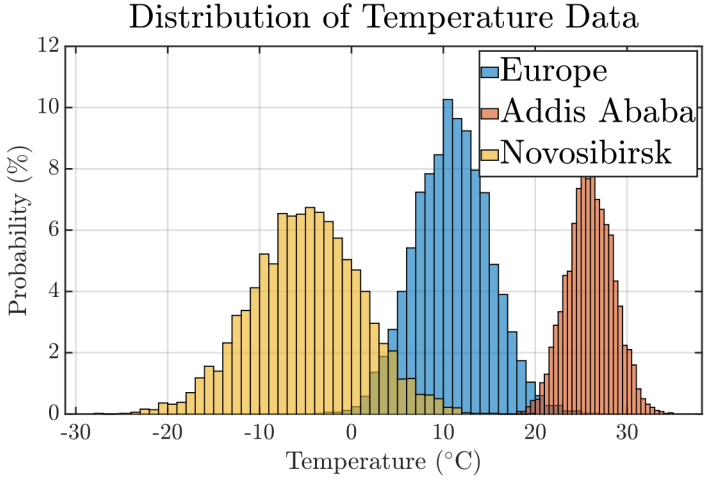

```
function plan = buildfile
plan = buildplan(localfunctions);
plan("packageDriftDetection").Dependen
cies = "test";
plan("packageSoHPrediction").Dependenc
ies = "test";
plan("test").Dependencies =
"validate";
```


Batteries - Dashboards - Grafa
Not Secure
batteryhealth.mwlab.io/d/H0NdyV5Vk/batteries?orgId=1&refresh=5s&from=now-30m&to=now
General / Batteries
Last 30 minutes
Battery
All
Feature
All
Notes (1 panel)
Battery and Model Health
Battery Health
1 81.0%
2 82.9%
3 83.7%
4 87.2%
5 81.4%
6 82.3%
7 88.6%
8 96.7%
9 85.6%
10 89.7%
11 84.4%
12 81.7%
13 85.5%
14 84.6%
15 79.5%
16 83.7%
17 84.0%
18 82.3%
19 81.0%
20 93.5%
Battery 1 Health and Drift Status
Battery 1 SoH
1
0.95
0.9
0.85
0.8
0.75
0.7
15:20
15:25
15:30
15:35
15:40
15:45
Battery 1 All Drift Status

The static data assumption rarely holds in the real world

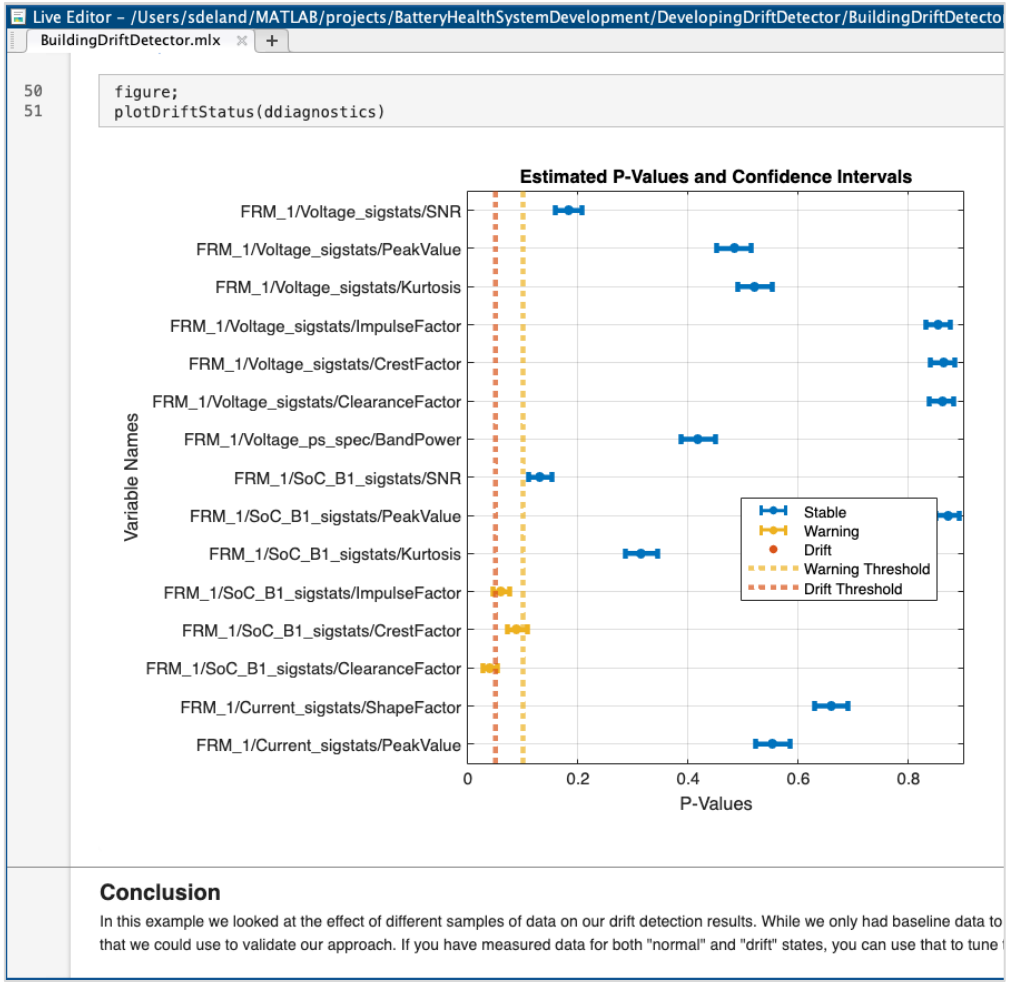






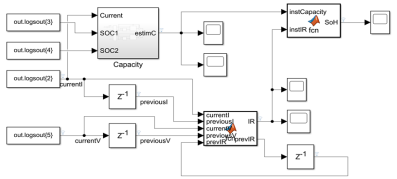
Developing drift detection with `detectdrift`

- Use historical data (training data) to create a baseline distribution
- Generate synthetic data to test for drift
  - This will be replaced by streaming data in the production system

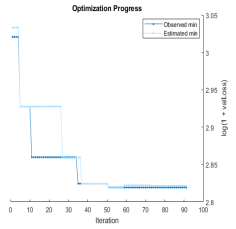


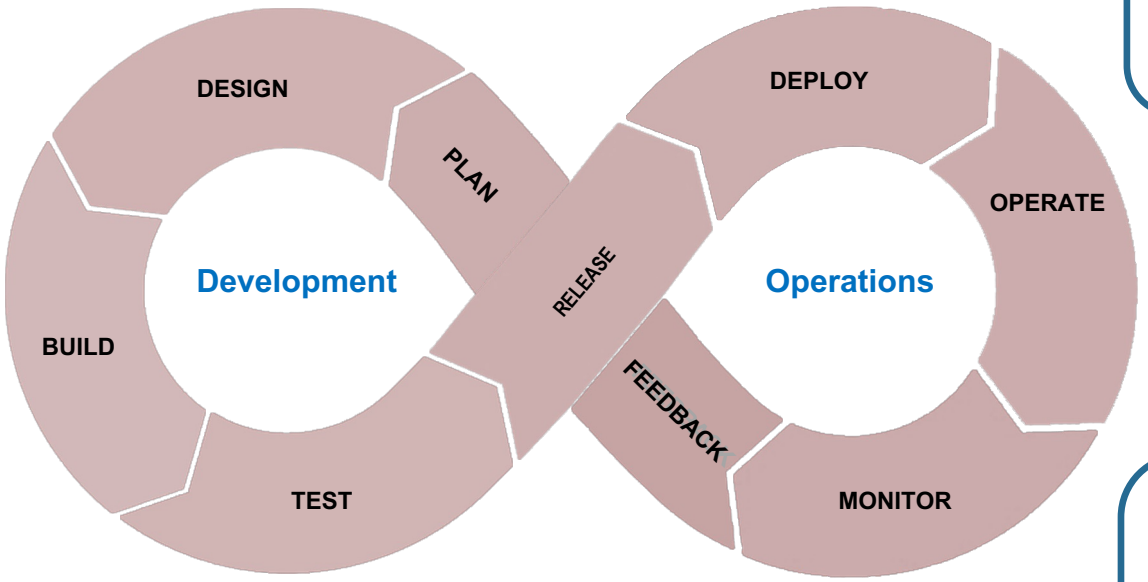
Update model when drift is detected

Data labeling



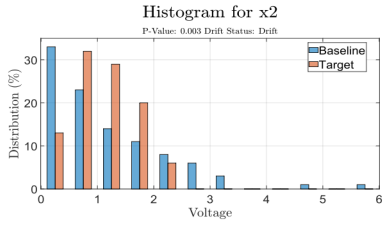
Retrain



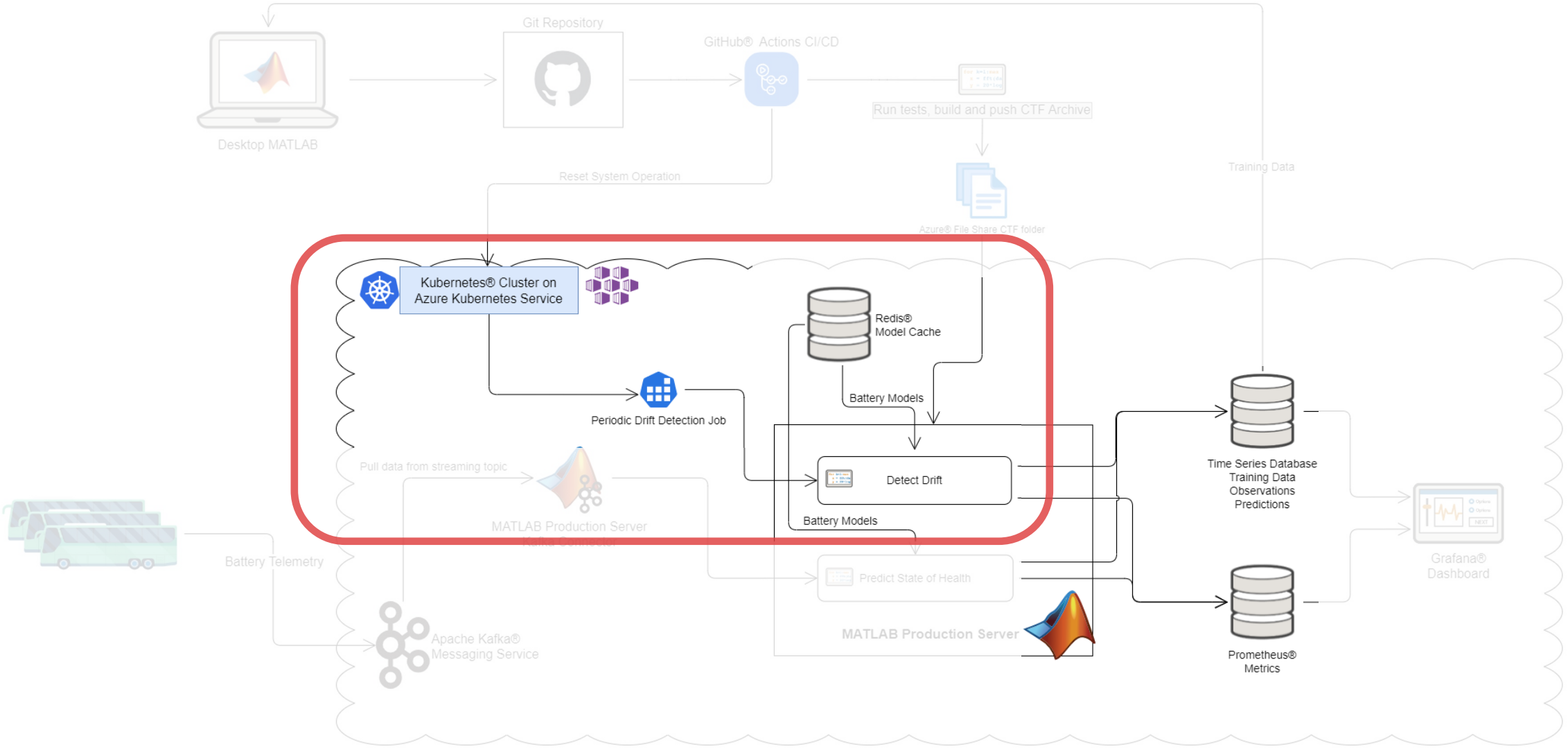


$$\hat{f}_2(x)$$

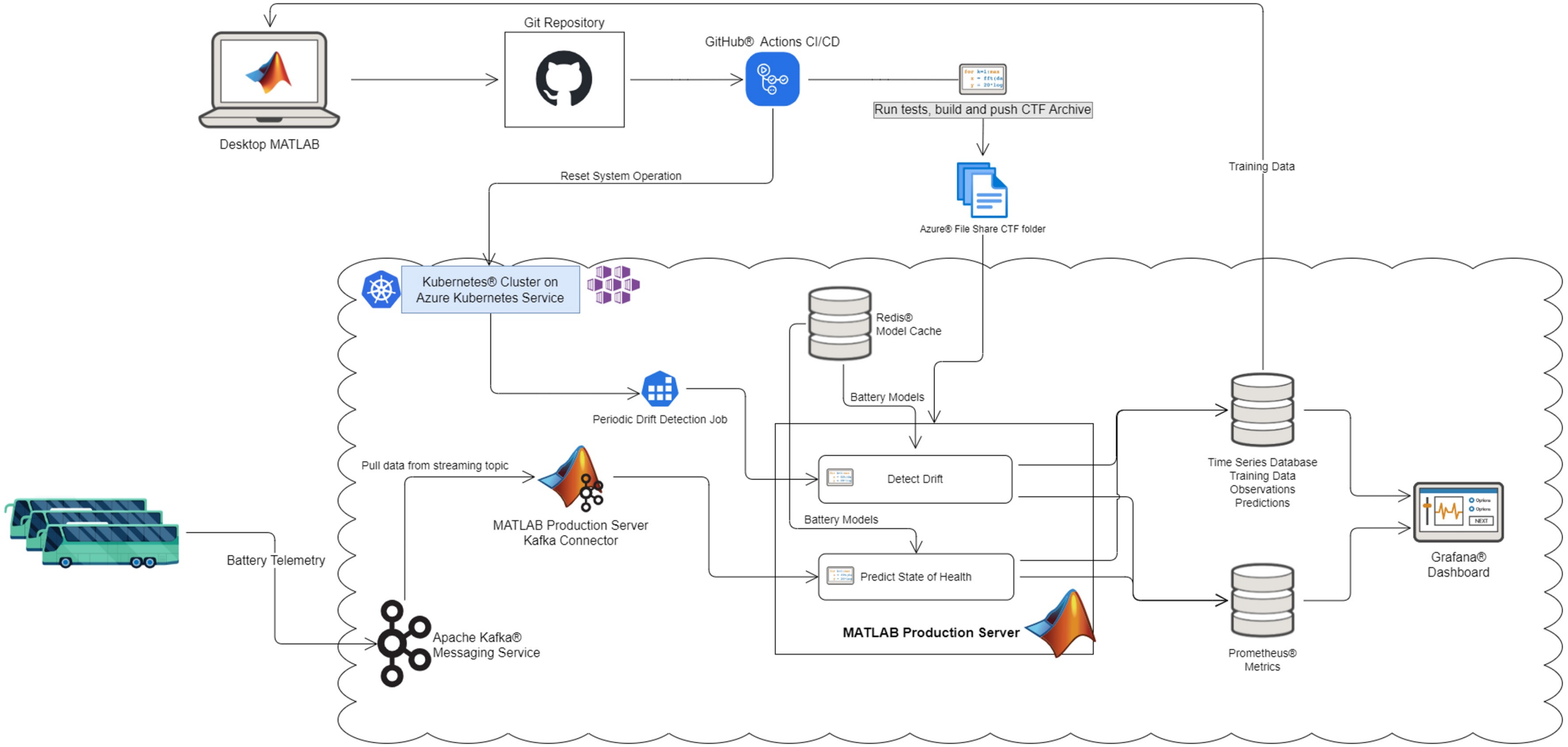
Drift Detection



Update infrastructure to periodically run the drift detection function



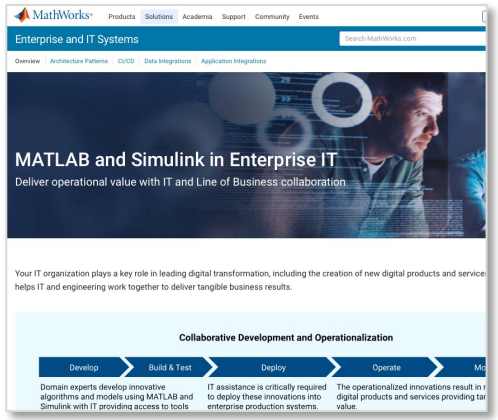
The Complete System



Key Takeaways

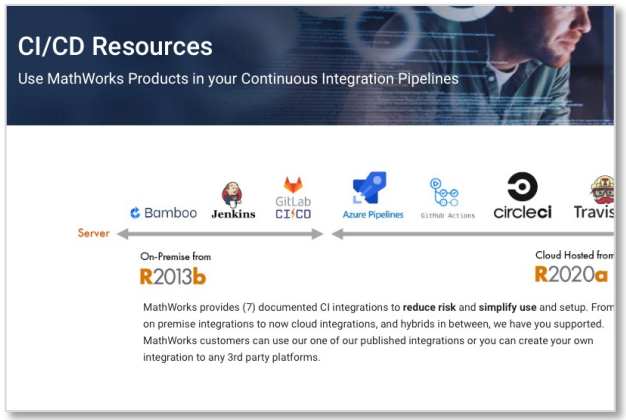
- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations
- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers
- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD

Learn More



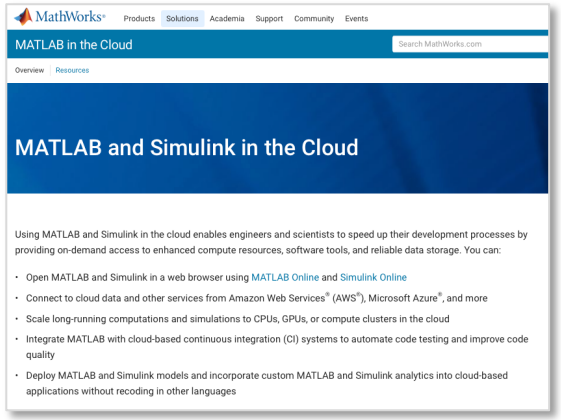
How MATLAB and Simulink are used with Enterprise IT

https://www.mathworks.com/solutions/enterprise-it-systems.html



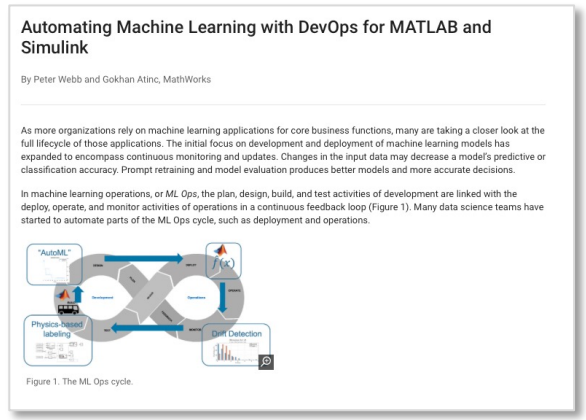
CI/CD Resources

https://www.mathworks.com/solutions/enterprise-it-systems/ci-cd.html



MATLAB and Simulink in the Cloud

https://www.mathworks.com/solutions/cloud.html



Automating Machine Learning with DevOps for MATLAB and Simulink

https://www.mathworks.com/company/newsletters/articles/automating-machine-learning-with-devops-for-matlab-and-simulink.html

Questions?

Attributions

- Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the Apache Software Foundation. The Apache Software Foundation has no affiliation with and does not endorse the materials provided at this event.
- The Grafana Labs Marks are trademarks of Grafana Labs, and are used with Grafana Labs' permission. We are not affiliated with, endorsed or sponsored by Grafana Labs or its affiliates.
- Microsoft, Azure, Azure Kubernetes Service, GitHub, GitHub Actions, and their associated logos are trademarks of the Microsoft group of companies.
- Prometheus, Kubernetes, and their associated logos are registered trademarks of The Linux Foundation.
- Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by MathWorks is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and MathWorks