

[KUBERNETES](#) / [NETWORKING](#) / [STORAGE](#)

# Design an Edge System for the Cloud Native Edge Infrastructure

2 Oct 2020 10:54am, by [Janakiram MSV](#)

In the [previous article](#), I discussed how Rancher's [K3s](#) lightweight Kubernetes distribution, [Calico](#) networking software, and the [Portworx](#) open source cloud native storage platform become the foundation for modern artificial intelligence (AI) and IoT systems that run at the edge. Let's design and deploy a solution that runs on this infrastructure.

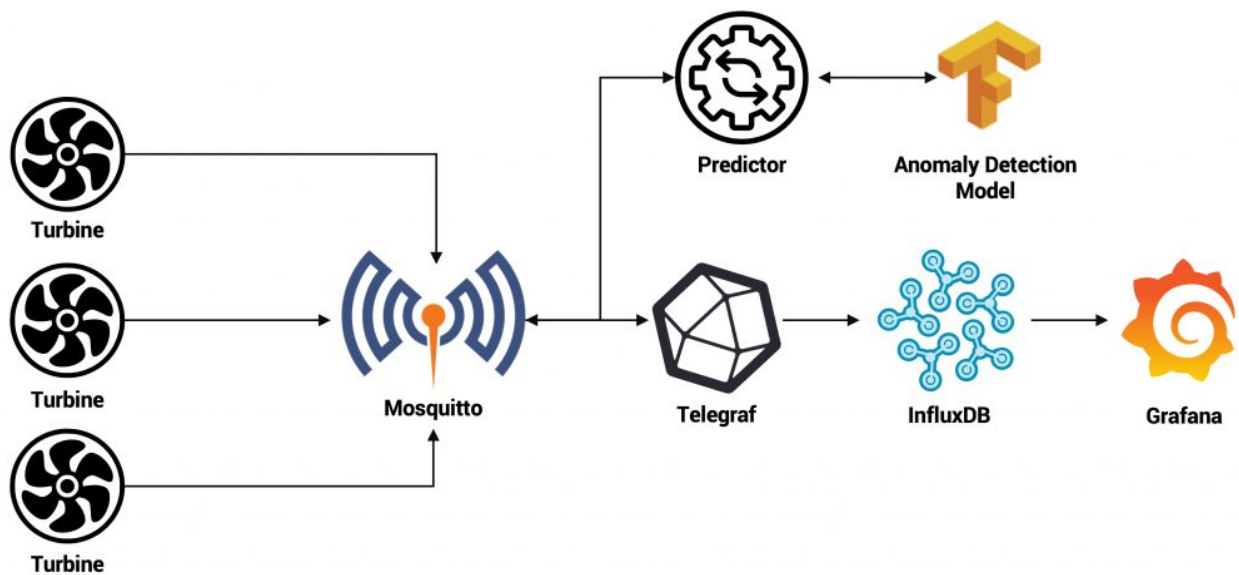
Based on a hypothetical scenario of monitoring fans belonging to turbines, we will build a predictive maintenance solution that will detect anomalies in fans. This acts as a reference architecture for designing and architecting an IoT/edge solution that leverages various open source and the cloud native technologies.

## Problem Statement

We are expected to design and deploy a solution that can ingest telemetry data from multiple fans and use the real-time stream to predict failures before they occur. The

## Solution Architecture

The sensors attached to the fans of the turbine provide the current rotational speed, vibration, temperature, and noise level. This telemetry data stream along with the deviceID from each fan acts as the input to the predictive maintenance solution.



[Mosquitto](#), a popular open source MQTT broker acts as the gateway for the sensors and a centralized message broker for the platform. The sensors ingest the telemetry data into the `fan/messages` topic of the Mosquitto broker.

Below is the payload published by each fan to the MQTT topic.

Architecture

Development

Operations

```
    "rotation": 528.5,  
    "temperature": 35.9,  
    "vibration": 191.7,  
    "sound": 46.8  
}
```

The predictor microservice is a subscriber to the same telemetry channel to which the fans are publishing. For each inbound data point, it invokes the anomaly detection service and publishes the result into a separate MQTT topic, `fan/anomaly`.

```
1 import time  
2 import requests  
3 import random  
4 import datetime  
5 import json  
6 import os  
7 import paho.mqtt.client as mqtt  
8  
9 broker_address = os.getenv('MQTT_HOST')  
10 dev_topic = os.getenv('MQTT_DEV_TOPIC')  
11 pred_topic = os.getenv('MQTT_PREDICT_TOPIC')  
12 scoring_url=os.getenv('SCORING_URL')  
13 d={}  
14  
15 client = mqtt.Client("pdm")  
16 client.connect(broker_address)  
17  
18 def on_message(mosq, obj, msg):  
19     rotation=json.loads(msg.payload)["rotation"]  
20     temperature=json.loads(msg.payload)["temperature"]  
21     vibration=json.loads(msg.payload)["vibration"]  
22     sound=json.loads(msg.payload)["sound"]  
23     telemetry=[rotation,temperature,vibration,sound]  
24     data={"params":telemetry}  
25  
26     response = requests.post(scoring_url, json=data)  
27     fault=json.loads(response.text)["fault"]
```

Architecture

Development

Operations

```
32     payload = json.dumps(u, ensure_ascii=False)
33     print(payload)
34     client.publish(pred_topic, payload)
35
36 def on_subscribe(mosq, obj, mid, granted_qos):
37     print("Subscribed: " + str(mid) + " " + str(granted_qos))
38
39 client.on_message = on_message
40 client.on_subscribe = on_subscribe
41 client.connect(broker_address)
42 client.subscribe(dev_topic, 0)
43
44 while True:
45     client.loop()
```

The `SCORING_URL` is an endpoint of the anomaly detection inference service. A deep learning model trained in TensorFlow is exposed through a Flask web service.

Below is the payload published to the MQTT topic by the predictor service:

```
{
  "deviceId": "1",
  "fault": "0"
}
```

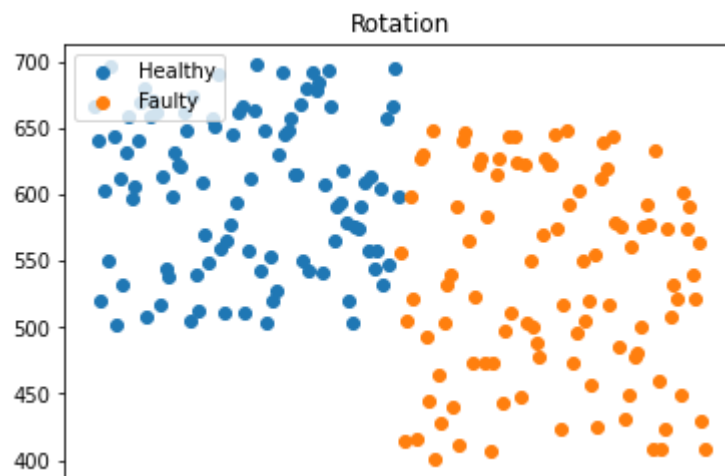
## Training the Anomaly Detection Model

A historical dataset with over 20,000 data points is used to train the anomaly detection model.

Architecture		Development		Operations		
6	15/06/16 11:56	556.6	47.6	154	59.9	1
6	04/05/16 9:30	414.2	52.5	200.9	59.8	1
3	03/01/16 2:13	602.8	44.8	151.3	45.2	0
5	12/01/16 21:59	505.6	51.2	136.1	55.3	1
2	03/01/16 20:23	549.5	36	151	45.7	0
2	01/01/16 4:50	695.8	41.3	108	48	0
5	14/07/16 0:08	598.9	43.1	218.2	49.2	1

From the dataset, it is observed that a fan's rotation decreases hours before failing. Along with that, the vibration, sound, and temperature values increase indicating impending failure.

A scatter plot of the rotation data shows this visually. The RPM of a fan falls to 400 from the normal average of 600.



Based on this, we can easily train a simple TensorFlow logistic regression model to predict the faulty fan. We start by getting rid of the timestamp and the *deviceID* column.

```
1 dataframe = pandas.read_csv("../data/fan.csv", header=None, skiprows=1)
2 del dataframe[0]
3 del dataframe[1]
```

The dataset is then split into train and test data after separating the features and label.

```
1 dataset = dataframe.values
```

we then create a neural network with 4 layers that does logistic regression.

```
1 model = Sequential()
2 model.add(Dense(60, input_dim=4, activation='relu'))
3 model.add(Dense(30, activation='relu'))
4 model.add(Dense(10, activation='relu'))
5 model.add(Dense(1, activation='sigmoid'))
6 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
7 model.fit(X_train, y_train, epochs=250, batch_size=32, verbose=0)
```

Finally, the model is saved and evaluated.

```
1 model.save("../model")
2 loss, acc = model.evaluate(X_test, y_test, verbose=0)
3 print('Test Accuracy: %.3f' % acc)
```

The TensorFlow model saved to the disk is loaded by the inference service to perform predictions on the data sent by the predictor microservice.

## Time-Series Data and Visualization

An instance of [InfluxDB](#) is connected to Mosquitto via [Telegraf](#). This configuration gives us an elegant mechanism of ingesting time-series data into InfluxDB without writing a line of code.

Below is the Telegraf configuration that bridges Mosquitto with InfluxDB.

```
1 [agent]
2     interval = "10s"
3     round_interval = true
4     metric_batch_size = 1000
5     metric_buffer_limit = 10000
6     collection_jitter = "0s"
7     flush_jitter = "0s"
8     debug = false
9     quiet = false
10    hostname = ""
11    omit_hostname = true
12
13    [[outputs.influxdb]]
14
15    urls = ["http://influxdb:8086"]
```

Architecture

Development

Operations

```

20
21  [[outputs.file]]
22  files = ["stdout"]
23  data_format = "influx"
24
25  [[inputs.mqtt_consumer]]
26    servers = ["tcp://mosquitto:1883"]
27    qos = 0
28
29    topics = [
30      "fan/#"
31    ]
32
33    insecure_skip_verify = true
34    client_id = ""
35    data_format = "json"
36    name_override = "fan"
37    tag_keys = ["deviceId"]
38    json_string_fields = ["rotation", "temperature", "vibration", "sound", "fault"]

```

The time-series data can be now queried from InfluxDB.

```

Connected to http://localhost:8086 version 1.6.6
InfluxDB shell version: 1.6.6
> use fan
Using database fan
> select deviceId, rotation, vibration, rotation, temperature, fault from fan
1601549174503260325 2
1601549174563537363 1      536.2      179.4      536.2      44.7
1601549174655780261 1
1601549176404204481 2      415.9      237.1      415.9      43.6
1601549176467849871 2
1601549176563850578 1      660.2      186.2      660.2      38
1601549176633265091 1
1601549178404990632 2      404.4      199.3      404.4      41.3
1601549178478653370 2
1601549178566136662 1      504.6      178.8      504.6      38.8
1601549178637318588 1
1601549180408122841 2      646.9      226.5      646.9      39

```

Finally, we connect a [Grafana](#) dashboard to InfluxDB to build a beautiful visualization for our AIoT solution.



Architecture

Development

Operations



In the next part of this tutorial, I will discuss the deployment architecture along with the storage and network considerations based on K3s, Calico, and Portworx. Stay tuned.

Janakiram MSV's Webinar series, "Machine Intelligence and Modern Infrastructure (MI2)" offers informative and insightful sessions covering cutting-edge technologies. Sign up for the upcoming MI2 webinar at <http://mi2.live>.

FEATURE

TUTORIAL

THE NEW STACK UPDATE



Architecture

Development

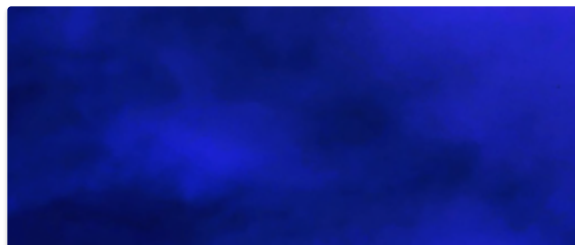
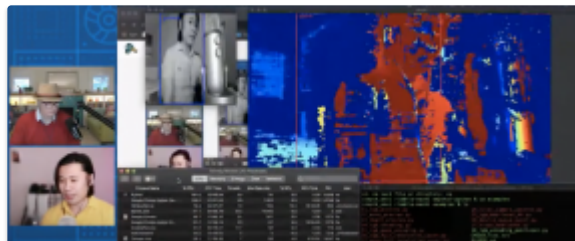
Operations

Email Address

Subscribe

We don't sell or share your email. By continuing, you agree to our [Terms of Use](#) and [Privacy Policy](#).


## RELATED STORIES

CONTAINERS / KUBERNETES / MACHINE  
LEARNING**Accelerating Development with  
Container Run Times,  
Kubernetes and GPUs**12 Mar 2021 9:16am, by [Alex Williams](#)API MANAGEMENT / DEVELOPMENT / MACHINE  
LEARNING / SPONSORED**The Developer's Menu for  
Machine Learning: oneAPI and  
Your New Hardware**3 Mar 2021 9:46am, by [B. Cameron Gain](#)

## Architecture

## Development

## Operations

 OPENSHIFT

OpenShift Commons Briefing: Deterministic vs Probabilistic Security: Leveraging Everything as Code with Steve Giguere (StackRox)

MARCH 12, 2021

---

 **MayaData**

WordPress deployment using OpenEBS on DigitalOcean Kubernetes

MARCH 12, 2021

---

 **HashiCorp**

Boundary Desktop Beta & Boundary 0.1.8 Released

MARCH 12, 2021

---

 **CLOUD NATIVE**  
COMPUTING FOUNDATION

Jaeger persistent storage with Elasticsearch, Cassandra & Kafka

MARCH 12, 2021

---

 **influxdata**

Stargazers and Trailblazers: It's Pi Day!

MARCH 12, 2021

---

 **Vates**  
Solutions Open Source

Xen and the RISC-V Hypervisor Extension

MARCH 12, 2021

---

 **HAPROXY**

Using HAProxy as an API Gateway, Part 5 [Monetization]

MARCH 12, 2021

---

 **Infoblox**  
NEXT LEVEL NETWORKING

## Architecture

## Development

## Operations



5 Ways to level up your remote engineering skills

MARCH 11, 2021



Don't Talk About it, Show it

MARCH 11, 2021



How to Use Channel Assembly with AWS Elemental MediaTailor to Launch Virtual Channels from Existing Sources

MARCH 11, 2021



Exploits on Organizations Worldwide Doubling every Two Hours after Microsoft's Revelation of Four Zero-days

MARCH 11, 2021



Top 9 Takeaways from IstioCon 2021

MARCH 11, 2021



Dynatrace Managed release notes version 1.212

MARCH 11, 2021



RudderStack Supports NetSuite and Xero as Cloud Extract Sources

MARCH 11, 2021

## Architecture

## Development

## Operations

MARCH 11, 2021



Integrating Apache Kafka Clients with CNCF  
Jaeger at Funding Circle Using  
OpenTelemetry

MARCH 11, 2021



Built-in Helm chart scanning with Checkov!

MARCH 11, 2021



Chaos Engineering With Continuous Delivery

MARCH 11, 2021



Don't let supply chain security risks poison  
your organization

MARCH 11, 2021



How open source communities are driving  
5G's future, even within a large government  
like the US

MARCH 11, 2021



Nerdlog Roundup: All Your Hosts, Clusters,  
and Apps in a Single View with New Relic  
Navigator

MARCH 11, 2021



Five Years in the Making

MARCH 11, 2021

## Architecture

## Development

## Operations

MARCH 11, 2021



Live Hoot Episode Recap – Istio Debugging

MARCH 11, 2021



What's New: Updates to Event Intelligence,  
Compliance and Reporting, and More! by  
Vera Chan

MARCH 11, 2021



Review of Four Hyperledger Libraries- Aries,  
Quilt, Ursa, and Transact

MARCH 11, 2021



Forbes names CircleCI to America's Best  
Startup Employers list

MARCH 11, 2021



The CI/CD War of 2021: A Look at the Most  
Popular Technologies

MARCH 11, 2021



Shaping a Career Around Vintage Toy Organs

MARCH 11, 2021



Attackers Won't Stop With Exchange Server.  
You Need a New Playbook

MARCH 10, 2021

## Architecture

## Development

## Operations

MARCH 10, 2021



Performance Monitoring Support for React Native

MARCH 10, 2021



Preventing CSRF Attacks

MARCH 10, 2021



When (& Why) You Should Use Change Data Capture

MARCH 10, 2021



How to Architect Your LogStream to LogStream Data Flows

MARCH 10, 2021



Vulnerability data about open-source software should be open too!

MARCH 10, 2021



Data intelligence is the new oil

MARCH 10, 2021



Announcing Support for DISA STIGs in Puppet Comply

MARCH 10, 2021

## Architecture

## Development

## Operations

MARCH 10, 2021



Streamline ServiceNow testing with a test automation strategy

MARCH 10, 2021



Redis 6.2—the “Community Edition”—Is Now Available

MARCH 09, 2021



How to Establish Your Service Level Objectives (SLOs)

MARCH 09, 2021



Infrastructure as Code Security Through Programmatic Controls

MARCH 09, 2021



Linting Rego with... Rego!

MARCH 09, 2021



Fast Java Made Easy with Quarkus and JHipster

MARCH 07, 2021



API Templates: How to Build a GraphQL API in 3 Easy Steps

MARCH 02, 2021



## Architecture

## Development

## Operations



Reflections on Black History Month at  
LaunchDarkly  
FEBRUARY 25, 2021



ThousandEyes 2021 Predictions: Six  
Forecasts for the New Year  
JANUARY 25, 2021



Why Is APM Important? Breaking Down the  
Benefits  
JANUARY 25, 2021



WSO2 Partner Awards 2020: Recognizing Our  
Extended Family for a Job Well Done  
JANUARY 20, 2021



Protecting Data In Your Cloud Foundry  
Applications (A Hands-on Lab Story)  
JANUARY 13, 2021



Tolly Group confirms Citrix ADC's  
performance leadership  
JANUARY 12, 2021



We Replaced an SSD with Storage Class  
Memory. Here is What We Learned.  
AUGUST 27, 2020

Architecture

Development

Operations

**ARCHITECTURE**

Cloud Native

Containers

Edge/IoT

Microservices

Networking

Serverless

Storage

**DEVELOPMENT**

Cloud Services

Data

Development

Machine Learning

Security

**OPERATIONS**

CI/CD

Culture

DevOps

Kubernetes

Monitoring

Service Mesh

Tools

**THE NEW STACK**

Ebooks

[Architecture](#)

[Development](#)

[Operations](#)

[About / Contact](#)

[Sponsors](#)

[Sponsorship](#)

[Disclosures](#)

[Contributions](#)

---

© 2021 The New Stack. All rights reserved.

[Privacy Policy](#), [Terms of Use](#).