ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA125

INTEGRATING IoT DEVICES INTO THE CERN CONTROL AND MONITORING PLATFORM

B. Copy*, M. Bräger, A. Papageorgiou Koufidis, E. Piselli, I. Prieto Barreiro
CERN, Geneva, Switzerland

Abstract

The CERN Control and Monitoring Platform (C2MON) offers interesting features required in the industrial controls domain to support Internet of Things (IoT) scenarios. This paper aims to highlight the main advantages of a cloud deployment solution, in order to support large-scale embedded data acquisition and edge computing. Several IoT use cases will be explained, illustrated by real examples carried out in collaboration with CERN Knowledge Transfer programme.

INTRODUCTION

C2MON [1] is a monitoring platform developed at CERN and since 2016 made available under an LGPL3 open source license. C2MON is at the heart of the CERN Technical Infrastructure Monitoring (TIM) that supervises the correct functioning of CERN's technical and safety infrastructure. TIM handles about three million messages per day, and serves as the central alarm management system for more than one hundred and fifty thousand alarms. C2MON relies on Java Messaging Services (JMS) [2], as well as caching and clustering technologies, to deliver transactional fail-safe data distribution. C2MON exhibits features specifically targeted at industrial control systems [3]. The Publisher-Subscriber pattern enabled thus, is key to a scalable and robust IoT infrastructure [4]. As exposed in the form of a development roadmap in 2017 [5] and thanks to recent development in cloud technologies, the C2MON deployment model was transitioned to adopt more agile runtime platforms, which had the immediate effect of simplifying the investigation and resolution of complex JMS issues encountered in production. The transition to a cloud deployment model based on Kubernetes [6] also makes C2MON more suitable for instant deployment on commercial hosting platforms such as provided by Amazon or Google.

C2MON ARCHITECTURE OVER KUBERNETES

Kubernetes as a deployment platform [6] provides a clean and efficient abstraction for scaling concerns and orchestration: individual units of process execution (called *pods*) are added or removed as required by runtime health metrics (such as CPU load, memory usage or the presence of critical errors), while process configuration and stateful requirements (such as data persistence or process cluster ordering) are transparently provided to individual pods according to templates. The recent introduction of tools such as Kustomize [7] makes it even simpler to tailor and combine a common set of Kubernetes templates for multiple scenarios, injecting configuration and writing scaling directives in a much more efficient and reproducible manner than traditional cluster management tools such as Ansible [8], due to the fact that the deployment environment is completely factored out and dissociated from hardware and operating system concerns.

* brice.copy@cern.ch

ADAPTING C2MON FOR BETTER INTEGRATION TESTING AND DEPLOYMENT

Simplifying Adoption for C2MON Users

C2MON aims to make it easier for new and existing users to (leverage its monitoring value) by focusing on the following points:

- Shipping a one click deployment.
- Allowing extension and (re)configuration of a running stack.
- Maximizing service resilience and fault tolerance.

One Click Deployment Kubernetes is the project of the Cloud Native Computing Foundation with the highest percentage of production usage, as of October, 2018 [9]. C2MON users can bootstrap a complete stack, such as the one specified below in Fig. 1, using a single command.

Configuring the C2MON Stack Managing the YAML [10] files for a Kubernetes stack can grow to quite a complex task, since the format provides no support for extension, or any form of versioning. YAML is also criticized for being unsafe and producing unexpected behaviors [11]. Addressing some of these issues, Kustomize [7] is an open source tool which provides:

- A declarative approach to configuration management.
- Component reuse capabilities.
- Integration into existing version control workflows.

By complying with Kustomize workflows, the C2MON Kubernetes deployment has been modularized into directories which correspond to layers. Additional configuration options, such as resource properties files can be added in those directories and they are automatically converted into Kubernetes ConfigMaps [12] and Secrets [13], which are then used in the pods. Thus, every component of C2MON software can be customized over the same base image and even reconfigured during runtime by applying a rolling update [14].

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA125

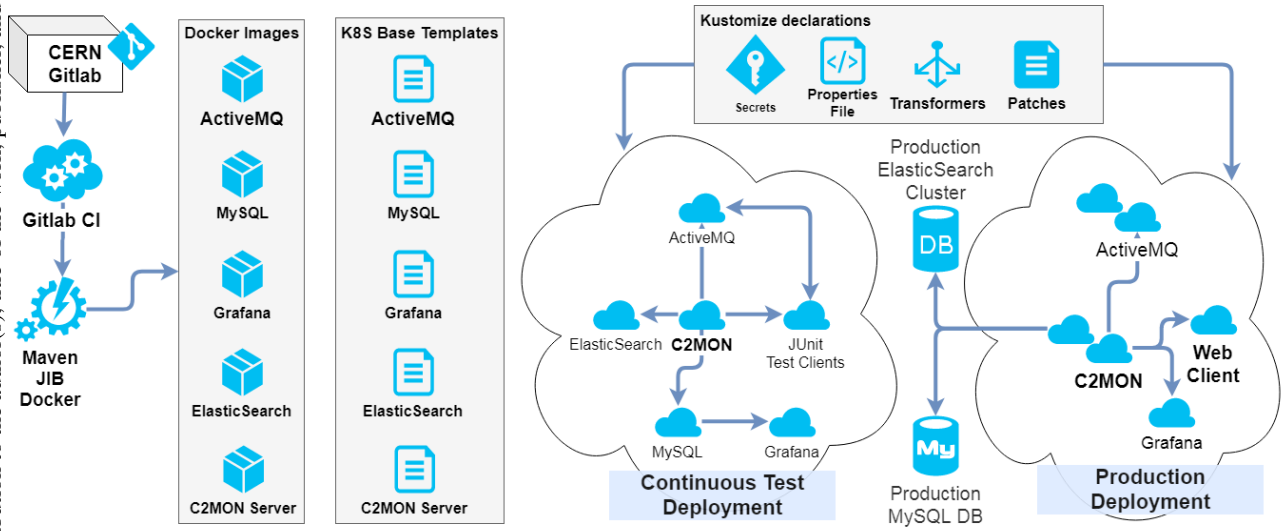

Figure 1: Combining Docker images, Kubernetes templates and Kustomize directives to produce Test and Production deployments, including redundancy and dependency on external services

Enabling GitOps

GitOps [15], is the practice of using a version control system as a single source of truth for managing Kubernetes clusters. The primary objective is to manage deployments using the same modern standardized practices that are applied to the code base, such as pull requests and code reviews. Maintaining all deployment information in the code repository provides a clear and coherent history of deployments, a direct view into important production configuration details and the ability to apply updates in a simple and systematic manner.

BOLSTERING JMS RESILIENCE

Message queue systems, implemented using JMS in Java are increasingly gaining support over REST for handling cloud workloads [16–18]. Proponents of JMS focus on its key strengths:

- **Reliability** as most message queue brokers come with a 100% message delivery guarantee.
- **Elasticity** as producers and consumers are independently scalable without any concurrency issues.
- **Asynchronous processing** which is a feature of paramount importance in an environment where pod crashes are not very common, but even an expected part of workflows. This leads to a more robust architecture, as individual component failures cannot cascade across the system, while high workloads can be managed with Kubernetes autoscaling.

Chaos Engineering in Kubernetes

Chaos engineering is the practice of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production [19]. While some chaos engineering solutions such as chaosmonkey [20] focus on testing during production operations, C2MON focuses on an integration testing workflow, aiming at exposing bugs before they appear in production. Chaos engineering in C2MON is carried out using an in-house framework. Using Kustomize, one can generate multiple Kubernetes environments which examine different scaling and architecture configurations or replicate existing production setups. A series of scenarios are then run on each environment, simulating a selection of failures, such as server crashes or network outages. Concurrently, the framework is running tests to constantly evaluate that the system is exhibiting Steady State Behavior in terms of stability, throughput and quality of service.

Environments and Configurations

The performance of different solutions was examined and evaluated across different configurations ranging from a simple publisher-subscriber architecture to a complex parallel and concurrent C2MON stack. In the case of C2MON, four different message queuing (MQ) configurations were overlaid over three base environments, then subjected to three different scenarios, generating a simulation matrix of thirty-six samples. To monitor the performance of the MQ layer, Prometheus [21], and Prometheus AlertManager [22] were added to the simulation and configured to monitor the Java agents in the system. To evaluate system throughput, Service level agreement (SLA) compliance and system stability, JUnit [23] tests were written and run during the simulations. The chaos engineering framework managed building the overlays and running the scenarios, as well as exporting the results from inside the Kubernetes pods that run the tests, which turned out to be a non-trivial task.

Findings

The simulations successfully and clearly demonstrated the performance and robustness of different configurations, as well as the system's overall response to different conditions.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA125

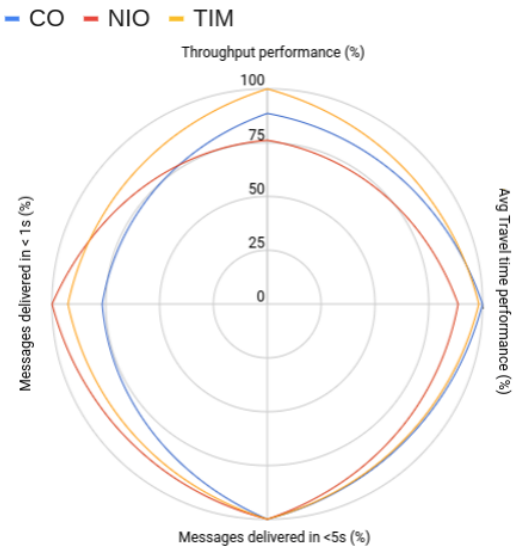

Figure 2: Performance of different ActiveMQ configurations during idle state. Scaled to best performance on each axis. For a more detailed description, see Annex A.

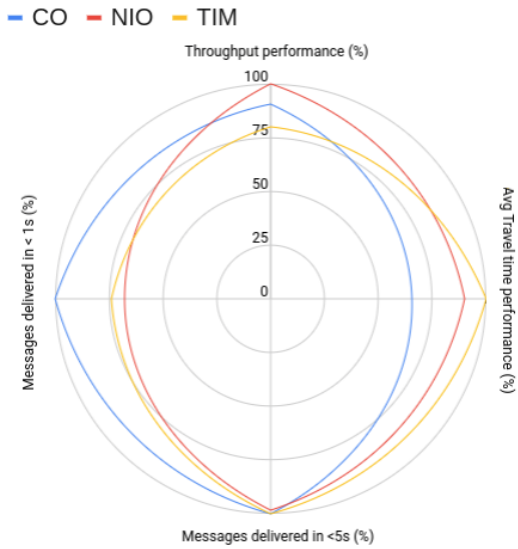

Figure 3: Performance of different ActiveMQ configurations during significant system load. Scaled to best performance on each axis. For a more detailed description, see Annex A.

Figures 2 and 3 visualize some of the findings. The simulations helped to pinpoint sub-optimal configurations that could not easily be diagnosed in a production or even a standard test environment. Additional findings about the issues encountered when running Apache ActiveMQ in a containerized cloud environment, alongside working solutions, were reported to the core Apache ActiveMQ development team.

TRANSITION TO IOT TECHNOLOGIES

C2MON, as a best-of-breed open-source data acquisition and monitoring platform, was initially designed around 2009, when cloud technologies were not as prevalent as today. Emerging IoT technologies have also initiated a shift from traditional industrial controls protocols to more lightweight Internet-friendly ones (such as MQTT [24] or WebSockets [25]).

IOT DEVICES PARTICULARITIES

IoT devices have limited connectivity and runtime resources but allow to perform data acquisition at low-cost. The Arduino eco-system [26] provides a plethora of community-supported sensor libraries that enable the prototyping and productization of new data acquisition devices at record speed.

IoT devices, in comparison to personal computing devices, are characterized by :

- Their operating range which varies from a few meters (for Bluetooth) to kilometers (for cellular or LoRa [27] devices).
- Their low power usage and requirement to run on autonomous power such as batteries or solar panels, in remote places.
- Their built-in support for low CPU standard protocols to ensure wide connectivity.

The openness and wide availability of cloud deployment options offered by C2MON make it an ideal platform for the release and monitoring of IoT devices. In collaboration with the CERN Knowledge Transfer office, the C2MON developers have already taken part in multiple IoT usage scenarios.

USAGES OF C2MON IN IOT SCENARIOS IN PARTNERSHIP WITH THE CERN KNOWLEDGE TRANSFER TEAM

The CERN Knowledge Transfer (KT) team has the mandate to disseminate and productize CERN technology so it can benefit society at large beyond high-energy physics research. CERN KT provides a legal and financial framework for third-party companies to adopt CERN technologies for maximum mutual benefit; as part of the CERN KT portfolio, this applies to C2MON, as illustrated in the two following case studies.

Grocers of the Future – Tracking Fresh Produce from Crop to Shelf with C2MON

As part of the CERN Challenge Based Innovation [28], a team of six university students (team Loop) sponsored by the Italian sustainable grocery packaging producer *CPR Systems*, developed a working tracking service for fresh produce transport crates, using RFID and on board sensors. The tracking service, based on C2MON technology and design principles, enables follow-up of the location, travel time and storage conditions of fruits and vegetables transiting through the transport system. Such detailed information is essential to *CPR Systems* and goods producers, and also a strong sell-

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA125

ing point for consumers that can see on the shelf all relevant information concerning the goods they are buying.

Securaxis – Road Traffic Monitoring in Smart Cities

Securaxis, a Geneva (Switzerland) based startup company, is working to combine edge computing, machine learning and C2MON technology [29] to perform smart monitoring of situations where traditional methods (such as video-surveillance, sound recording) are both too costly and too intrusive to operate. By employing smart sensors to analyze the data and only propagate significant events to a centralized monitoring system, Securaxis has already received funding to monitor road activity on bridges, or monitor the presence and the population of protected species in a given habitat. Technologies such as LoRa [27] and MQTT [24] are particularly relevant to scale up affordable deployments in this context.

CONCLUSION

The usage of Kubernetes [6] and associated technologies to deploy C2MON in test and pre-production environments has already proven an undeniable gain in delivering consistent performance and reliability test results, and will in the future continue to allow development to fine-tune our deployment configurations and technological choices with great precision. The CERN Accelerator infrastructure team is also taking steps to evaluate and provide an operational cloud based on Kubernetes [30], which will allow the CERN TIM service (based upon C2MON technology) to greatly simplify production deployments in the near future.

REFERENCES

[1] M. Bräger, M. Brightwell, E. Koufakis, R. Martini, and A. Suwalska, "High-Availability Monitoring and Big Data: Using Java Clustering and Caching Technologies to Meet Complex Monitoring Scenarios", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper MOPPC140, pp. 439–442.

[2] Java Message Service, https://en.wikipedia.org/wiki/Java_Message_Service

[3] E. Byres, "SCADA Security Basics: Integrity Trumps Availability", http://tiny.cc/scada-reliability, 6 Nov 2012.

[4] S. Vavassori, J. Soriano, and R. Fernández, "Enabling Large-Scale IoT-Based Services through Elastic Publish/Subscribe Sensors", *Sensors*, vol. 17, p. 2148, Sept 2017. doi:10.3390/s17092148

[5] B. Copy *et al.*, "C2MON SCADA Deployment on CERN Cloud Infrastructure", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 1103–1108. doi:10.18429/JACoW-ICALEPCS2017-THBPL01

[6] Kubernetes, https://en.wikipedia.org/wiki/Kubernetes

[7] Kustomize, https://github.com/kubernetes-sigs/kustomize

[8] Ansible, https://en.wikipedia.org/wiki/Ansible_(software)

[9] K. Barnard, "CNCF Survey: Use of Cloud Native Technologies in Production Has Grown Over 200%", http://tiny.cc/cncf-cloud-usage, *Cloud Native Computing Foundation*, 29 Aug 2018.

[10] YAML, https://en.wikipedia.org/wiki/YAML

[11] YAML: probably not so great after all, https://arp242.net/yaml-config.html

[12] Kubernetes ConfigMap, https://cloud.google.com/kubernetes-engine/docs/concepts/configmap

[13] Kubernetes Secret, https://kubernetes.io/docs/concepts/configuration/secret/

[14] Performing a rolling update in Kubernetes, https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/

[15] GitOps, https://www.weave.works/technologies/gitops/

[16] Microservices Messaging: Why REST Isn't Always the Best Choice, http://tiny.cc/rest-messaging

[17] Microservices communications. Why you should switch to message queues, http://tiny.cc/microservices-mq

[18] REST vs Messaging for Microservices – Which One is Best? http://tiny.cc/rest-vs-mq

[19] Principles of Chaos Engineering, https://principlesofchaos.org/

[20] Chaos Monkey, https://github.com/Netflix/chaosmonkey

[21] Prometheus, https://prometheus.io/

[22] Prometheus AlertManager, https://prometheus.io/docs/alerting/alertmanager/

[23] JUnit 5 User Guide, https://junit.org/junit5/

[24] MQTT Version 3.1, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[25] RFC 6455 The Websocket protocol, https://tools.ietf.org/html/rfc6455

[26] What is Arduino?, https://www.arduino.cc/en/Guide/Introduction

[27] LoRa, https://en.wikipedia.org/wiki/LoRa

[28] CERN Challenge Based Innovation, https://www.cbi-course.com/

[29] Securaxis, first Innovaare incubatee of CERN technology http://cern.ch/go/7dzw, Parkinnovaare, Switzerland, Schachen, Aarau, 12 Nov 2018

[30] R. Voirin *et al.*, "Containers in Controls Workshop", Workshop at ICALEPCS'19, Brooklyn, NY, USA, 6 Oct 2019. https://indico.cern.ch/event/823284/