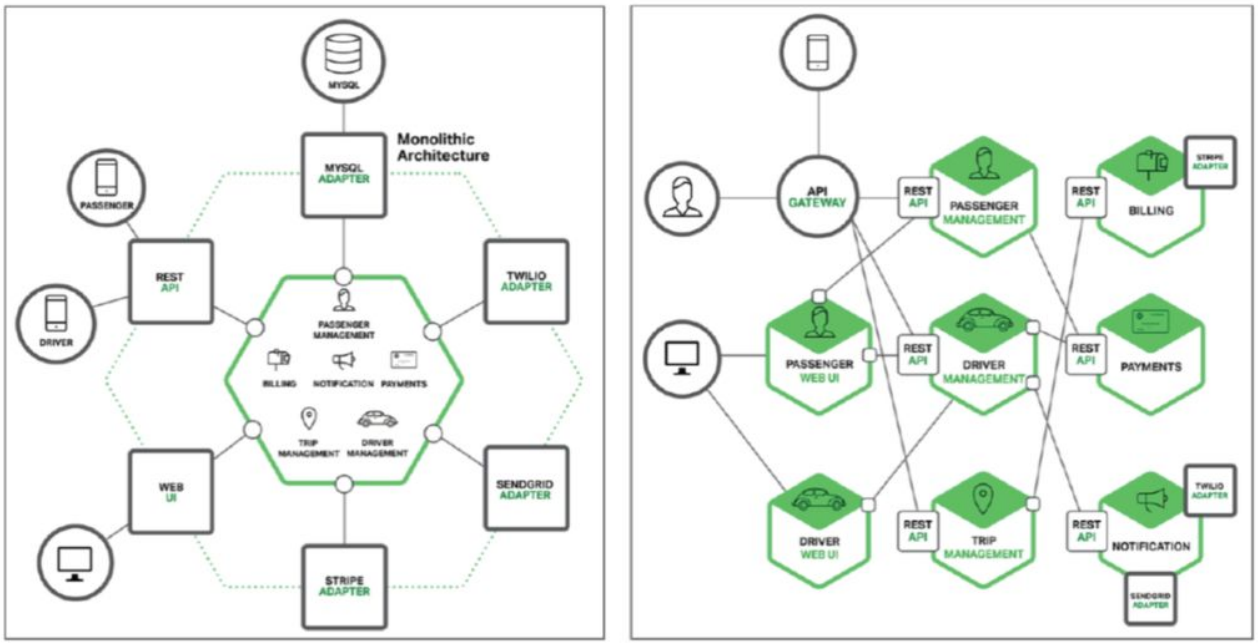
The OpsCruise MONITORING Manifesto: The ROAD to Autonomous Operations



From Monoliths to Microservices' (Seer: Leveraging Big Data to Navigate the Increasing Complexity of Cloud Debugging.' Hot Clouds 2018)

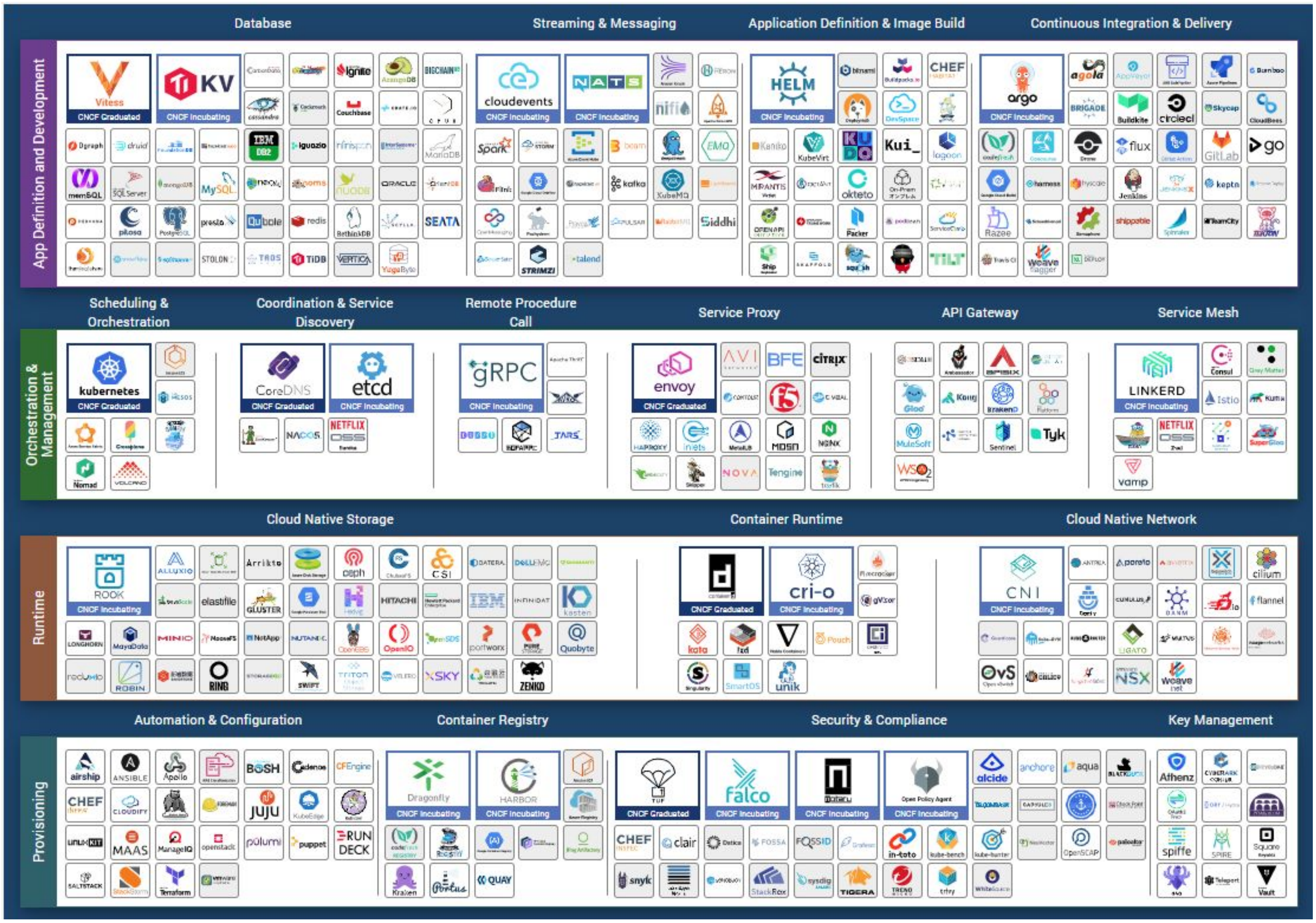
Cloud Applications are Different . . . But more Common than Different

The movement to the cloud has fundamentally changed how applications are now being written and deployed. This was driven by the need to exploit the benefits of the cloud, and the increased agility in getting new business services up and running quickly. Applications are required to be scalable, faster, automatically deployed, and support continuous delivery. This has now been achieved by the use of microservices architecture where the application comprises independent smaller software services, designed to serve specific tasks. As these services, many packaged in containers, and some as SaaS, communicate with each other to support fully functional software. Cloud native applications built from microservices will, by some estimates, show up in 90% of all applications (apps) by 2022[1].

[1] IDC FutureScape https://www.idc.com/getdoc.jsp?containerId=prUS44417618

Three resulting trends that have profound impact in managing applications are now evident.

1. Applications are much more diverse in their makeup. The proportion of open source application components have been growing steadily, close to 40% or more on average,[2] versus custom proprietary software development. This means more and more applications deployed by different businesses are using a common set of well-known open source applications.



From CNCF Cloud Native Interactive Landscape: https://landscape.cncf.io/

2. Even when managed services are provided by cloud vendors, they fall into common categories. This opens up the concept of a common model that works across legacy and cloud services for common component classes. For example, Aurora, RDS from Amazon are similar to relational DB services from Google, Microsoft Azure and Oracle, as are Amazon's DynamoDB and Google's Spanner for scalable databases.

[2] Why open source software adoption is accelerating in the enterprise. https://www.techrepublic.com/article/why-open-source-software-adoption-is-accelerating-in-the-enterprise/

The snippet above from the Cloud Native Computing Foundation (CNCF landscape infographic illustrates how these common classes of cloud application landscape have evolved.

3. The next major trend that came with the move to cloud is how we now manage and orchestrate applications. As the first CNCF project, Kubernetes has become the fastest growing project in the history of open source software. Its popularity was driven by: portability across any cloud environments, public or private; ability to run on VMs or bare metal; high availability at both the application and the infrastructure level including a reliable storage layer; a very large community of contributors; and a vast ecosystem of complementary open source tools designed specifically to work with it.

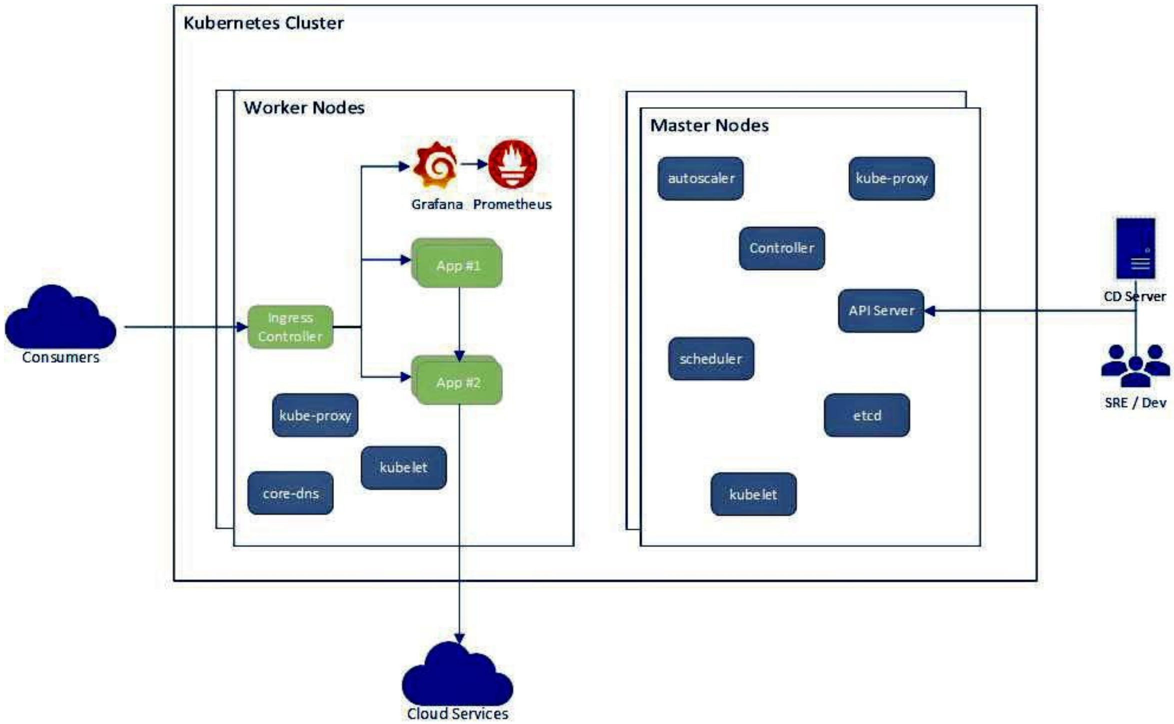
Cloud-Native Changes How We Monitor and Manage Applications

Microservices applications are characterized by not only large scale in terms of number of heterogeneous components, containers or pods, nodes or VMs, but also highly dynamic structure, transient workloads, and changes within the application components. Another important characteristic of microservice architecture is the much higher communication to computation ratio compared to monoliths[3]. This larger communication between services has a bigger impact on performance including bottlenecks that shift under different load conditions across the microservice chains.

Comprehending and managing performance of microservice applications is much more challenging. The old ways of monitoring are no longer enough!

To make matters worse, legacy monitoring systems pre-dating Kubernetes have been built on disparate, closed, and proprietary silos of data collectors and its associated analyses. Ops teams had to build up and train on multiple dedicated and specialized tools for logs, metrics, and traces. With microservices applications orchestrated by Kubernetes, Ops teams have to time multiplex between these different tools to understand the health of their applications, with more manual effort required to maintain them.

[3] Gan et al, "The Architectural Implications of Cloud Microservices"' Hot Cloud 2018



Prometheus monitoring of Kubernetes applications

Specifically, because real time application metrics of applications can now be provided by Prometheus, the second open source CNCF project that is quickly becoming the de facto open monitoring and trending system. With built-in and active scraping, storing, querying, graphing, and alerting based on time series data, a rich data model and query language, and scalability options, there are few reasons not to use it in Kubernetes application. In fact, in Gartner's latest report of April, 2020, they estimate by 2025, 50% of new cloud-native application monitoring will use open-source instrumentation instead of vendor-specific agents for improved interoperability, up from 5% in 2019.[4]

The Data You Need is Already Available from Open Source Telemetry!

The Kubernetes and cloud ecosystem from CNCF has many free monitoring, such as Prometheus for real-time metrics, and Fluentd and Loki for logs, telemetry tools that can provide the information needed for actionable observability.

OpsCruise has shown that using data from such open source tools that provide metrics, events, flows and logs, one can create detailed insights into the application operations. And, then use it further to proactively identify emerging problems, isolate sources of problems and determine corrective actions.

[4] Gartner, Inc - Magic Quadrant for Application Performance Monitoring - Published 22 April 2020 - ID G00466308

Standardization has many more benefits: moving to open standards that have growing communities behind them also is future-proofing. Besides reduced cost in paying for proprietary monitoring, it has other lower costs including lower costs for training staff with fewer common consistent tooling.

With Prometheus to monitor Kubernetes application real-time metrics, as well as other open source tools such as Fluentd and Loki for logs, Jaeger for traces, we now have a fully open standardized application monitoring system.

Actionable Observability - We Can't Wait for the Crash to Happen!

There is an imperative need to have 'early warning systems' to reduce the war room effort that Ops faces. Given the increased complexity of cloud application in terms of the volume and rate of change a reactive response to application problems is becoming less acceptable. Post-facto causal analysis and fault isolation that occurs in war rooms are faced with an increasing number of objects, containers, services, events and logs that are becoming unmanageable and untenable.

Legacy monitoring tools are great for viewing dashboards and setting thresholds for alerts. Unfortunately, while Ops teams can get a shower of alerts, sometimes many which are false, they often do not identify or help isolate the cause of the problem, requiring significant human effort in remediation. As a result a lot of time is spent with skilled staff in war rooms.

What Ops needs is to detect problems and take corrective actions to minimize the war rooms. In short Ops teams need closed-loop control of their applications!

Building Real-Time Intelligence on the Open Source Stack

OpsCruise's approach is to build a model-based control system for cloud applications. Given the complex, heterogeneous, and dynamic distributed systems, a multi-stage sequence of processing is required. Each step or stage poses a different problem statement requiring a different solution as shown below.

The objective of each stage in this seven-step pipeline is to gain progressively increasing visibility into and application understanding across its components so that emerging problems can be detected, analyzed, isolate faults and determine the required corrective action.

Application understanding is built from gathering and processing information collected from open sources of monitoring as well as configuration information from Kubernetes and cloud platforms.

- Structure and topology of the application
- Dependencies across application, and up and down the platform stack
- Predictive dynamic behavior model

Collecting data from its open data collectors running non-intrusively in the control plane (Stage 1),

OpsCruise integrates the information to auto-construct the application structure, and enrich it with real-time metrics, including all operational flows between components (Stage 2).

Once the structure has been identified, predictive behavior models for all components are automatically built from the real-time metrics at runtime. The models are built at scale across all application components using novel machine learning techniques that capture the normal operating regions of the component under different load conditions and with minimal supervision (Stage 3).

The model exploits curated knowledge, such as attributes of known open source applications and common classes of behavior models, i.e., for Pods/Containers, Databases, Load Balancers, Queues, Application VMs, etc. With the behavior model used in predictive mode, OpsCruise detects unexpected and undesired changes that indicate emerging performance anomalies. Details from the anomaly analysis together with application dependencies are used to find the root cause and isolate faults (Stage 4).

To determine the remediation corrective action the anomaly is classified (Stage 5) at a granular level using contextual knowledge of the environment and information from the earlier stages.

The control action or recommended fix (Stage 6) associated with the classified anomaly is forwarded to operators (e.g., Kube operators) or scripted or automation frameworks (Stage 7). Feedback on the efficacy of the corrective actions are used to improve the root cause - fault isolation -remediation cycle.

In the above process OpsCruise ingests and analyzes a diverse set of telemetry data from across the stack in real time to make informed decisions and recommendations for managing the application. Leading the analysis with real-time metrics and using logs to both confirm the specificity of the fault isolation and the recommended fix. Further, the pipelined process is designed to be run in a proactive manner leading Ops to a path of autonomous operations similar to using cruise control in automobiles.

The Result: Happier SRE/DevOps teams and Savings on Cloud Resources!

In production, OpsCruise has proven it can make significant improvements around the observability of cloud applications with the following business outcomes:

50% improvement in SRE productivity. OpsCruise can get organizations on the path to supporting more applications at higher release velocity with existing staff. Organizations have been able to improve their SRE / Dev ratios by as much 50%.

40% more alerts handled by L1. Because OpsCruise alerts are highly enriched and prescriptive, it means fewer need to be escalated to expensive L2/L3 resources for resolution.

20% fewer cloud resources. A lack of performance understanding leads to over-sized instances and pod allocations in K8s. OpsCruise has enabled organizations to make adjustments using as much as 20% fewer resources without impacting performance or risk.

Higher availability and more performance digital services. The relationship between application performance and customer satisfaction, and ultimately, revenue has been proven in many industries. OpsCruise is helping customers identify performance degradations before users notice and solving them more rapidly. That has an immeasurable impact on the business.