Monitoring Hybrid Cloud Workloads using Open source technologies.

Presented by: Joe Calcada.
For Spring CMG.
April 24 2018.
Introduction to Microservices, Docker, & Kubernetes
Monitoring Hybrid Cloud with Open Source Tools
Demo
An engineering approach focused on decomposing an application into single-function modules with well defined interfaces which are independently deployed and operated by a small team who owns the entire lifecycle of the service.

Microservices accelerate delivery by minimizing communication and coordination between people while reducing the scope and risk of change.
Microservice Architecture

Developer Benefits
- Small team development
- Repeatable
- Consistent
- Faster response and deployments
- Independent lifecycle
Microservice Architecture

StockTrader Application Architecture

[Diagram showing the architecture of a StockTrader application, including Microservice Builder, BROWSER, Trader, Portfolio, Db2, Redis, Stock Quote, Loyalty Level, Notification, MQ, Open Whisk, API Connect, Quandl, and Slack.]
What’s A Container?

Not a middleware container – no interceptors, no transactional or security support …

Nothing to do with EJB or Web containers

A standard way to **package** an application and all its dependencies so that it can be moved between environments and run without any changes.

Containers work by isolating the differences between applications **inside** the container so that everything **outside** the container can be standardized.

Implemented typically as a light weight mechanism to build and deploy user code and data along with the required components with bare minimal Linux kernel
Containers Accelerate Software Delivery

Any application stack

- Static website
- User DB
- Web frontend
- Queue
- Analytics DB

Development VM

QA server

Customer Data Center

Public Cloud

Production Cluster

Contributor’s laptop

Any hardware environment
## Common Container Use Cases

<table>
<thead>
<tr>
<th>#</th>
<th>Scenario</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DevOps</td>
<td>Want to drive cultural and technology change aligned to a DevOps methodology</td>
</tr>
<tr>
<td>2</td>
<td>Continuous Integration / Continuous Deployment</td>
<td>Want to increase rates of innovation by shipping more software faster and improving developer &amp; operations productivity;</td>
</tr>
<tr>
<td>3</td>
<td>Software Upgrades / Platform Refresh</td>
<td>Want to revisit architecture as part of v2v upgrades or platform refresh; re-architect for portability</td>
</tr>
<tr>
<td>4</td>
<td>App Modernization / Microservices</td>
<td>Want to transition from monolithic application code bases to microservices and distributed processing architectures; new features delivered as microservices; exploit serverless computing</td>
</tr>
<tr>
<td>5</td>
<td>Path to the Cloud</td>
<td>Want to migrate workloads to cloud, re-architect for portability</td>
</tr>
<tr>
<td>6</td>
<td>Big Data Processing</td>
<td>Want to provide self-service big data access and analytic processing environments to different lines of business</td>
</tr>
<tr>
<td>7</td>
<td>Infrastructure Optimization</td>
<td>Want to maximize infrastructure efficiency and reduce operating costs by increasing server density, reducing licensing costs and maintaining application portability</td>
</tr>
</tbody>
</table>
Docker – Mission

Docker is an open platform for building distributed applications for developers and system administrators.

Build  Ship  Run

Ubuntu  MySQL  MongoDB  NGINX  Any App

Node.js  CentOS  Redis  Apache  Anywhere
Docker Containers Provide Extreme **Portability**

A platform to build, ship, and run applications in “containers”.

Developers & SysAdmins love the **flexibility and standardization** of Docker

**Lightweight**

Containers are “light” users of system resources, **smaller** than VMs, start up **much faster**, and have **better performance**

**Ecosystem-friendly**

A **new industry standard**, with a vibrant ecosystem of partners.

1300+ community contributors; 100,000 third-party Docker projects on GitHub

**User-friendly**

Developers build with ease and ship **higher-quality applications**

SysAdmins deploy workloads based on **business priorities** and policies.

"**Containers managed by Docker are effective in resource isolation. They are almost on par with the Linux OS and hypervisors in secure operations management and configuration governance.**"

Joerg Fritsch, Gartner Analyst, Security Properties of Containers Managed by Docker, January 7, 2015
Docker uses a copy-on-write (union) file system.
New files (and edits) are only visible to current/above layers.

Layers allow for reuse:
- More containers per host
- Faster start-up/download time – base layers are "cached"

Images:
- Tarball of layers (each layer is a tarball)
# Speed of Deployment

<table>
<thead>
<tr>
<th></th>
<th>Obtain within...</th>
<th>Manual deployment takes...</th>
<th>Automated deployment takes...</th>
<th>Starts in...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bare Metal</td>
<td>Days</td>
<td>Hours</td>
<td>Minutes</td>
<td>Minutes</td>
</tr>
<tr>
<td>VM</td>
<td>Minutes</td>
<td>Minutes</td>
<td>Seconds</td>
<td>&lt; Minute</td>
</tr>
<tr>
<td>Container</td>
<td>Seconds</td>
<td>Minutes</td>
<td>Seconds</td>
<td>Seconds</td>
</tr>
</tbody>
</table>
Near Bare-Metal Performance

![Bar chart showing Linpack GFLOPS for Native, Docker, and KVM]

Container Orchestration

- Ease the management and monitoring of large numbers of disparate containers, distributed across multiple hosts

- Container scheduling
  - Placement of containers on the most appropriate host based on constraints
  - Rescheduling if a container or a host fails
  - Ability to rollout/rollback updates whilst maintaining availability

- Deployment of multiple containers that form an application from version controlled configuration

- Policies for placement, security, performance, HA

- Routing of inbound and inter-container requests (service discovery and routing)

- Enterprise integration with existing tools e.g. CI/CD and SSO/RBAC
Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

Kubernetes has a number of features. It can be thought of as:

- a container platform
- a microservices platform
- a portable cloud platform and a lot more.

Kubernetes provides a container-centric management environment. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. This provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), and enables portability across infrastructure providers.

Kubernetes is comprised of a set of independent, composable control processes that continuously drive the current state towards the provided desired state.

Source: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/
Kubernetes Architecture

- UI
- API
- CLI

Kubernetes Master

- etcd
- API Server
- Controller Manager Server
- Scheduler Server

Worker Node 1
Worker Node 2
Worker Node 3
Worker Node n

Image Registry
Monitoring hybrid cloud Environments

• *Traditional Environment* means the on-premise environment of physical or virtualized servers that has been common for years with both automated and non-automated provisioning.

• Application platform for developing and managing on-premises, containerized applications (PaaS/CaaS) (IE. IBM Cloud Private).

• One-stop cloud computing solution which provides multiple types of solutions (IaaS/SaaS/PaaS/CaaS/FaaS). (IE. IBM Cloud)

• *3rd Party Cloud* may be either on-premises or cloud-based, depending on the provider.
Monitoring the Environments

Watch the workload’s infrastructure/platform: Infrastructure is divided into on-premise where you are responsible for the platform down to the hardware, and off-premise/cloud where your service provider supports the infrastructure and your solo concern is that the platform is available.

Monitor Cloud-Ready workloads: These are workloads that are suitable for running in the cloud, but their heritage is from the traditional environment. Applications running on Virtual Machines that can be lifted and shifted to the cloud are the classic example of Cloud Ready applications.

Monitor Cloud-Native workloads: These are workloads that were designed to run in the cloud. Container, runtime & serverless applications are the typical kind of workload that is Cloud Native.

Collect logs: Since the multiple workloads create a wide variety and large amount of logs, it is critical to have a mechanism to collect and make sense of all the log entries. The collection and aggregation of logs is where problem analysis begins.
# Monitoring the Environments

<table>
<thead>
<tr>
<th>Service Management</th>
<th>Event Management</th>
<th>Notification</th>
<th>Collaboration</th>
<th>Runbooks</th>
<th>Ticketing</th>
<th>Dashboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Management</td>
<td>Log File Agent</td>
<td>Log Analysis</td>
<td>ElasticSearch</td>
<td>Log Analysis</td>
<td>IBM Cloud Log Service</td>
<td>Log Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloud Native Workloads</td>
<td>N/A</td>
<td>Prometheus</td>
<td>APM</td>
<td>IBM Cloud Monitoring Service</td>
<td>APM</td>
<td>NOI /CEM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOI/CCEM</td>
<td>APM</td>
<td></td>
<td>NOI /CEM</td>
<td>APM</td>
</tr>
<tr>
<td>Cloud Ready Workloads</td>
<td>NOI /CEM</td>
<td>APM</td>
<td>N/A</td>
<td></td>
<td>NOI /CEM</td>
<td>APM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>NOI /CEM</td>
<td>APM</td>
<td>NOI /CEM</td>
<td>Console</td>
<td>NOI /CEM</td>
<td>APM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOI /CEM</td>
<td>NOI /CEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment Type</td>
<td>Traditional Environment</td>
<td>IBM Cloud Private</td>
<td>IBM Cloud</td>
<td>3rd Party Cloud</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19
Levels of CSMO in Private Clouds (ICP for example)

- **Level 0 – Out of the box:**
  - Built-in Dashboard for metrics
  - Built-in Log console
- **Level 1 – Default monitoring workloads**
  - Deploy Prometheus for metrics
  - Deploy Elasticsearch (ELK) for Logging
- **Level 2 – External Monitoring and Logging**
  - External APM monitors metrics and thresholds
    - Synthetic Transactions
    - Elasticsearch stacks forwards to external Elasticsearch stack
      - Consumed externally
- **Level 3 – Integration with CSMO toolchain**
  - Forward events to a “Manager of Managers” (ie. Omnibus)
  - Forward Log events to “Manager of Managers” (ie. Omnibus)
  - can open incidents and integrate with control desk tools and runbook tools
Within IBM Cloud Private there are several ways to view high level Key Performance Indicators

1. The ICP dashboard shows a high level overview of resource usage and status.
2. The Kubernetes Dashboard shows CPU and memory consumption of the full cluster and deployments.
3. Within an application you can view performance of the pod.

All well and good but you can only look and “not touch”. There are APIs available.

We need to be able monitor and alert on Out of Bounds conditions for the applications and their infrastructure.
What is the base level of Monitoring/Logging? (Level 1)

Monitoring and logging in ICP

1. The ICP dashboard shows the status, metrics and log entries for all the workloads & Infrastructure.
2. An Elasticsearch stack collects logs from all the workloads & Infrastructure. Multiple Elasticsearch stacks are possible (but each log can only reside in one stack)
3. A Prometheus stack collects logs from the Kubernetes workload
4. Customizable Grafana and Kibana dashboards are available for use
5. Prometheus can send alerts to external system (email, webhooks)
Prometheus is a Cloud Native Computing Foundation (CNCF) project, is a systems and service monitoring system. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.  
https://prometheus.io/

Prometheus has several components for Time Series Data Collection, an Alert Manager and the Prometheus Server which scrapes and stores the data.

Prometheus is installed via a Helm Chart into ICP.
IBM APM v8 (level 1+2)

IBM APM is a comprehensive solution that helps you manage the performance and availability of applications that are deployed on premises (private), in a public cloud, or as a hybrid combination. This solution provides you with visibility, control, and automation of your applications, ensuring optimal performance and efficient use of resources.

You can reduce and prevent outages and slowdowns around the clock in a hybrid application world as Cloud APM assists you in moving from identifying performance issues to isolating where the problem is occurring and diagnosing issues before your business is impacted.
Grafana is an open source metric analytics & visualization suite. It is most commonly used for visualizing time series data for infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control.  
https://grafana.com/grafana

While Prometheus has limited dashboarding capabilities of its own, regular display of metrics will be done using Grafana dashboards.

Prometheus and the Grafana Dashboards are installed together as applications into ICP using Helm charts.
The Elasticsearch stack (previously known as the ELK Stack) is a Log Management system which is composed of 3 layers:

- Kibana is the extensible user interface for configuring and managing all aspects of the Elastic Stack.
- Elasticsearch is a distributed, JSON-based search and analytics engine designed for horizontal scalability, maximum reliability, and easy management.
- Logstash is a dynamic data collection pipeline with an extensible plugin ecosystem and strong Elasticsearch synergy.
- Beats is a platform for lightweight shippers that send data from edge machines to Logstash and Elasticsearch.

Elasticseach is packaged as a series of Helm charts and are deployed automatically when ICP is installed (can be customized and/or opted-out of)
What are the advanced Logging capabilities for ICP? (Level 3)

You can forward logs to an external repository so that you can consume logs in your own existing log management system.

Logs cannot be forwarded to multiple repositories.
Manager of Managers probes and observers collect events from multiple data sources including Kubernetes, Prometheus & DevOps monitoring.

APM monitors Private Cloud (both workloads and Infrastructure) and forwards events to Manager of Managers.

MoM Event Management layer performs de-duplication, correlation, grouping, etc...
Build to manage
  • Generate “health” events from within your application
  • Testable APIs

Synthetic Transactions (I.e. Selenium tooling)
  • User experience
  • Coverage for components outside of your application
Custom Dashboards with Grafana

Pre-built dashboards are available here https://grafana.com/dashboards and can be imported to your Grafana application – the pre-built dashboards may require some “adjusting” for your environment.
THANK YOU