eBook

Container Monitoring at Scale

A Guide to Metrics-Driven Analytics for Containerized Applications

Stela Udovicic
VMware

WAVEFRONT by vmware

Contents

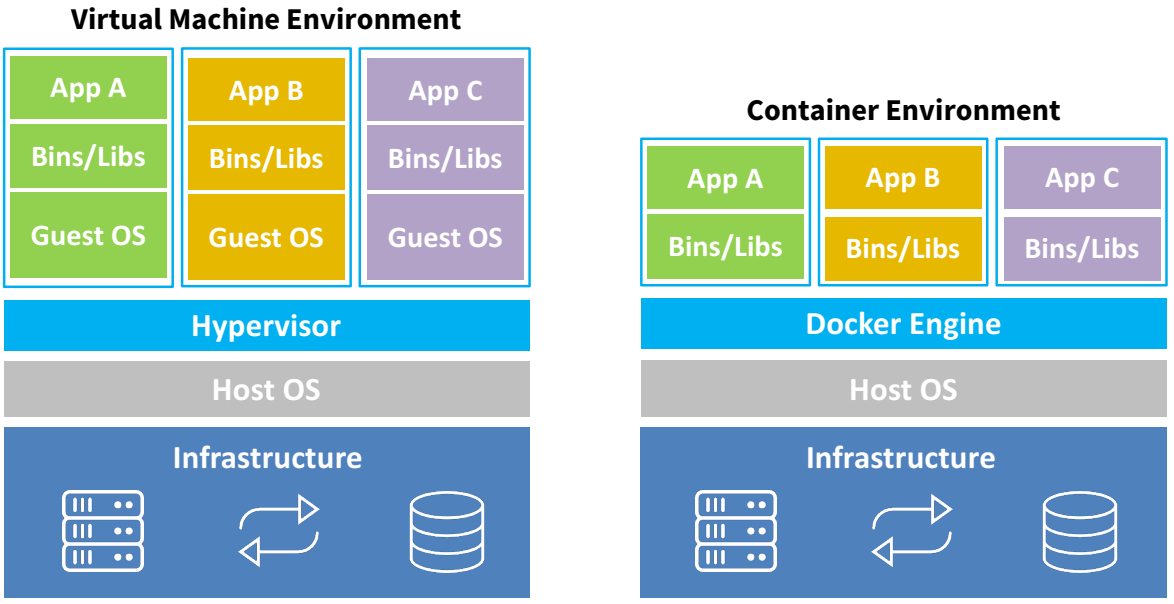
Introduction

Container adoption among developers and DevOps teams is unlike anything we have experienced before. And it's because of containers' unique characteristics such as their small footprint, minimum overhead, agility and portability. Containers help to move faster with application innovation. Developers are empowered to experiment, knowing what worked on their laptop, will work in prod or test. But it's the very same container attributes that are fueling innovation such as fast spin up/down, greater dynamism, and further abstraction, that are also making them more challenging to monitor. Containers create additional monitoring complexity due to their short lifespan, highly dynamic nature and lean functionality. Thus, container monitoring tools must reflect some of the same container attributes: they must also be lightweight, agile and scalable.

In this e-book, we'll provide you insights into the challenges and solutions for container monitoring. We will also to help you understand the unique value of a new approach to monitoring well suited to containerized applications, based on metrics-driven analytics, an approach pioneered by Wavefront by VMware. Finally, we'll also provide real-world case studies of how two Wavefront customers, FashionID and Mixpanel, use Wavefront's metric analytics platform to efficiently monitor their containerized cloud applications and orchestration systems at production scale.

What are Containers?

Containers deliver an abstraction mechanism that enable developers and DevOps teams to design, deploy, and run applications while keeping them isolated. Using containers, engineers can wrap an application with all needed components such as libraries and dependencies and deliver it as a package, in a container. That container can run on any environment, in QA or in production, in the the same way it ran on the developer's machine, making containers extremely portable. It's this portability combined with isolation and small footprint that's propelling container adoption, fueling innovation at unprecedented speed. Containers resemble virtual machines, as both provide abstraction and isolation. However with containers, the application is not packaged with the entire virtual OS, but only the essential elements for an application to run. The diagram below illustrates the main differences of containers from virtual machines.



There are several types of containers mechanisms, such as CoreOS's rkt or Cloud Foundry's garden, but Docker containers are the most popular with a massive ecosystem. Since its introduction in 2013, Docker adoption has grown to hundreds of millions of Docker containers in use today, with continued rapid growth.

Container Orchestration

When the number of containers in production starts to grow, so does the need for container management and orchestration. Container orchestrators provide management tasks such container automation, scaling, proper resource allocation, management, health checks, networking (load balancing, discovery, etc.), and more. Several popular container orchestration and management systems include:

- **Kubernetes** – open source and developed by Google, but now part of Cloud Native Computing Foundation backed by many companies including VMware, IBM and others
- **Amazon ECS –** Amazon's container management service for Docker containers but only supporting EC2s
- **Docker Swarm** – Docker's own orchestrator which is now part of Docker Engine, integrated with Docker APIs

Container orchestrators are gaining in popularity because they remove management worries for developers. However they introduce another layer of complexity. Whichever orchestrator chosen, you now have another abstraction layer to monitor. Later in this eBook, you'll learn how to use metrics to gain real-time insights into the performance of applications, containers and orchestration systems.

What is Container Monitoring?

The requirements that containers put on monitoring tools drive the need for a different monitoring approach from traditional methods. For example, traditional approaches poll hosts to assess resource load; or collect logs to troubleshoot infrastructure issues; or deploy probes for network monitoring. Since containers are so lightweight, they have a short lifespan, and can be spun up and down easily. But from this very dynamic nature, the traditional monitoring methods don't scale. They end up being very slow, coarse, and expensive – introducing significant overhead into the container image, and therefore impacting what's being monitored.

Fortunately, Docker provides a very efficient way to export metrics directly from containers that doesn't impact containerized service performance. The Docker Stats API can stream key metrics from container resources. These metrics can be gathered using various open source collectors. One of the easiest to deploy and run is cAdvisor, a project originated by Google. The cAdvisor tool is actually a container itself and it collects, aggregates, and exports performance and usage metrics about all running containers. It tracks and retains telemetry for each running container including historical resource utilization, distributions (histogram metrics) of complete historical resource utilization, network metrics, resource isolation parameters and many more.

The best way to analyze them is to use a metrics-driven monitoring and analytics platform that's designed for high-scale, and with the flexibility to match the dynamics of monitoring containerized applications.

This means visibility into not only container resources but also into the performance of applications running in containers. Such a platform is Wavefront by VMware. If your use of containers is expected to be relatively small and limited, then you may consider an in-house, open-source platform (e.g. Graphite, OpenTSDB, Prometheus), albeit it will pose challenges with scalability, maintainability and expanding requirements down the road.

Here are some issues to keep in mind before choosing an open-source platform for container monitoring. Over the long term, the total cost of ownership (TCO) is actually higher, particularly as your metrics volumes rise and developers' requirements expand. As the exponential rise of containers is something to expect relatively quickly, open-source platforms have scaling limitations, maintenance complexities, and a lack of high availability options – often resulting in monitoring platform outages, slow enhancement velocity, and sunk engineering time attributed to the growth.

More than 70% of Digital Enterprises Are Frustrated with Building Monitoring Tools with Open Source

Build Apps, Not Monitoring

Digital Enterprises Need to Scale Container Monitoring Without Effort

Maintaining in-house software is distracting
Address reliability and security, day one
Deploy world-class monitoring faster, easier

Container Monitoring is Challenging

Containerized Applications Require a Unique Analytics Approach

As explained at the beginning, Docker containers help developers deliver code faster by freeing them from dependencies to unique environments. Containers also help DevOps teams be more agile helping them create and scale new services quickly. However, monitoring containerized applications is challenging. The dynamic and ephemeral nature of containers requires quick troubleshooting and action driving the need for real-time visibility, or otherwise, adds significant delays in data processing and visualization.

Explosion of container metrics compared to traditional software architectures

Also, breaking monolith applications into microservices results in a dramatic increase in a number of containers leading to a proliferation of instrumentation points which can overwhelm your monitoring infrastructure and monitoring pipelines (networking, etc.). Multiple additions of 100+ metrics or even 1,000+ should not incur any jump in your processing costs, nor fill up your disk space quickly.

Prior to containers, our customers gathering traditional application metrics including OS metrics could gather ~ 100-150 metrics (OS and application metrics) – more if you add virtual machine or custom code metrics. For the sake of simplicity, say you have 5 containers per host and you collect 50 metrics per container itself (system metrics, resource utilization, etc.). This means that for one host running 5 containers you now have 500 metrics (250 metrics for containers and ~ 250 per running application). Now say you have spun up and down your 5 containers (10x), all of a sudden you have 5,000 metrics. As mentioned, this is a simplified example but it illustrates the need for a flexible platform that's cost optimized for quick storage and analysis of a massive amount of metrics.

mixpanel

"I cannot imagine the world without time series databases and metrics-driven analytics software like Wavefront."

- Arya Asemanfar
Manager Infrastructure Team, Mixpanel

High Velocity Metrics

Another important element stemming from the ephemeral nature of containers is that modern containerized applications exhibit more dynamic and erratic behavior. To ensure their continuous availability and performance, it's becoming essential to analyze high-velocity metrics on distributed systems (microservices) that handle thousands, tens, or hundreds of thousands of requests per second. This is where traditional monitoring tools ften fall short as they can't keep up with the rate of container and microservices changes. If your containers are spun up and spun down within seconds, and you have monitoring tools that provide aggregated metrics in minute+ buckets, than you have missing visibility data and may miss critical incidents.

Unified Visibility Needed Across Many Hierarchy Levels - Service, Host, Cluster, Container

Another critical aspect of container monitoring is the need to monitor the performance of containerized application in addition to the container resources utilization. Metrics gathered from application and overall container services need to provide visibility on any hierarchy level – service, host, cluster, container and other resources which makes it easy to optimize performance across the entire system and also monitor aggregated performance (see histogram section for more info).

Why Metrics-driven Analytics is the Best Approach for Container Monitoring

The Wavefront unified metric analytics platform delivers out-of-the-box comprehensive visibility across all applications, microservices, Docker containers and orchestration systems including Kubernetes, Amazon ECS, Docker Swarm, and Mesos. Visibility is provided across all hierarchical elements from container resources utilization up to services level. It's very simple to get started with monitoring your containerized applications and container orchestration systems.

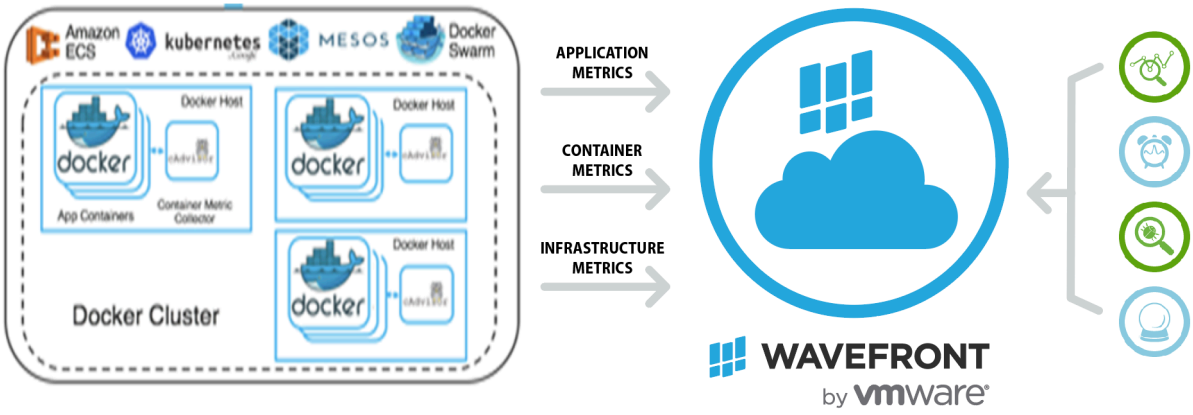
For metrics collection from containers, the Wavefront platform uses a lightweight agent, cAdvisor, that runs natively as a Docker container itself. cAdvisor collects various performance metrics and sends them to the Wavefront cloud service. Users have the flexibility to configure how often metrics are sent to Wavefront service depending on the granularity they'd like to see; the default is 60 seconds. Once the container metrics are collected, developers and DevOps teams can use pre-built dashboards and alerts to get started quickly.

By collecting, processing, storing and visualizing metric data from custom applications, microservices, containers, and cloud infrastructure performance metrics, Wavefront empowers developers and DevOps teams to develop, run and manage containerized services easily. Or, you can customize it all using the Wavefront Query Language to tailor analytics to your unique environment.

You can get the latest Wavefront's cAdvisor image from Docker Hub. For more details on configuring container metrics ingestion, please explore Wavefront's documentation, e.g. for describing how to monitor Kubernetes, or Amazon ECS. There are many other packaged applications and integrations to explore including pre-packaged dashboards and alerts for getting visibility into performance and cost of various AWS cloud services.

With the rise of containers, container monitoring alternatives are sprouting too. Based on extensive experience with our customer base who have evaluations many options, it's worthwhile to mention some of the limitations of other options. If you resort to using logging tools, your monitoring becomes heavy, slow and costly. If using container-specific solutions that look at low level OS processes, there is the question of security and limited narrow visibility into only container-specific elements. Some metrics platforms provide out-of the-box visibility into the container environments, but their dashboarding cannot be customized, making it challenging to add or remove new services or implement other specialized changes.

Wavefront by VMware provides both lightweight, comprehensive and customizable solution for monitoring across all container environments as well as beyond containers and across all applications, cloud services and infrastructures.



Key Benefits of Metrics-driven Analytics for Container Monitoring

Real-time Containerized Application Behavior Visibility

Wavefront provides packaged container dashboards that provide real-time insight into key container resources and application metrics. Developers and DevOps teams can use open source agents (cAdvisor) to instrument container environments and get insight into the aggregate behavior of their applications and infrastructure at scale.

Scale Containers without Scaling Monitoring Cost

Using Wavefront for container-driven metrics analytics, engineering teams can achieve significant cost savings as they can easily add new container metrics, without worrying about incremental costs. Since metrics are extremely efficient to store and process, you can trend and retain at minimum cost all container metrics for analysis and capacity planning. Or just to analyze transient behavior even after specific containers instances aren't running.

Avoid Perfect Storms of Container Alerts

With many microservices and containers that spin up and down, DevOps teams can easily get overwhelmed with tending to thousands of alerts for containers that may not even be running. Wavefront intelligent driven alerting can help teams focus only on important alarms while avoiding alert storms. Using Wavefront Query Language, you can create alerts on any condition and suppress unwanted alerts, avoiding unecessary late-night calls.

Early Indicator of Incidents

Developers and DevOps teams can use unified Wavefront metrics and alerts to proactively isolate leading anomaly indicators. Since metrics analytics is fast, and as such an excellent match for containers' short lifespan, it helps you detect and alert on unpredictable containerized application behaviors and changes across your entire application environment.

Self-service Customized Container Metrics for Developers

Developers can take advantage of the Wavefront Query Language to create customized metrics and be fully independent while focusing on application development. For instance, you can tag container metrics to help correlate container performance with host performance and higher service domains, saving time in problem isolation.

Collaborate and Share Container Insight

Containers are meant to be portable and to fuel innovation. Using Wavefront's integration with DevOps tools such as PagerDuty, HipChat or Slack, developers and DevOps teams can collaborate and share container dashboards and metrics across all teams. Metrics-driven analytics becomes a critical feedback loop of the DevOps toolchain. Once an issue is detected in production, based on metrics insight, you can roll-back code update and identify issues within seconds.

Key Metrics for Container Monitoring

What You Should Be Watching and Why?

There is a wealth of performance and utilization metrics that can be collected from containers (Docker Stats API) and container orchestration systems APIs (Kubernetes, Amazon ECS, etc.). The beauty of the Wavefront platform for developers and DevOps teams is that it offers you the flexibility to start with a set of packaged container metrics, which give you insight into the health of your overall containerized application environment. It also enables you to go beyond packaged dashboards, and define your own custom metrics and correlate the containerized application performance metrics with the business metrics for unified visibility. For container environments, some commonly used performance metrics include:

Docker Container Metrics

SUMMARY USAGE
- Containers Running
- Host CPU Usage
- Host Storage Usage
- Host Memory Used

CPU
- CPU Usage by Container
- CPU Usage from Containers
- Total Container and System CPU usage

MEMORY
- Memory Usage Bytes by Container
- Container Memory Usage %
- Memory Usage Bytes from Containers

NETWORKING
- Bytes Received by Container
- Bytes Transferred by Container
- Networking Errors by Container

STORAGE
- Disk Usage Summary
- Disk Usage by Devine

Kubernetes Metrics

SUMMARY USAGE
- Nodes
- Namespaces
- Pods
- Pod Containers
- Average Node Memory Utilization
- Cluster Memory Usage
- Average Node CPU Utilization
- Average Node Storage Utilization

NAMESPACES
- Pod Count by Namespace
- Pod Container Count by Namespace
- Memory Usage by Namespace

NODE
- CPU Utilization by Node
- CPU Usage Rate by Node
- Memory Utilization by Node
- Memory Usage by Node
- Bytes Received Rate by Node
- Networking Errors by Node
- Bytes Transferred Rate by Node
- Uptime by Node
- File System Usage by Node
- File System – Available Storage by Node
- File System – Usage %

POD
- CPU Usage Rate by Pod
- Memory Usage by Pod
- Memory Page Faults by Pod
- Bytes Received Rate by Pod
- Bytes Transferred Rate by Pod
- Networking Error by Pod
- File System Usage by Pod
- CPU Usage Rate by Pod Container
- Memory Usage by Pod Container
- Memory Page Faults by Pod Container
- File System Usage by Pod Container
- Uptime by Pod Container
- File System Usage by Pod

Amazon ECS Metrics

SUMMARY USAGE
- Services
- EC2 Instances
- Task Definition Families
- Containers

CPU
- CPU Usage by ECS Service (CloudWatch)
- Avg CPU Usage by EC2 Instance (CloudWatch)
- Average CPU Usage by Task Definition Family
- CPU Usage by Container

MEMORY
- Memory Used by ECS Service (CloudWatch)
- Average Memory Used by Task Definition Family
- Memory Usage by Container

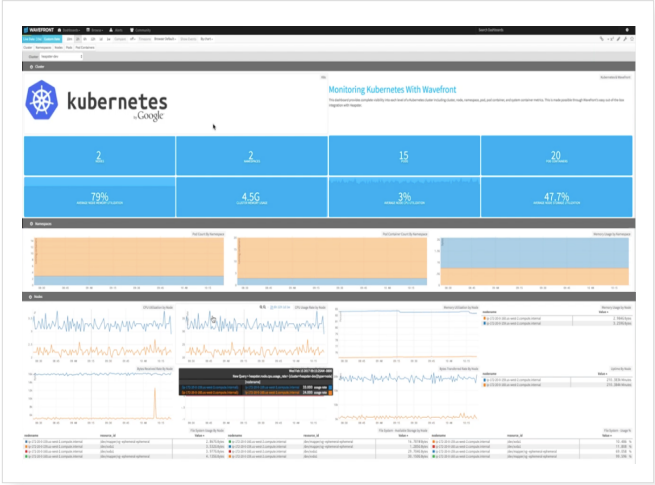
NETWORKING
- Bytes Received by EC2 Instance
- Bytes Out by EC2 Instance
- Bytes Received by Task Definition Family
- Bytes Transferred by Task Definition Family
- Networking Errors by Task Definition Family
- Bytes Received by Container
- Bytes Transferred b y Container
- Networking Errors by Container

STORAGE
- ECS - EBS Cluster Volume Size
- ECS - EBS Cluster Write Bytes
- ECS - EBS Cluster Read Bytes
- Disk Bytes Read by EC2 Instance
- Disk Bytes Written by EC2 Instance
- Disk Usage Summary

Monitoring Kubernetes with Wavefront

Real-time Visibility into Kubernetes Orchestration, Containers and Microservices

Kubernetes is becoming the most popular container orchestration system. It was originally created by Google and it's now part of Cloud Native Computing foundation. For developer and DevOps teams, Kubernetes helps manage, automate and scale their containers so they don't need to worry about mundane management tasks such as scheduling, resource allocation, and more. As such Kubernetes introduces additional abstractions which shield engineers from understanding how app components and containers are scheduled and moved. As a result, you may lose important visibility which can delay troubleshooting.

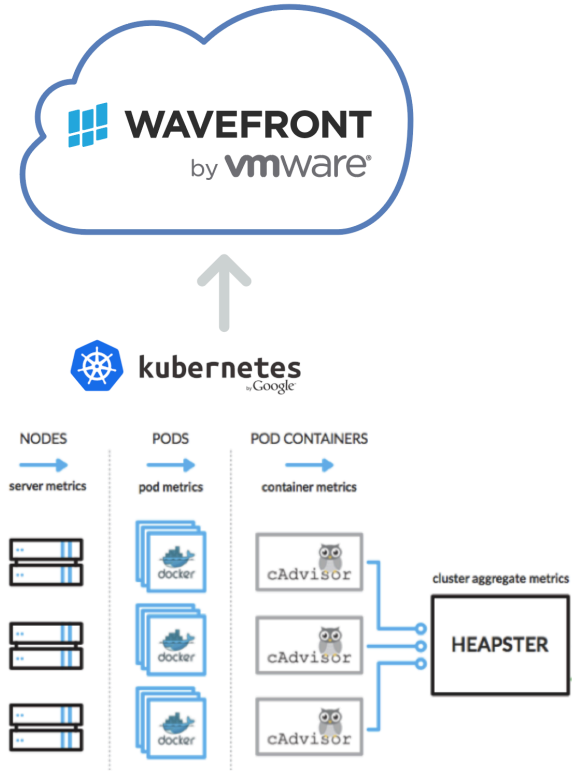


The Wavefront platform collects, analyzes and visualizes performance metrics from the entire Kubernetes orchestration environment including nodes, pods cluster, down to individual container performance metrics and important events such as behavioral errors, at any scale.

Wavefront's tagging and ability to filter metrics by tags is very important as it enables grouping and tracking important entities (containers or others) as they are moved around and scheduled by Kubernetes.

Wavefront provides packaged Kubernetes and container dashboards that deliver visibility into high-level services as well as granular container metrics and its resource consumption. The prepackaged **Wavefront Kubernetes** dashboards include:

- **Node metrics:** average memory, file system, CPU, and storage utilization
- **Namespace** metrics: Memory usage by namespace, various counts (containers, pods)
- **Pod metric:** CPU usage rate, bytes transferred and received rate
- **Container metrics**: uptime by container, CPU, memory and file system usage, as well as memory faults



Later in this eBook, see how Wavefront customers go beyond packaged dashboards and are able to create custom correlated dashboards for containers, tailored to their environments where they correlate service level visibility with individual container performance.

Monitoring AWS ECS with Wavefront

Real-time Visibility into an Amazon ECS Environment

The Wavefront platform collects and correlates metrics from Amazon ECS, CloudWatch, CloudTrail, EBS and many other Amazon services. As a result, it provides comprehensive out-of-the-box visibility into all levels of Amazon ECS environment including all EC2 instances, containers, and task definitions families. Since Wavefront provides insight into both cloud infrastructure and containerized application performance, DevOps and TechOps teams can make sure there is sufficient resources for containers while developers don't need to worry about the rate of container changes (spin-ups and spin downs) and can focus on the microservices applications they are working on.

Using metrics-driven analytics, DevOps and developer teams can detect any performance issues, remediate container bottleneck resource utilization, and alert on any container conditions for early anomaly detection. The Wavefront platform visualizes:

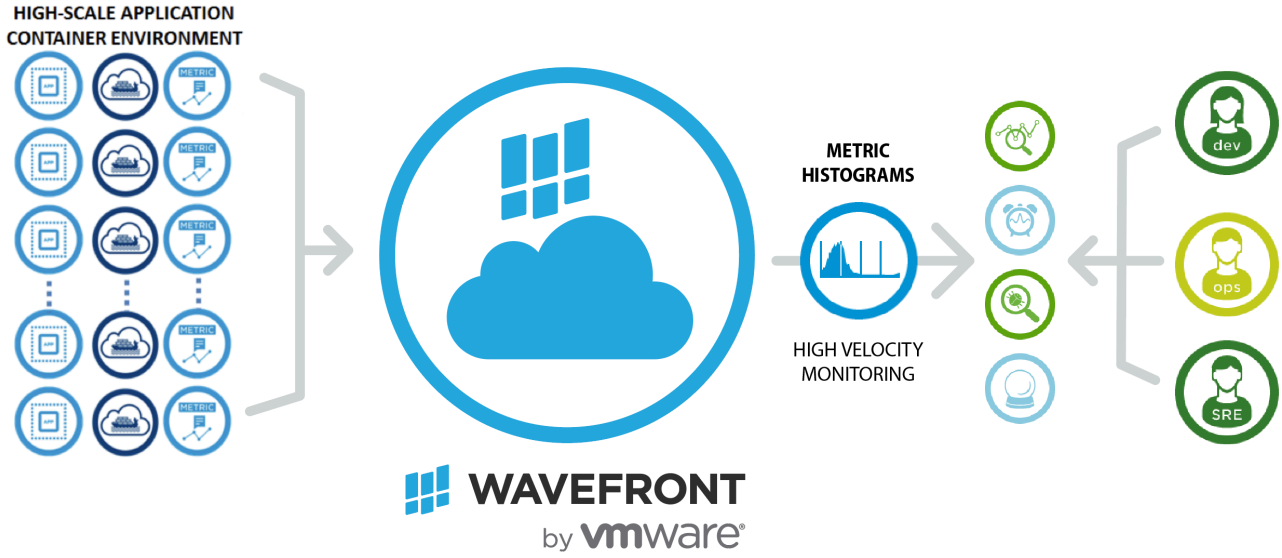
- **EC2 instances metrics**: CPU, memory, networking throughput (bytes in, bytes out), disk reads and writes
- **Task definitions metrics**: CPU usage, memory usage, bytes transmitted/received, networking errors
- **Container metrics**: CPU usage, memory usage, bytes transmitted/received, networking errors
- **ECS-EBS storage metrics**: ECS-EBS volume size, ECS-EBS cluster read, write bytes

Hyper Scale Visibility: Using Histogram Metrics for Container Analytics

As containers and microservices can be highly distributed and can generate an extremely large amount of monitoring metrics often at high velocities to keep up with the amount of monitoring data from containerized applications, Wavefront customers are rapidly turning to histogram metrics and analytics. Histogram metrics which convey detailed distribution information that gives DevOps and developer teams a comprehensive, affordable, always-on and sweeping view and early indication of incidents across all environments while retaining granular historical visibility. When evaluating using histograms against solutions such as distributed tracing, you should have in mind significant cost savings with histograms metrics compared to distributed tracing. Unlike distribution, histograms don't have any major impact on distributed code nor intense resource utilization while providing the full coverage that is on all the time.

The Wavefront platform now captures and computes distributions for all your high-velocity metrics, i.e., use its full query-driven analytics on your metrics using the histogram data type. Our software service also preserves and stores histogram data so that it can be processed later as you need. With Wavefront histogram support, you can:

- Reliably measure and aggregate quantiles/percentiles of your high-velocity metrics such as application response times and services SLAs
- Accurately measure and aggregate quantiles/percentiles of high-velocity metrics from multiple sources or other dimensions
- Calculate aggregated percentiles across multiple sources



Unified Visibility: Monitoring Complete Container Environment and Beyond

For developer and DevOps teams, the Wavefront platform provides integrated metrics monitoring and analytics beyond their containerized environment into all their existing applications, microservices, private, public and hybrid cloud infrastructures including insight into key AWS cloud services (billing, EC2, ECS, Lambda, EBS, ELB, etc.). Key differentiation of the Wavefront platform is outlined below:

Advanced Query-driven Analytics	• 100 advanced analytical functions • Detect leading indicators • Detect anomalies across apps, and cloud infrastructures
Intelligent Alerting	• Intelligent query-driven alerting helps with proactive detection of issues before customers' impact • Integrate with popular ticketing tools
Massive Scale and Availability	• Massively scalable time-series database • Built-in high availability functionalities for maximum SLAs • Metrics histogram types handle high-velocity metrics for hyper scale
Interactive, Customizable & Packaged Dashboards	• Interact and customize to unique app and cloud monitoring needs • Foster critical code, services and infrastructure troubleshooting (export, linking) • Improve DevOps team collaboration
Complete API for Extensibility	• Integrate with developer and DevOps tools both commercial & open source • Get instant visibility with pre-built OOB AWS & container integration

Case Study: FashionID

Metrics-driven, Centralized and Correlated Insight into Kubernetes Environment, Microservices and Containers

FashionID is a German online retailer that's pioneering new concepts of online fashion and apparel shopping. FashionID's developers are running about 60 microservices, and they are in the process of moving them to Kubernetes. They developers are running hundreds of containers in production. Since moving their services to containers, the FashionID DevOps team was able to maximize their cloud resource utilization while ensuring they were ensuring their online business is running smooth, no matter whether they have tens or many hundreds of containers.

The Fashion ID infrastructure and development team are using Wavefront's metrics-driven analytics to monitor their entire cloud-native microservices environment. With continuous real-time visibility into their microservices, their DevOps team can proactively resolve any customer-impacting issues. Because both business metrics and infrastructure metrics are centralized in Wavefront, they can correlate the performance of their business metrics, such as how many orders are going through, with their applications performance and infrastructure utilization. The Metrics they monitor include: the overall transaction throughput, response times (min, max, 95th percentile or others), relative to resource utilization. If there is a drop in their checkouts, they are able to immediately troubleshoot and resolve any issue.

Key Benefits

•Real-time insight into business performance by correlating business metrics with infrastructure metrics

•Visibility into microservices, containers and Kubernetes environment including resources consumption memory, CPU, network resources consumption per pod

•Microservices performance including throughput and compute, memory and other resource utilization

•Proactive alerting and enabled DevOps process with easy integration with other tooling

•Easy container scaling with continued insight in Wavefront and without rapid cost increases

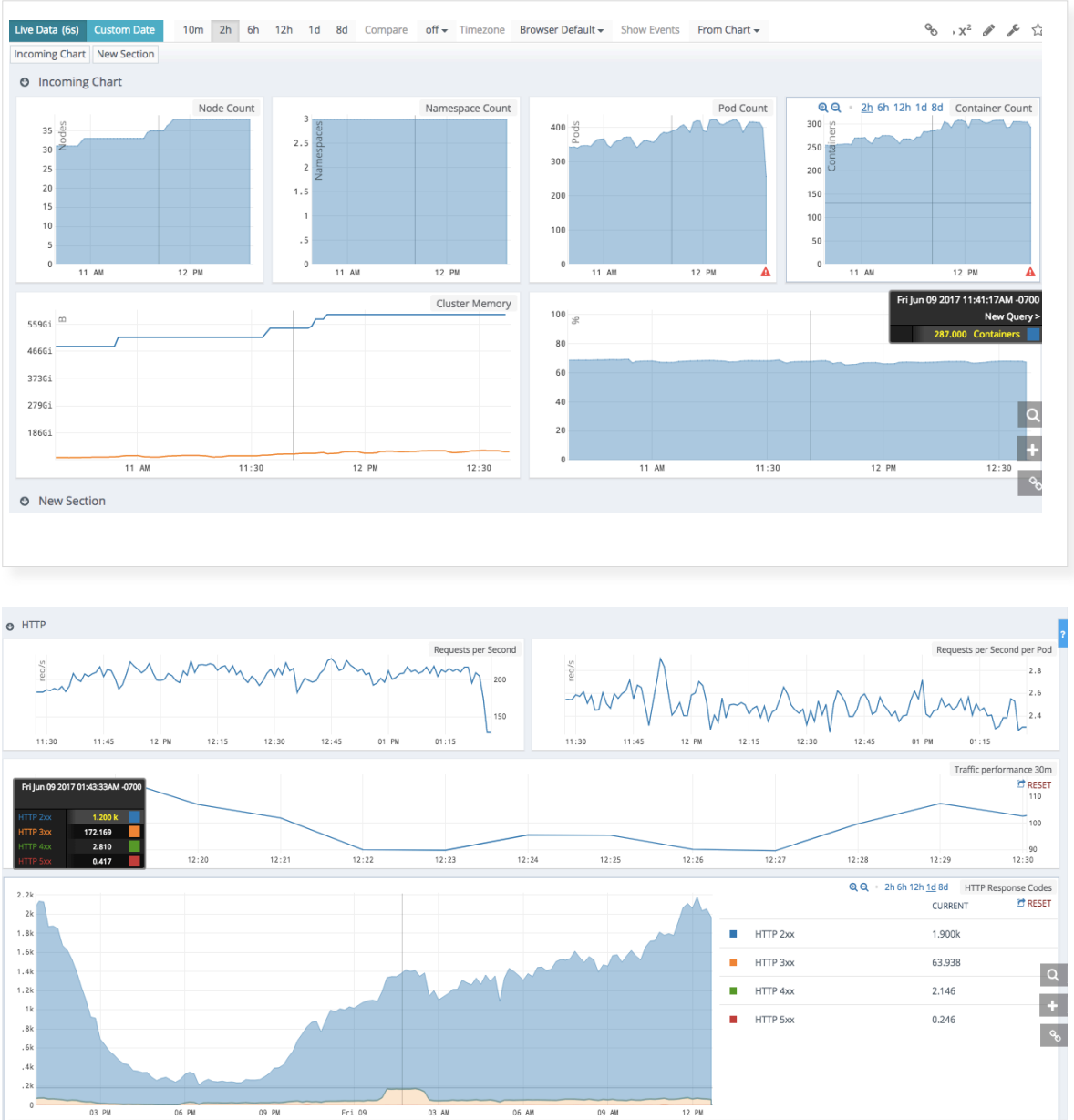
"It is very powerful to have all business and microservices metrics in Wavefront. We can see in real-time that our business and infrastructure are running as expected. Wavefront metrics give us information in the right scale and in the right point of view."

Marc Riegal
Developer/Manager,
FashionID

Case Study: FashionID

Fashion ID Kubernetes Wavefront Dashboards

Real-time Visibility into Cloud, Microservices and Containers



Dashboards for Kubernetes and containerized app monitoring.

Case Study: Mixpanel

Monitoring Container Resource Utilization and Troubleshooting Containers

Mixpanel helps teams and companies build products that users love through its analytics software for apps, websites and other digital products. The SRE and infrastructure team at Mixpanel use Kubernetes to manage their containers. They push code to production several times per day and run ~20 different containerized services, with thousands of active container instances. As Mixpanel's team moved from bare-metal to containers, they needed to maintain visibility into their applications and understand their resources utilization. Mixpanel's developers and SREs were already using Wavefront for monitoring their application metrics through statsD, and they found it very easy to expand Wavefront to ingest metrics from their containers and Kubernetes.

With continuous insight into the performance of their containerized applications, Mixpanel's engineers gained the ability to monitor their resources cost in real-time, saving hours of troubleshooting time and increasing availability of their applications. Several of their cases are described below.

Monitoring Kubernetes Cluster Utilization

Mixpanel SREs use Wavefront to understand various applications' resource utilization. They can understand if cores reserved by applications are those that are utilized when their services are running. This visibility gives them real-time savings and they are avoiding paying for unnecessary resources reservation if those are underutilized.

Removing Code and Resources Bottlenecks

Mixpanel SREs are also monitoring their typical CPU usage, memory and all underlying resources at all Kubernetes and containers levels. For example, by understanding the state of resourses usage, they can see if pods are stuck in pending state which is indicative of resources bottlenecks. This helps them quickly reallocate resources and alleviate issues maximizing application availability.

Troubleshooting Key Services & Code Performance Inside Containers

Using Wavefront, Mixpanel developers are centralizing and analyzing additional important metrics, beyond standard Heapster collector metrics, using Kubernetes API. In real-time, they can see the state of each pod and container as well as pod and container restarts which are indicative of their application issues or crashes. Compared to logging, this gives them much faster metrics-driven insights as they can easily and quickly alert on code issues before any customers are affected.

Gaining Correlated Visibility into Containers, VMs, and Infrastructure

With Wavefront metrics-driven analytics, Mixpanel's engineers can correlate metrics from various infrastructure elements and identify the root-cause of incidents. For example, they alert on increased latencies of one of their services and almost instantly figure out the cause of that latency. Since all the data is in Wavefront, they were able to identify when one of the pods where they saw container issues and latency was running on a VM where they had underlying networking issues. Wavefront dashboards helped to understand that due to network issues, CPU usage started dropping, they had less network utilization. All that was correlated in Wavefront and they were able to restart problematic VM and reschedule container on another VM which resolved the issue. Without Wavefront, it would have taken them hours to troubleshoot and resolve these types of issues. But since all metrics were readily accessible, available and centralized, it only them several minutes to diagnose the root-cause.

Why Wavefront Metrics-driven Analytics?

Know first about anomalies in your cloud application in production. Boost performance proactively. Drive accountability for everyone.

QUERY-DRIVEN ANALYTICS

The most powerful query language in monitoring running against a unified, full detail, metrics store in real-time with no limits.

INTELLIGENT ALERTING

The results are higher quality alerts, anomaly detection, and crucially valuable insights that no other monitoring tool can offer.

MADE FOR ENTERPRISES

For serious SaaS and digital businesses where performance, reliability, scale and support are essential to their business.

#1 IN SELF-SERVICE METRICS FOR DEVOPS AND DEVELOPER TEAMS

Find out why Wavefront helps you build and run great software at **wavefront.com**

Get Started with Wavefront by VMware

LEARN MORE:

Webinar: Intro to Metrics Monitoring
Webinar: To Log or to Metric
Talk: How to Use Analytics for Metrics Monitoring
Blog: Read Wavefront Blog

CONTACT US

DEMO CLIPS:

Find Signals in Noise | Data Exploration | Prevent False Alarms | Search by Behavior | Full Resolution Data | Correlation Function

WATCH NOW

USE METRIC ANALYTICS FOR:

MICROSERVICES APPLICATIONS / CONTAINERS / CLOUD / INFRASTRUCTURE

WATCH NOW

Stela Udovicic is the product marketing lead for Wavefront by VMware (VMware acquired Wavefront). Prior to VMware, while at Wavefront, as Sr. Director, Product Marketing, Stela lead Product, Solutions and Partner Marketing. Prior to Wavefront, Stela ran Splunk's DevOps solutions marketing, and executed the DevOps campaign, one of the best performing campaigns at Splunk. She led marketing activities with key strategic partnerships in DevOps space including Atlassian, Puppet, and others. Prior to DevOps, Stela was driving product marketing of Stream and Splunk's key applications including Splunk Apps for NetApp, Cisco ACI, Microsoft Exchange, all SDN apps. She was a marketing lead for Cisco, NetApp and other strategic partners. She holds a MSc in Electrical Engineering. She has presented at many major conferences, including Splunk.conf, Cisco Live, RSA, DevOps Days, PuppetConf, VMworld, NetApp Insight, and others.

Stela Udovicic
Product Marketing, VMware. **Twitter:** @stela_udo

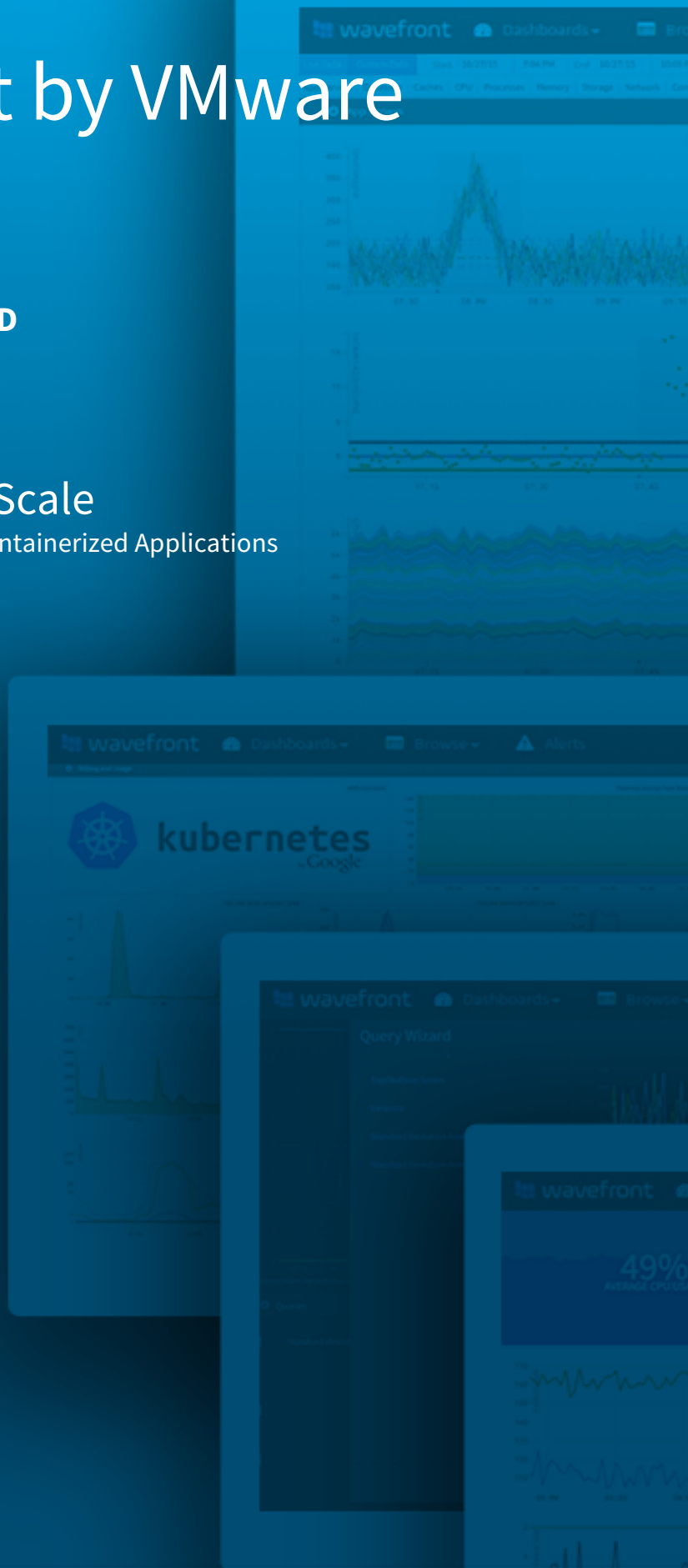
Try Wavefront by VMware for Free

WAVEFRONT.COM/GET-STARTED

eBook

Container Monitoring at Scale

A Guide to Metrics-driven Analytics for Containerized Applications



v4