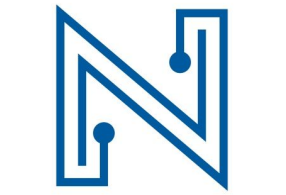
Top 5 Challenges of Container Monitoring



NETSIL

Introduction

The Proliferation of Container

Containers are increasingly becoming a critical component of modern cloud applications. Containers deliver speed and portability for software development and align closely with the prominent industry trends of DevOps and microservices.

While containers are great for software developers, they create new challenges for operations teams running containerized applications in production. Particularly the traditional techniques for application monitoring such as logging, using code-instrumented agents or tracking infrastructure metrics have hit a wall due to the fundamentally new abstraction layer of containers and their short life spans.

Container Monitoring Challenges Overview

Before we dive into container monitoring challenges, it is worth noting that containers and microservices are closely associated with each other. Containers are a crucial piece for packaging, delivery and run-time execution of services.

Therefore, a lot of challenges of microservices are relevant to containers and vice-versa. Following is a condensed list of some of the most pressing challenges with container monitoring. As an additional resource we would strongly recommend this talk by Adrian Cockroft from Gluecon as well as 'Mastering the Chaos' talk by Josh Evans from Netflix.

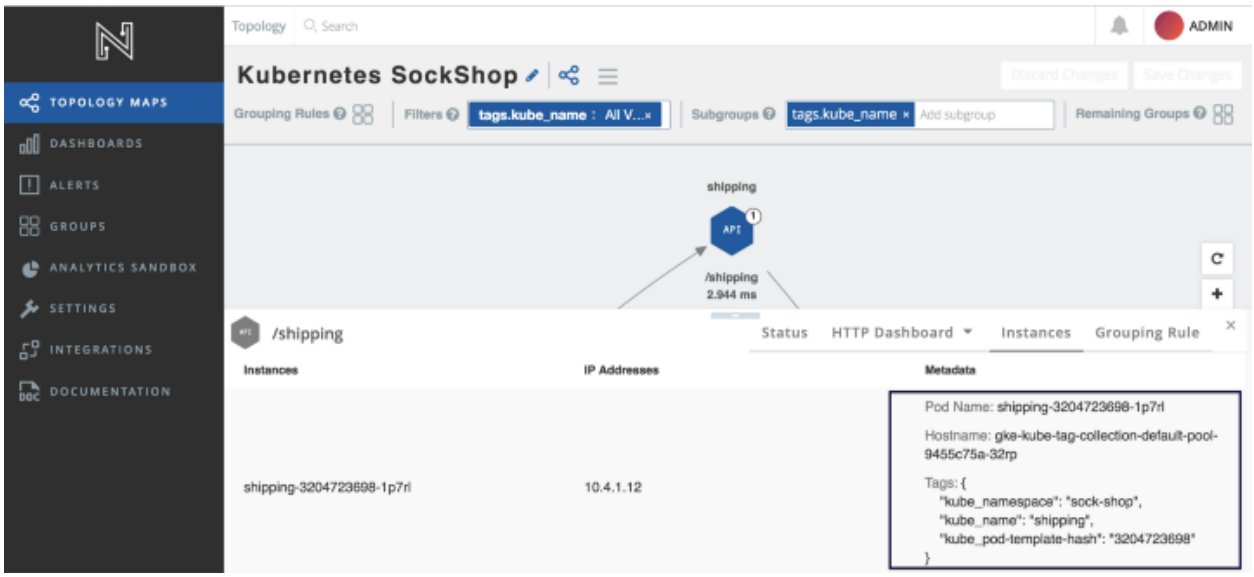
Challenge #1: Reduced Lifespan

Containers are expected to have much shorter lifespan than VMs. Reduced lifespan could be due to increased frequency of production roll-outs causing previous versions of containers to be replaced by new ones. Or it can be auto-scaling that dynamically creates and deletes containers as service load changes.

Irrespective of the cause, the reduced lifespan and fleeting nature of containers makes it hard to track them and investigate them for root cause analysis.

Netsil's Solution

This challenge is directly addressed by AOC topology maps which record all containers and their interactions along with metadata such as container_id and container_image. AOC users can go back in time and investigate the topology to understand containers that were active when any incident happened. In other words, AOC topology maps serve as dynamic, visual record of all containers as well as their interactions in the application.



Kubernetes Metadata for Pods in Netsil AOC

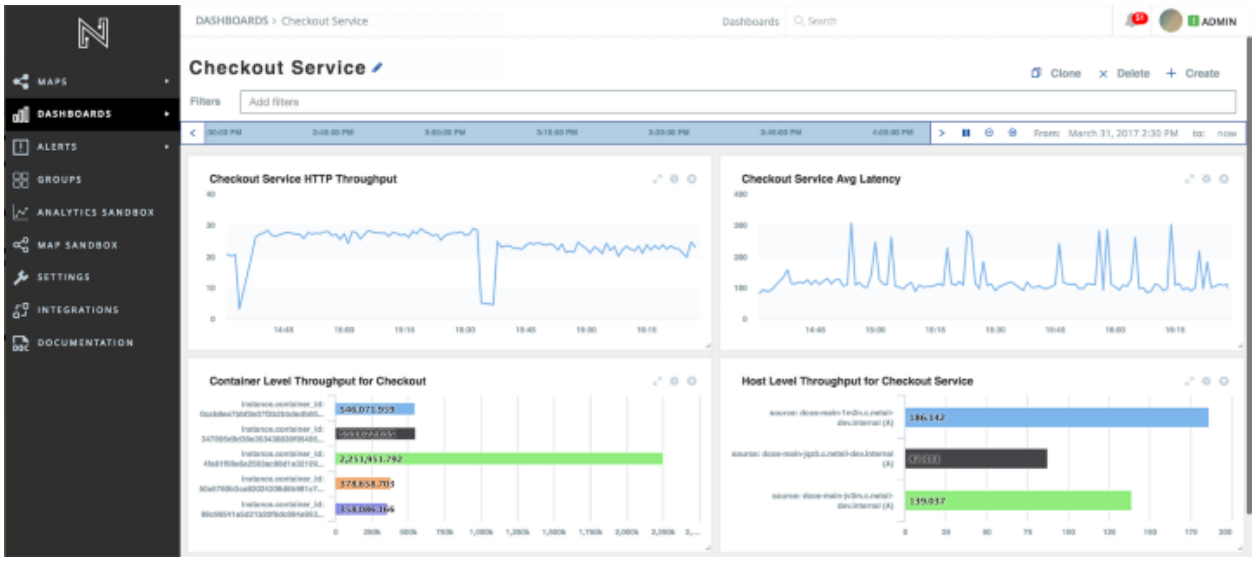
Challenge #2: Container Portability

Portability is one of the most attractive features of containers. But as the containers move, their run-time environments may change. For example, a perfectly happy set of containers could start experiencing issues if bunch of “noisy neighbor” containers are spun-up on their hosts. Or the same happy set of containers could see performance degradation if they are moved to a different server with storage or network issues.

The portability makes it very hard to correlate service level issues to container, host and infrastructure issues.

Netsil’s Solution

Netsil has a significant advantage in addressing this challenge because Netsil delivers application monitoring by observing network interactions among containers. The service, container and host are always tied together in the network packet flow. Therefore, with Netsil, users get automatically correlated, complete picture of service, container and infrastructure health.



Netsil AOC Automatically Ties Service, Container and Infrastructure Level Metrics

Challenge #3: Contained Failure

Containers are closely tied to the architectural paradigm of microservices. Microservices applications often have fault-tolerance and redundancy built in the architecture. This means that multiple redundant containers, running identical copies of code, could be powering a service within an application. In such an architecture, the failure of one container does not need to generate a paging-alert waking up SREs in the middle of the night.

Most legacy monitoring tools are not equipped to handle such situations and they end up flooding operations teams with low-level, low-impact alerts.

Netsil's Solution

Netsil addresses this by introducing the concept of service-level monitoring. Services are essentially groups of containers performing functionally equivalent tasks. Operations teams can either use the built-in grouping algorithm in the AOC or define custom grouping rules for containers. They can then monitor and set alerts on the health of services as a whole rather than worrying about individual containers.

Golden Signals for Kubernetes Service

Challenge #4: Cascading Failures

In modern applications, multiple components interact among each other to fulfill transactions. It is well understood that failure of one component in the dependency chain will have cascading effect on the entire application.

Another type of cascading failure happens when few containers that are part of a load balanced service start experiencing issues. When few containers fail to handle their share of load, then load starts to increase on other functional containers. If the situation is left unchecked, then, eventually, all the containers that are part of the service start performing poorly resulting in a spike in latency, throughput and/or error rates for the service.

This can be viewed as an initial lateral propagation of failure within the service component, before the failure starts impacting other services in the dependency chain.

Netsil's Solution

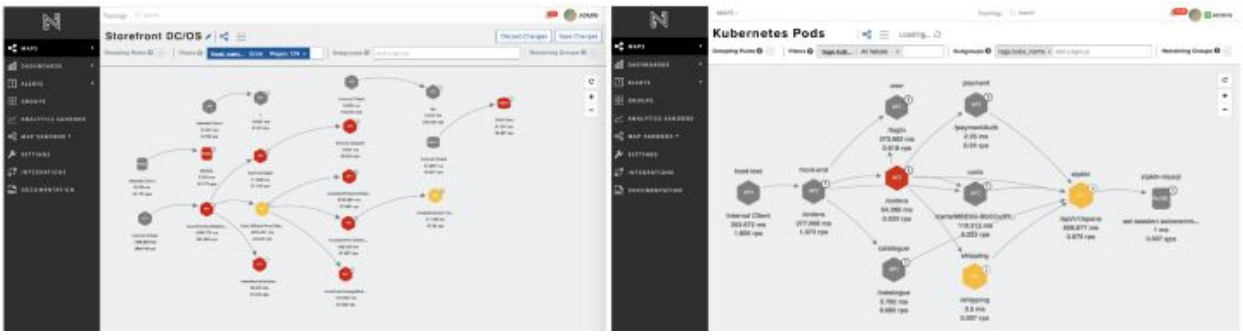
Cascading failures is one of the reasons why SREs everywhere emphasize the importance of monitoring service-level golden signals of latency, error rates, throughput and saturation. In the AOC, operations teams first leverage the topology map to easily understand service dependencies and identify the root cause of cascading failures. Secondly, each node and edge in the topology map has their dedicated golden signals dashboard. So the operations teams get complete visibility into dependencies as well as the golden signals for each service.

Challenge #5: Container Framework

In order to manage collections of containers specialized frameworks such as Kubernetes, Mesosphere DC/OS and Docker swarm are used. These frameworks have their own abstractions such as pods, services, namespaces, etc. which are valuable for operations teams to understand and monitor containerized applications.

Netsil's Solution

The AOC integrates closely with Kubernetes, Mesosphere DC/OS, Docker and AWS ECS. The figure below shows a map of Kubernetes cluster and of a Mesophere DC/OS cluster.



Auto-discovered Maps of Mesosphere DC/OS and Kubernetes Applications

Conclusion

Container Monitoring Simplified with Netsil AOC

Containers are great productivity booster for software industry. Using Netsil Application Operations Center (AOC), operations teams can successfully embrace containers in production environments. Starting with the real-time topology map, AOC provides complete visibility into health and performance of services, containers and infrastructure.

You can get started with AOC in less than 10 minutes using our single container manifests that work with Mesosphere DC/OS, Kubernetes and Docker. Get started here.

About Netsil

Netsil is pioneering an innovative approach that leverages service-interactions as the source of truth for monitoring containerized applications. This approach greatly simplifies container monitoring. The Netsil Application Operations Center (AOC) enables operations teams to gain complete visibility into containers and frameworks such as Kubernetes and Mesosphere DC/OS, without any code change to containers or applications.

Netsil has recently announced single container manifests of the AOC. In less than 10mins, you can start mapping and monitoring your container clusters with the AOC. Manifests are available for Docker, Kubernetes and Mesosphere DC/OS environments.

Netsil Application Operations Center (AOC) Overview

The Netsil Application Operations Center (AOC) is the converged application monitoring platform for operations teams. SREs and DevOps engineers use the AOC to monitor and deliver on Service Level Objectives (SLOs) for business critical applications. The AOC delivers the industry's first auto-discovered, real-time topology map of the entire application. The application topology map is combined with per-second, real-time metrics from the entire stack. The operations teams leverage AOC to:

- **Create multiple maps for various components of applications and understand dependencies among services.** The AOC doesn't require any code change to application or container images. It listens to service interactions and auto-discovers the application map. For e.g., starting with the initial auto-discovered topology map, operations teams can zoom-in on specific frameworks such as Kubernetes and create a map using kubernetes tags such as pod names (see fig 1 below).
- **Alert and diagnose production issues using AOC's real-time metrics and analytics.** For the metrics, the AOC leverages not only the data from service interactions but also obtains metrics from system specific sources such as cAdvisor for Docker and Kubernetes metrics. For e.g., starting with service-level health, operations teams can progressively drill down and investigate container, host and infrastructure-level metrics to identify the root cause.