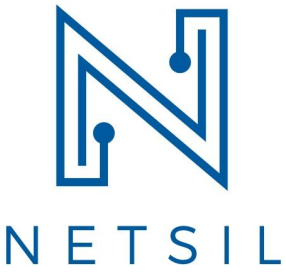
Kubernetes Monitoring with Netsil



Overview

Kubernetes has emerged as the undisputed leader among container orchestration systems. Kubernetes not only does scheduling and orchestration but also provides holistic solution for service routing, service discovery, load balancing, deployment management, etc. However, Kubernetes sophistication brings its own challenges particularly when it comes to monitoring production clusters of Kubernetes. The success of Kubernetes deployment is going to depend on a strong monitoring solution that helps Site Reliability Engineers (SREs) and operations teams to visualize the deployment, monitor key SLAs around services and zoom into the problem as quickly as possible.

Netsil has created the industry's first "Google map for your cloud app". Netsil's innovative approach auto-discovers real-time map of entire application including Kubernetes clusters. SREs and Ops can map Kubernetes from multiple levels - deployments, namespaces, services or pods. Additionally, Netsil has real-time, per second metrics for monitoring the entire Kubernetes cluster. And the best part about Netsil is that the map and monitoring metrics do not require any code change to containers or applications!

In this whitepaper, we will provide overview of Kubernetes operational challenges. Then we will provide details on Netsil's "map and monitor" approach for addressing Kubernetes monitoring challenges.

Kubernetes Monitoring Challenge

Kubernetes monitoring challenges are a combination of container monitoring challenges and additional considerations from Kubernetes framework.

Container Monitoring Challenges

These challenges are specific to Container Management. They deal with container lifecycle, portability, monitoring individual v/s overall application and cascading failures. Let us look at those in detail:

Container Lifecycle and Portability: Containers are short-lived and can be scheduled across multiple hosts and even across environments. This presents challenges in monitoring an ever shifting container workload across environments, especially when you need to zoom into an issue and investigate via root cause analysis. While container portability is a key reason for its adoption, the containers could experience a degradation in performance due to a different environment or host machine issues. ince you need a different approach to viewing the overall health of your application in terms of high level abstraction of services, monitor those and then zoom-in in case of issues.

Monitor Workloads v/s Individual Instances: Due to the brevity of Container Lifecycle and the fact that they could be brought down and scheduled automatically across hosts, you need a approach that focuses on high level construct of monitoring services, which gives a better indication of the overall health of the system rather than individual containers and hosts. It is not optimum to watch an individual instance and observe the metrics from that instance, since you need to look at the whole application. A failure on a particular node is not necessarily a signal on the health of the application.

Isolated and Cascading Failures: In a Microservices application, multiple services interact with each other to fulfill a transaction. Each service is powered behind the scenes by containers. The failure of containers behind a load balanced service, puts in additional load on the remaining services. This is a starting step to not only degrade that particular service but could have a cascading effect on the performance of other services that depend on the former one. The challenge then is to get notified of multiple golden signals in a service (error rates, latency, throughput and saturation) as early as possible to avoid its failure cascading over to other services and impacting the overall application.

Kubernetes Considerations for Monitoring

Mapping to Kubernetes Abstractions: Kubernetes brings in new abstractions like Pods, Services, Namespaces and more. It is important that any monitoring solution embraces these abstractions and helps operations teams visualize and monitor the systems in an easier fashion. It is important that we look at services rather than individual instances and machines and are able to integrate into features like Kubernetes labels and namespaces to help visualize the deployments.

Monitoring Kubernetes: It is important to monitor the key parts of Kubernetes provided components to ensure that your deployment is healthy. This includes key Kubernetes components like etcd, kubelets, kube-proxy, and scheduler. As an example, etcd, which holds the critical configuration settings and service discovery details for your cluster, should be monitored for consistency, latency rates and consensus protocol metrics.

Insights into Kubernetes Internals: Kubernetes is a complex piece of software with multiple components providing support for service discovering, scheduling, container management, agents, service routing and more. These parts all come together to serve a request, but what will happen if a service invocation fails? In a distributed system with multiple moving parts and infrastructure services, one challenge is to trace the entire path of that failed invocation and determine the exact reason for what went wrong. Did the Service Discovery fail? Did the service not get routed? Did the Node have enough resources to schedule the Pod? A Monitoring solution needs to map this flow out and help you zoom into where the failure occurred.

Netsil's Approach to Kubernetes Monitoring

Netsil provides a novel approach such that operations teams can successfully embrace containers in production environments. Netsil's approach does not rely on collecting metrics from instances since containerized environments are highly transient across the infrastructure and to construct a high level application context out of these metrics is difficult. At the core of Netsil's philosophy is to collect network interactions in real-time and use them as the source-of-truth for app observability. These network interactions or services interactions is a more reliable way of understanding the health of your overall application.

How does Netsil do Monitoring?

Netsil uses Packet Capture (pcap) to capture network traffic and forwards them to a cluster of Stream Processors (SP). The SP reconstruct the application layer protocol based on these packets. The TCP/IP packets are reconstructed to classify the interactions into different protocols and services. For example the SP's can determine if it is a HTTP interaction based on the HTTP Request/Response protocol. In a similar fashion, the SP can determine if it is a MySQL, DNS or other protocols. In addition to identifying the protocol or interaction, the packets are also analysed for multiple golden signals like latency, error rates, throughput and more. In addition to pcap, Netsil also employs agents like dd-agent and statsd for infrastructure metrics of Docker Containers.

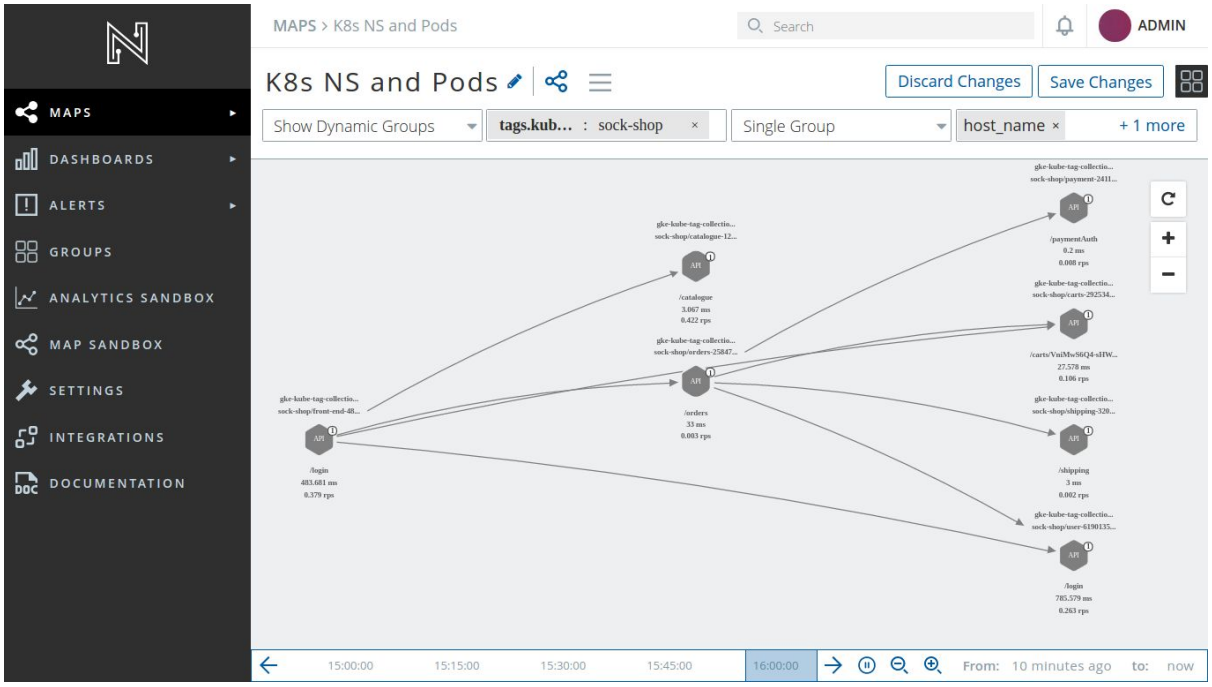
Netsil now can marry the insights that it gets from pcap processing with various Kubernetes artifacts like Pods, Namespaces, deployments and more. And all of this is done while retaining a framework-agnostic approach with no change to the application code.

In an earlier section, we discussed the challenges in monitoring Kubernetes workloads. Netsil addresses those challenges via unique features described below:

Topology Maps

Netsil AOC auto-discovers maps of your Kubernetes applications and is able to present the operators a simplified view of the application. The AOC collects existing metrics, uses service discovery, and Kubernetes features like labels and namespaces to present a top-down service topology view of the application. Services are automatically identified by grouping together instances in real-time based on common characteristics without any input from the user or relying on metadata.

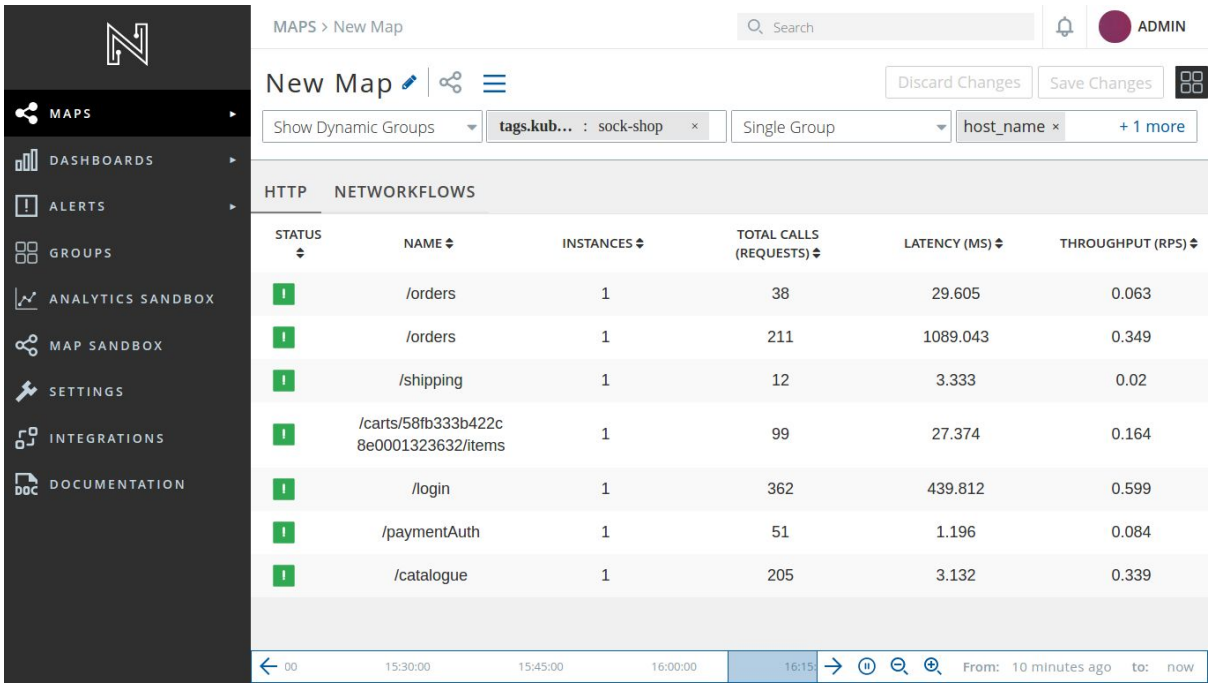
Netsil is able to create this topology map based on information that it gets from cAdvisor. It automatically detects a new container via Docker events subscription and then pulls the Kubernetes Metadata via cAdvisor. Each Pod information is then forwarded to the Stream Processor (SP) that is running on AOC.



Since Netsil uses pcap processing, it captures the inner workings of Kubernetes such as DNS proxies, health checks, etc. As a result, you can understand any Kubernetes issues like service routing right via the Topology Map.

Netsil comes along with default Topology Maps which filter by Kubernetes Namespaces, Hosts, Pod Templates, etc., but the operator can also create their own Topology Maps by creating dynamic grouping based standard labels like kube_namespace and then viewing the services by grouping the host_name and pods together. The screenshot above shows a list of all services in the Microservices reference application : Sock Shop.

An alternative list view of all the services and how they are performing in terms of interactions, latency and any errors is shown below:

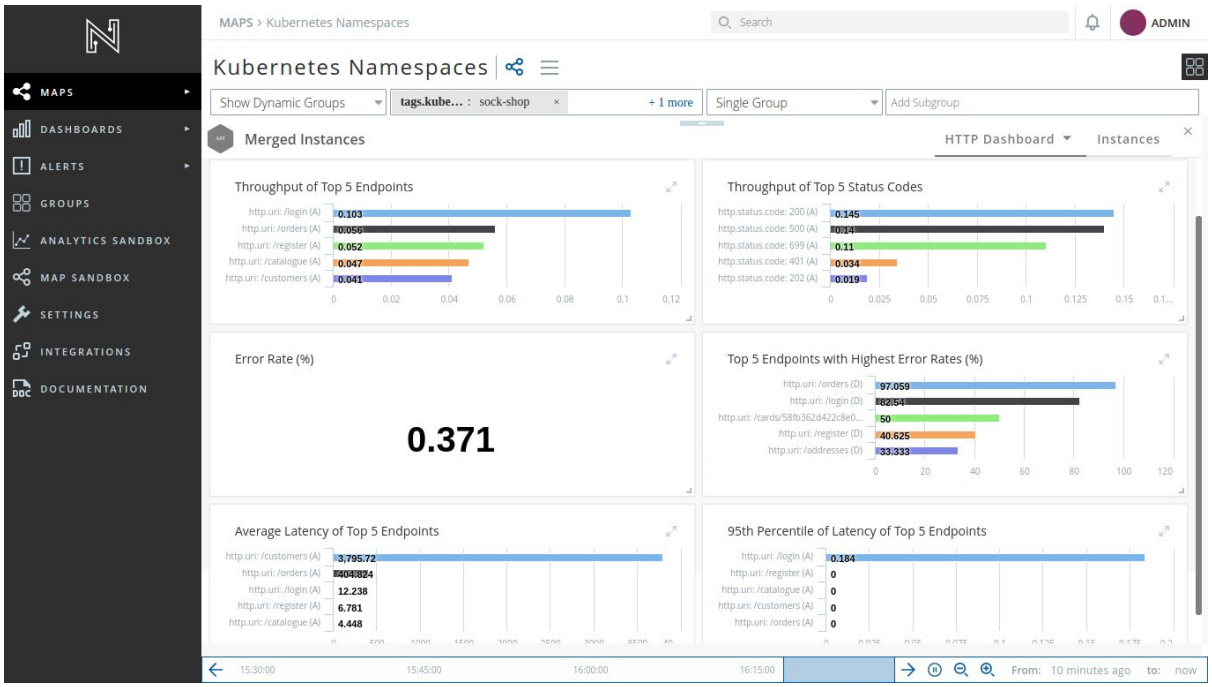


Service Level Monitoring

Netsil uses service level monitoring and it gives the operator a unified view of all the services in the application. 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.

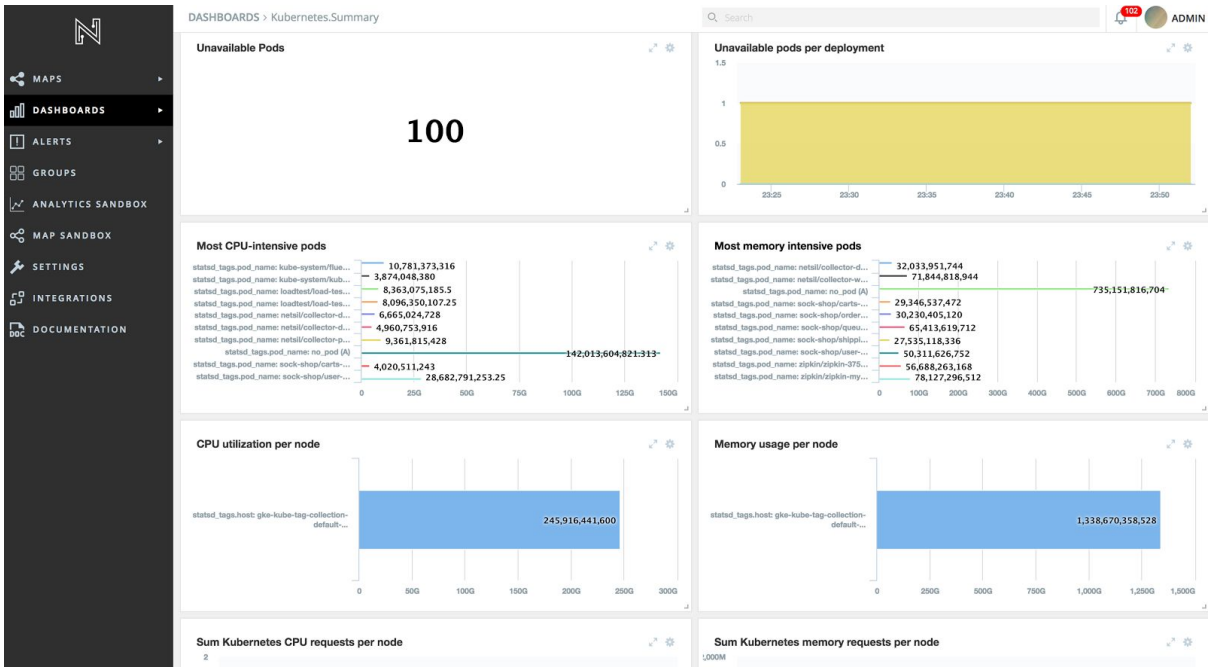
Netsil is able to achieve this by merging two key pieces of information at the Stream Processor level. It merges the packet traffic that has originated from Pod IP and combines it with the Pod metadata, correlating it with the Pod Id, since it is part of the Pod metadata that was forwarded to the Stream Processor.

The ability to define alerts at service level helps prevent flooding the support team with multiple operational level alerts and is a key differentiator in monitoring Kubernetes deployments since containers can move across environments and see short lifecycles.



Kubernetes Infrastructure Visibility

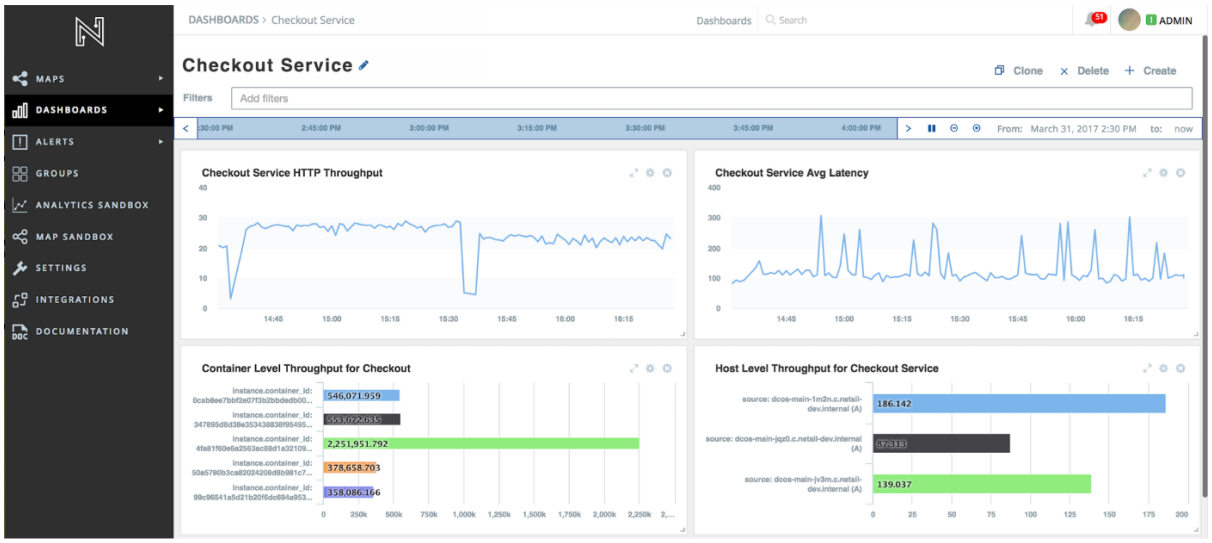
From an infrastructure perspective, it is important to have a high level view of the Kubernetes cluster. The Kubernetes Summary dashboard gives you a bird's eye view of infrastructure with vital metrics about network, compute and storage infrastructure. This level of information can be a great input to capacity planning and keeping an eye on overall usage.



Integrated Analytics

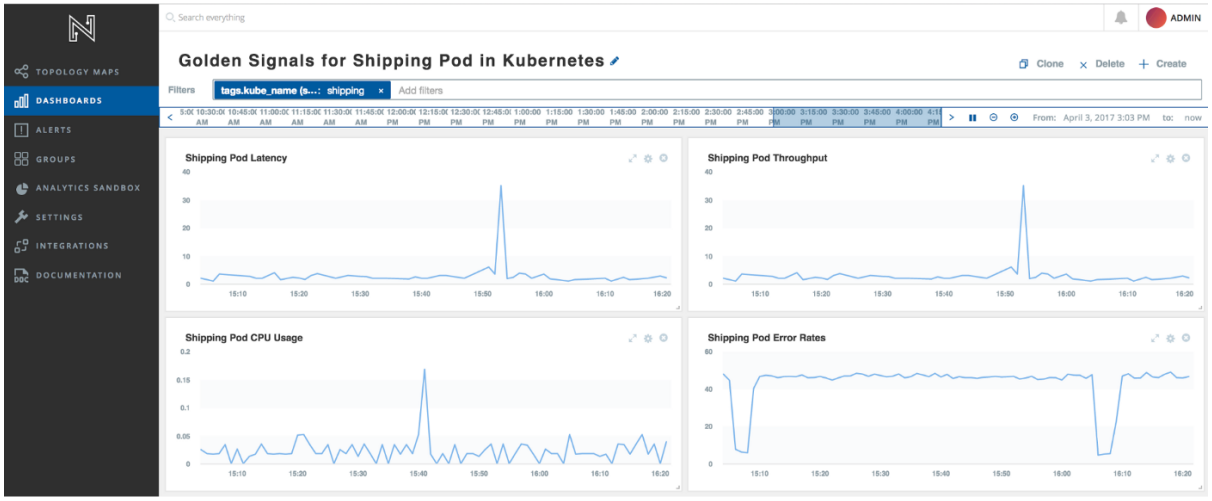
In case of an incident, it is important for SRE Teams to be able to zoom into the actual problem based on the service that is suspected of having a problem. Given that containers could have been destroyed and moved across infrastructures, it is difficult for traditional monitoring systems to recreate the scenario and more so to go back in time.

Netsil allows operators to set up custom SLAs on services and observe the golden signals for the service i.e. latency, error rates, throughput and saturation on any service.



You can then zoom right into the pods and containers that make up this service. Netsil captures infrastructure metrics for containers inside a pod via the dd-agent Docker integration. For containers running in Kubernetes these stats are tagged by the metadata for the pod. Filters can be used to further slice and dice based on the metadata.

All container interactions along with their metadata such as container_id and container_image are recorded and the operators can go right down to the pod and host level to investigate the incident.



Netsil ties down the analytics right from the service level to pods and eventually the hosts via the Netsil AOC topology maps. This is the key feature that helps SRE teams to investigate and perform root cause analysis in an efficient fashion. The Time Travel feature allows the operators to resolve production issues collaboratively and in real-time.

Conclusion: The Benefits of Netsil AOC

- Netsil AOC Topology Maps provides you with automatically correlated and complete picture of service, container and infrastructure health for your Kubernetes Applications.
- Reduce Alert fatigue by visualizing your application as a set of services with custom SLAs on each of the service.
- Monitor golden signals for your service: latency, error rates, throughput and saturation and zoom into the pod and container level as needed.
- Recording of each container interaction allows your SRE Team to understand the problem via the Time Travel feature and resolve production issues collaboratively.

Getting Started with Netsil AOC

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

The Installation is done in two parts: Netsil AOC and the Netsil Collectors. The collectors are installed on your application instances (VMs or containerized environments) and mirror the service interactions & metrics back to Netsil AOC for real-time analysis.

For more information on getting started with Netsil, go to
https://docs.netsil.com/setup-guide/installation/k8s/