May 2022

NETWORK MONITORING OF CONTAINER CLUSTERS IN PRIVATE, PUBLIC, AND HYBRID ENVIRONMENTS

Shyam Kapadia

Anil Jangam

Deepika Sharma

NETWORK MONITORING OF CONTAINER CLUSTERS IN PRIVATE, PUBLIC, AND HYBRID ENVIRONMENTS

AUTHORS:
Shyam Kapadia
Anil Jangam
Deepika Sharma

ABSTRACT

Kubernetes (https://kubernetes.io/) has become the de-facto standard orchestrator for deploying modern cloud-native applications that employ micro-services-based architectures in which such micro-services are typically deployed via containers. Kubernetes uses abstractions of pods, namespaces, labels, selectors, specs operators and other various constructs that make it easy for application developers to rapidly deploy their workloads for test, development, and production environments. Such mixed/hybrid cloud environments make it challenging for Information Technology (IT) operators to manage, monitor, and troubleshoot such environments and implement consistent security policies. Accordingly, presented herein is a novel user interface dashboard through which automated network monitoring, troubleshooting, and visibility of containerized workloads can be viewed/managed, regardless of where the workloads are deployed in the cloud.

DETAILED DESCRIPTION

Many major cloud providers provide a managed Kubernetes (also known as "K8s") as a Service (KaaS) offering. K8s itself can be deployed on bare-metal servers directly or on virtual machines. In on-premise environments, K8s is typically deployed on server hardware that may be virtualized via virtual machine (VM) hypervisors.

In this new hybrid world, where K8s clusters will be deployed in a mixed environment, namely on-premise data centers and a mix of public cloud vendors, it becomes challenging for Information Technology (IT) operators to manage and monitor such environments such that an appropriate set of security policies are enforced. Further, troubleshooting can be especially challenging when clients in the on-premise or branch environments are trying to access services hosted in K8s clusters in private, public, or hybrid clouds.

Current on-premise data center (DC) network environments are typically deployed using fabric-enabled network devices. For example, bare-metal or virtualized servers are deployed in racks with appropriate Top-of-Rack (ToR) switches. External (aka North-South) traffic from workloads out of the data center are communicated via switches with appropriate border roles. In some instances, there might be multiple on-premise data center locations spanning geographical boundaries, that, in turn, are interconnected via some data center interconnect (DCI) technology. In addition, on-premise data centers may have connectivity to various public clouds, again typically controlled via a set of border devices and service appliances. Such connectivity maybe via the public Internet or some form of dedicated connection. Customers that utilize such deployments typically prefer some form of security (e.g., Internet Protocol Security (IPSEC)) and/or encryption for all traffic going to the public cloud.

Some cloud solutions provide a secure mechanism for connecting and extending consistent security policies from on-premise data centers to public clouds, as well as in hybrid cloud environments. Such solutions typically offer a front-end or gateway in the public cloud, for all traffic that will go to/from the on-premise DC from/to the cloud. On the on-premise side, border devices serve as the gateway for all traffic from/to the public cloud. A cloud infrastructure controller, which may manage/control application policies for an infrastructure, can be provided in connection with such current solution in conjunction with on-premise DC controllers, which makes it possible to realize such a solution from an orchestration point of view. While current cloud orchestration solutions provide an elegant mechanism through which secure connectivity can be orchestrated in private/public/hybrid cloud environments, such solution do not provide visibility into K8s deployments.

There are many off-the-shelf tools in the open-source world that offer monitoring of K8s clusters themselves such as monitoring compute health, resource utilization per pod, node, namespace, container, etc., various processing resource usage information, memory usage information, storage metrics and/or the like. However, such tools merely offer visibility that is cluster-scoped and have almost no bearing on how such information correlates to a network state. With multi-cluster K8s deployments, where some may be on-premise and some may be on different public clouds, some key metrics and visibility is often needed for troubleshooting scenarios in which clients and servers may span across different environments. For example, a client on an on-premise data center may want to access a web service advertised by a Virtual IP (VIP) behind a load balancer that, in turn, is being serviced by a set of replicated servers deployed in the public cloud. Thus, it may be difficult to monitor such inter-domain traffic of interest on-demand and in a direct manner so that network operators can have a consistent visibility and troubleshooting capabilities for K8s workloads, irrespective of where they are deployed.

This proposal provides a user interface dashboard through which K8s workloads may be monitored, regardless of where there are deployed. As noted above, a typical DC environment typically includes an on-premise infrastructure and/or fabric controller managing an on-premise data center and a cloud infrastructure controller managing the managed, public cloud environment. In accordance with this proposal, a user interface dashboard can be provided that serves as a common stitching point for operators to manage and monitor such a hybrid-cloud environment.

Recall that in a typical data center, all North-South traffic in and out of a data center egresses out of a pair of border devices. Similarly, on the public cloud side, a cloud router that is managed and instantiated by the cloud serves as the cloud gateway for all inter-cloud or cloud-to-on-premise traffic. Hence, all inter-cluster traffic in and out of K8s clusters deployed either in on-premise environments or in the public cloud, have specific ingress/egress points through which traffic is to flow, such as a border router for traffic to on-premise K8s clusters or traffic from on-premise clients to K8s clusters running in the public cloud.

Consider an example scenario, as illustrated in Figure 1, below, in which clients in an on-premise data center are trying to access a service hosted by a K8s cluster in a particular public cloud, shown in Figure 1 as Public Cloud #1). The deployment of the K8s cluster in Public Cloud #1 may be managed by a service offering of the cloud provider or it may be managed via K8s being deployed natively by an operator. For an application service hosted by the K8s cluster that needs to be advertised externally, the service is typically associated with a load balancer VIP that, in turn, is exposed via an elastic IP (Internet reachable). It is to be understood that the service provider for Public Cloud #1 will perform the appropriate Network Address Translation (NAT) for external traffic directed to the elastic service IP so that it is translated to the native K8s cluster address space. In some instances, an appropriate Domain Name System (DNS) name may be associated with the well-known service, so that it can be discovered and addressable. Further, it is to be understood that the service provider for Public Cloud #1 provides mechanisms to integrate DNS entries with on-premise environments. As discussed in further detail below, the user interface dashboard as proposed herein can provide a data center network administrator with the ability to monitor, troubleshoot, and have visibility into this traffic.

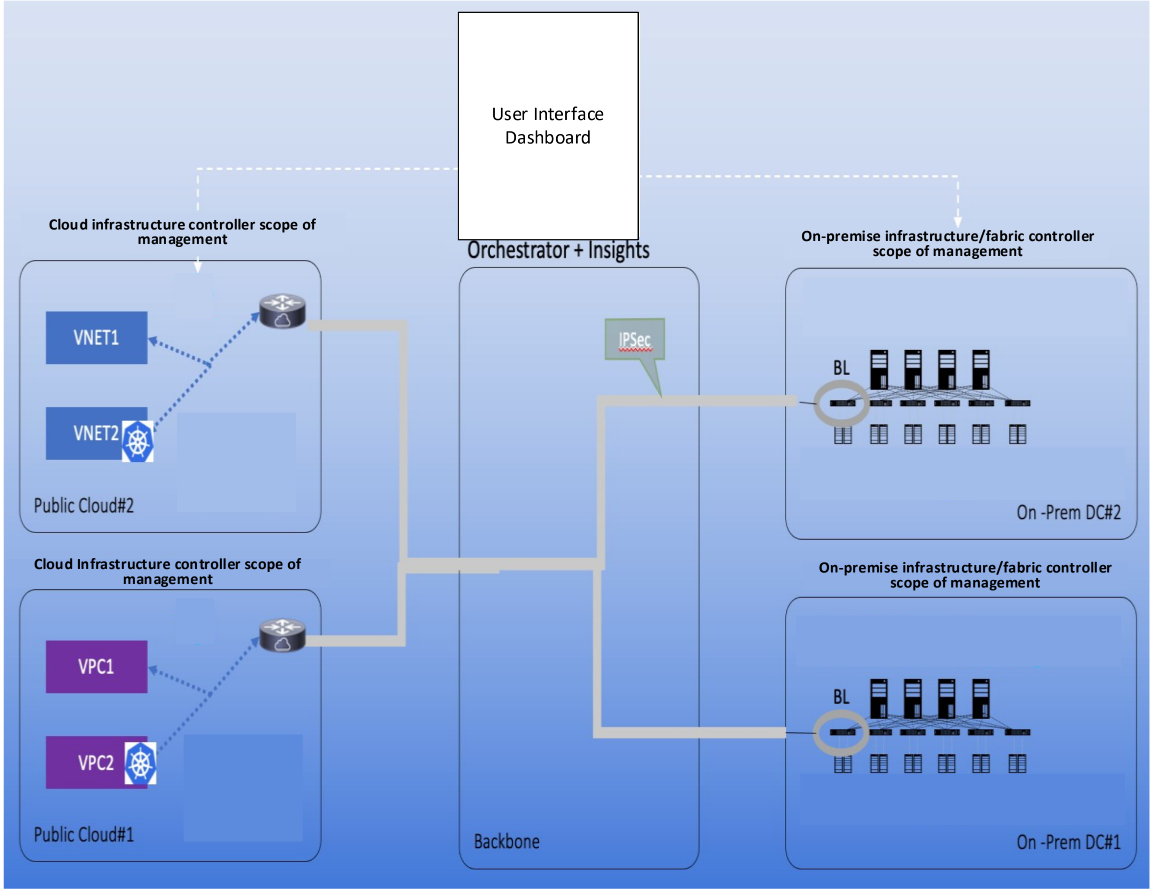
User Interface Dashboard
Cloud infrastructure controller scope of management
Orchestrator + Insights
On-premise infrastructure/fabric controller scope of management
VNET1
VNET2
IPSec
BL
Public Cloud#2
On -Prem DC#2
Cloud Infrastructure controller scope of management
On-premise infrastructure/fabric controller scope of management
VPC1
VPC2
BL
Public Cloud#1
Backbone
On -Prem DC#1

Figure 1: Example User Interface Dashboard to Monitor Workloads

For the user interface dashboard, as illustrated in Figure 1, it is assumed that the on-premise controller sites and the cloud controller sites would have already been on-boarded. In addition, the network administrator will have the ability to onboard the K8s cluster of interest involving Public Cloud #1. This can be performed, for example, via a standard Client/Cluster Certificate Signing Request (CSR) so that the dashboard will be able to make appropriate K8s Application Programming Interface (API) calls to an API server and register for appropriate notifications of interest. Read-only privileges may be sufficient for this purpose.

Using the standard K8s APIs, information about the cluster, including the pods, namespaces, load balancers, computes etc. will be gleaned by the dashboard. In addition, since the cluster is deployed in a public cloud, and the public cloud infrastructure is being orchestrated by a cloud infrastructure controller that interfaces with the APIs, the dashboard will be able to obtain further information about the K8s cluster, including the well-known elastic IP and port combinations that are exposed by the cluster and mapped to appropriate network resources by the cloud service provider. This will correspond to the well-known application services that are hosted by the K8s cluster.

The information about the well-known (IP, port) combination will then be programmed as specific access control list (ACL) flow rules of interest in an application provided by the dashboard. In this example, such a rule may be a (*, *, Destination IP, Destination port) rule that can be automatically pushed to border devices in the on-premise data center, so that they can begin performing flow monitoring for the client traffic being sent to Public Cloud #1 hosted application K8s service. The border devices, based on cloud scale Application Specific Integrated Circuits (ASICs), can support hardware telemetry in terms of monitoring flows, buffers, latency etc. at a fine-grained level based on the programmed rules.

Information about these flows can then be captured into the dashboard data lake, suitably analyzed, correlated, etc., and deep insights can be offered. Similarly, the flows can also be captured via a monitoring technology provided by the cloud gateway devices dashboard to cloud infrastructure controller triggered interest.

Any flow drops, latency changes, throughput, etc. involving these flows can be monitored through an application that provides network insights. Appropriate anomalies based on configured severity levels can be automatically provided to various network administrators (e.g., networkops, cloudops admins, etc.).

As this information is collected across various K8s clusters over time, the various insights, anomalies, and performance metrics can then be fed back into the system so that it can also make recommendations on what workloads are best suited for on-premise K8s clusters versus K8s clusters deployed in the public cloud. Even within public clouds, there can be recommendations on which public cloud maybe more suitable based on the historical data collection and metrics analysis.

While the above example involves a client that resides in the on-premise data center, with the K8s cluster was hosted in a public cloud, the mechanism for visibility proposed herein may be similarly applicable for all kinds of deployments involving K8s clusters in any mix of private, public, and/or hybrid cloud environments. The abstraction provided by cloud infrastructure controller that communicates with the public cloud provider's specific APIs allows the user interface dashboard to be public cloud agnostic, while still providing the same consistent level of visibility, troubleshooting, and network monitoring capabilities across all cloud environments.