**THUYẾT MINH GIẢI PHÁP PALO ALTO CN-SERIES CONTAINER FIREWALL**

1. **TỔNG QUAN**

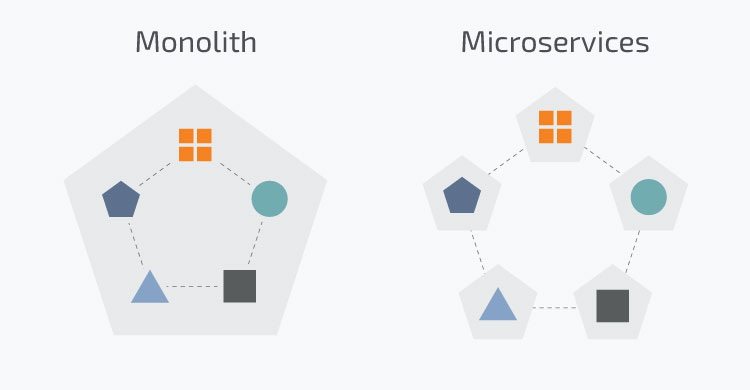
Việc đảm bảo ứng dụng hoạt động ổn định và đáng tin cậy khi chuyển giao giữa các môi trường khác nhau luôn là mối quan tâm của tổ chức. Trước tình hình đó, công nghệ Container đã ra đời, giúp đóng gói các ứng dụng cùng thành phần phụ thuộc và triển khai chúng trong các môi trường khác nhau mà không cần chỉnh sửa lại ứng dụng.

1. **Mối liên hệ giữa Container và kiến trúc microservices**

Kiến trúc microservices yêu cầu phải tách biệt các thành phần ứng dụng chính thành các phần độc lập, riêng biệt. Việc tách rời này cho phép các thành phần này hoạt động độc lập, giảm khả năng xảy ra lỗi hoặc gián đoạn toàn bộ dịch vụ.

Container đóng vai trò là vùng chứa cho các tác vụ hoặc microservices riêng lẻ. Bằng cách phân chia nhỏ từng chức năng ứng dụng thành container riêng, nâng cao khả năng phục hồi và khả năng mở rộng.

Hơn nữa, quá trình đóng gói cho phép cập nhật từng phần mà không ảnh hưởng đến những thành phần khác. Điều này tạo điều kiện cho việc triển khai nhanh chóng các nâng cấp về tính năng mới, giảm thiểu sự gián đoạn hoạt động trên toàn hệ thống.



1. **Các thành phần trong kiến trúc bảo mật Container**

Kiến trúc bảo mật container được thiết kế để bảo vệ container trong suốt vòng đời. Quá trình này bao gồm một số thành phần chính, mỗi thành phần tập trung vào một khía cạnh khác nhau của việc bảo mật Container. Các thành phần chính bao gồm:

* Bảo mật image của Container (Container images security).
* Bảo mật nơi chứa các image của Container (Registry security).
* Bảo mật Container (Container engine security).
* Bảo mật các điều phối hoạt động Container (Orchestrators security).

1. **Lý do phải bảo mật cho Container**

Bảo mật container đề cập đến việc bảo mật các **ứng dụng** được chứa trong container và cơ sở hạ tầng hỗ trợ chúng.

Vì việc container hóa đóng vai trò quan trọng trong quá trình phát triển phần mềm ngày nay nên bảo mật container là điều vô cùng cần thiết đối với các tổ chức, doanh nghiệp.

Các container chia sẻ cùng một kernel với máy chủ, vì vậy nếu một container bị xâm nhập thì có nguy cơ kẻ tấn công có thể khai thác lỗ hổng tiềm ẩn để ảnh hưởng để ảnh hưởng đến hệ thống máy chủ và các container khác bên trong máy chủ đó.

Vì các container được tạo từ image và những image này có thể chứa lỗ hổng nên kẻ tấn công khai thác chúng có thể giành được quyền truy cập vào các thành phần khác của hệ thống.

Cho dù ứng dụng chạy trên máy chủ bare-metal, máy ảo hay container thì đều phải đối mặt với cùng các mối đe dọa, ví dụ như ransomware, cryptojacking và botnet.

Các container thiếu khả năng bảo vệ chống lại các lỗ hổng chưa được vá và chưa được xác định. Các lỗ hổng của ứng dụng không phải lúc nào cũng được biết đến. Trong một số trường hợp, lỗ hổng được phát hiện sau rất nhiều năm tồn tại. Ngoài ra, khi một lỗ hổng được xác định và bản vá được cung cấp, có thể mất nhiều tuần hoặc thậm chí nhiều tháng để vá lỗ hổng.

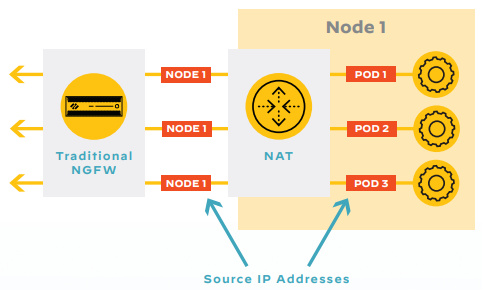
1. **Next Generation Firewall**

**Next Generation Firewall** cung cấp khả năng kiểm soát luồng dữ liệu tới tầng ứng dụng (layer 7) trong mô hình OSI thông qua kiểm tra application-level inspection, intrusion prevention, threat intelligence, URL filtering.

NGFW kiểm soát luồng dữ liệu từ các ứng dụng đi ra bên ngoài (outbound traffic), kiểm soát truy cập giữa các ứng dụng với nhau, kiểm soát truy cập từ bên ngoài vào các ứng dụng (inbound traffic).

NGFW đóng vai trò quan trọng trong việc đảm bảo an ninh an toàn cho hệ thống on-premises và Cloud. Tuy nhiên, Next Generation Firewall không được thiết kế để xử lý trong môi trường Container Platform.

Trong Container Platform, các pod ứng dụng chạy trên các node (có thể là server vật lý hoặc máy ảo) sử dụng Network Overlay để giao tiếp với nhau. Do đó luồng dữ liệu từ bên trong môi trường Container Platform ra bên ngoài đều mang địa chỉ IP của node làm nguồn. Do đó, các Firewall nằm bên ngoài cụm Container Platform không biết được luồng dữ liệu thực sự xuất phát từ đâu.



Để đảm bảo an ninh an toàn trong môi trường Container Platform, cần phải kiểm soát được luồng dữ liệu nằm bên trong môi trường Container Platform. Vì lý do đó, Firewall phải đặt bên trong cụm Container Platform để đạt hiệu quả tối đa.

1. **CN-Series Container Firewall**

Palo Alto Networks CN-Series Container Firewall là sản phẩm Next-gen Firewall được xây dựng và phát triển với mục đích bảo mật trong môi trường Kubernetes, chống thất thoát dữ liệu và các phương thức tấn công hiện đại. CN-Series Firewall cho phép đội vận hành và bảo mật hệ thống có thể kiểm soát luồng dữ liệu tới tầng ứng dụng (layer 7) trong mô hình OSI. Với khả năng mở rộng linh hoạt, CN-Series Firewall cũng đáp ứng được quy trình phát triển/tích hợp ứng dụng của các nhóm Dev/Ops hiện đại.

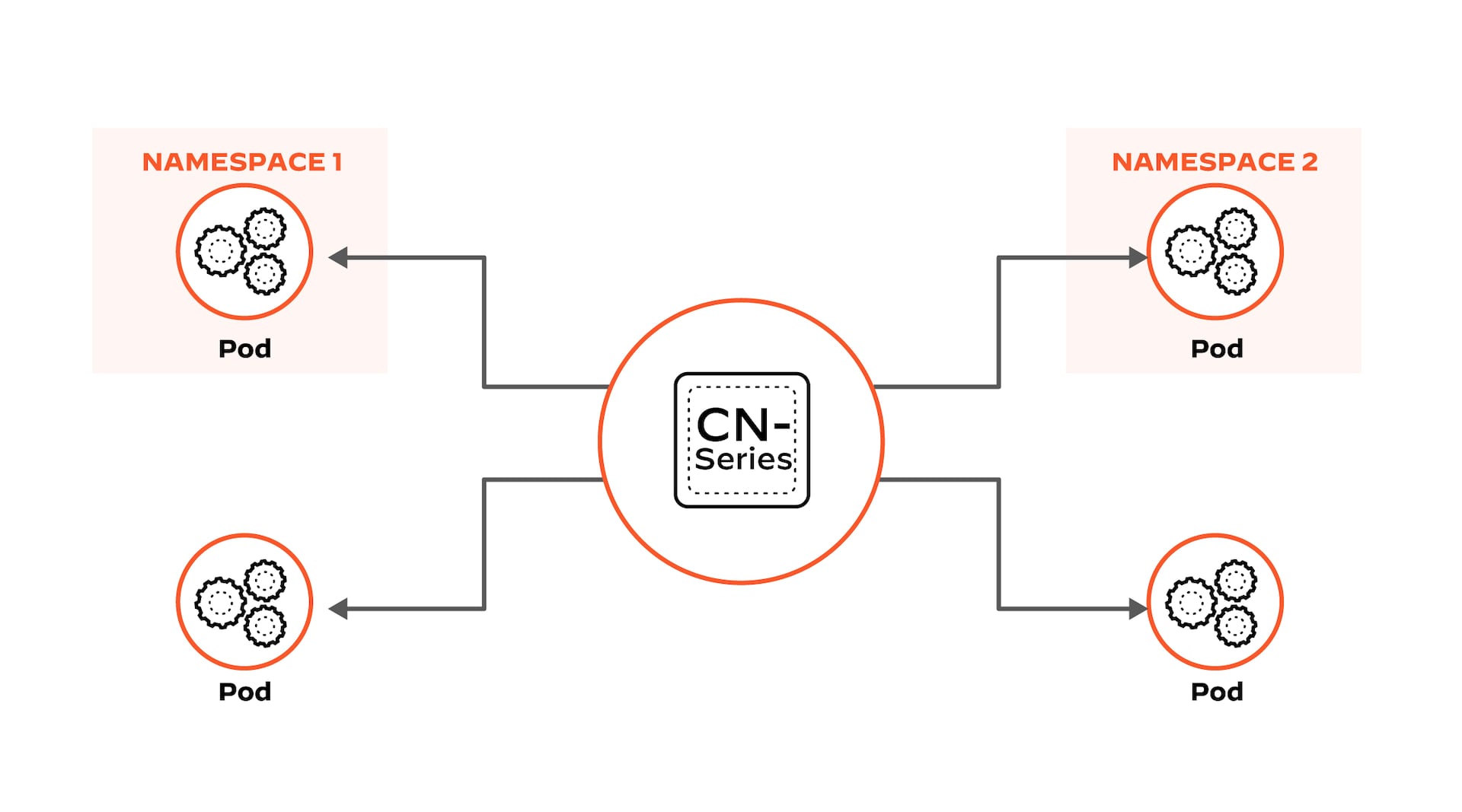
CN-Series Firewall sử dụng công nghệ App-ID để xác định cả ứng dụng và nội dung trong kết nối, cung cấp khả năng kiểm tra nội dung đầy đủ thay vì kiểm soát truy cập layer 3/layer 4 như tường lửa truyền thống. Bên cạnh thế mạnh là khả năng kiểm soát truy cập tới lớp ứng dụng (layer 7), CN-Series cũng cung cấp các tính năng ngăn chặn và phân tích các mối đe dọa chuyên sâu thông qua dịch vụ dựa trên nền tảng đám mây (cloud-based) như: Threat Prevention, URL Filtering, WildFire, DNS Security.

Palo Alto CN-Series có thể được triển khai dễ dàng bằng các công cụ thân thiện như Helm charts và Terraform.

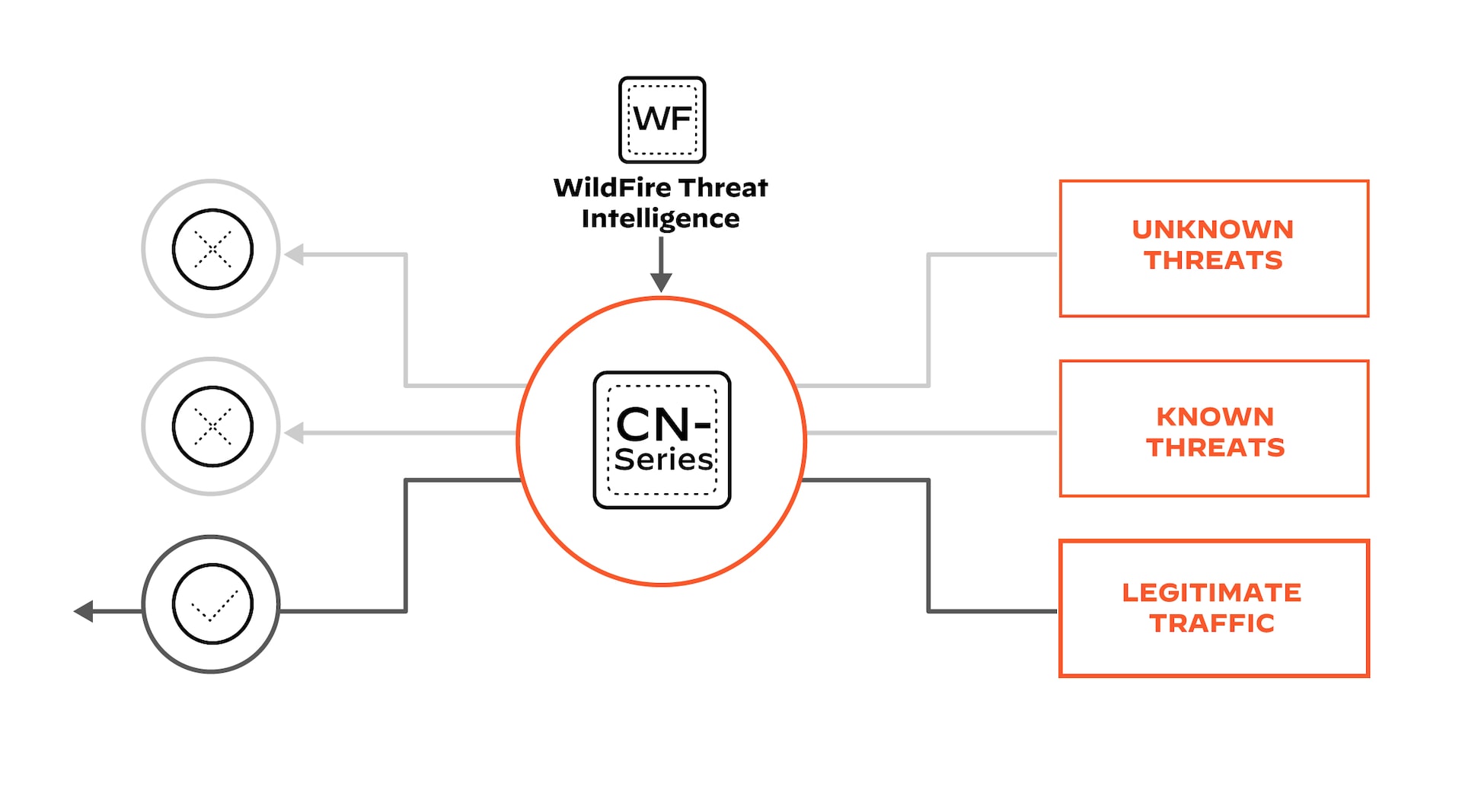
CN-Series được quản lý tập trung thông qua giải pháp quản trị tập trung Panorama giúp đội vận hành và bảo mật hệ thống có cái nhìn tổng quan về hiện trạng an ninh an toàn của tổ chức.

1. **TÍNH NĂNG CHÍNH**
2. **KIỂM SOÁT LUỒNG TRUY CẬP VÀ NGĂN CHẶN CÁC MỐI ĐE DỌA**

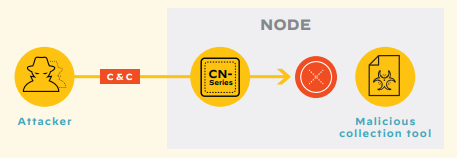
Kiểm soát luồng truy cập: Có khả năng hiển thị traffic của ứng dụng trong môi trường Kubernetes trên giao diện quản trị tập trung panorama, từ đó thiết lập các policy kiểm soát luồng ứng dụng đảm bảo tuân thủ mô hình bảo mật Zero Trust.



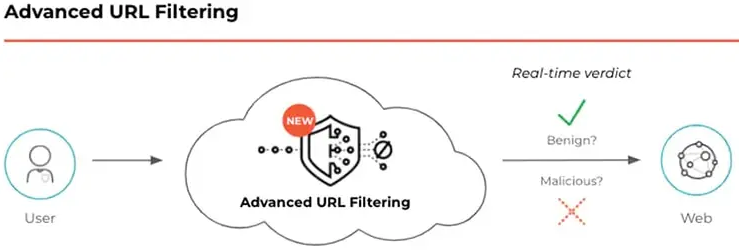
Ngăn chặn các cuộc tấn công và phân tích các mối đe dọa trong môi trường giả lập sandbox: Tính năng Threat Prevention và WildFire cho phép Firewall CN-Series ngăn chặn các mối đe dọa, các file độc hại, các cuộc tấn công đã biết và chưa biết.



Chống thất thoát dữ liệu và chặn/lọc thông qua URL: Palo Alto CN-Series với khả năng giải mã luồng dữ liệu SSL cho phép ngăn chặn rò rỉ các thông tin nhạy cảm của tổ chức ra bên ngoài.



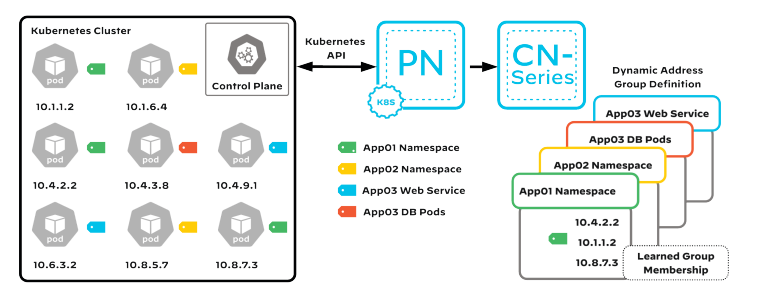
Kết hợp với tính năng URL Filtering sử dụng công nghệ Machine Learning để phân loại URL ngăn không cho người sử dụng truy cập các website độc hại được xây dựng để phát tán malware và lấy cắp thông tin.



1. **KHẢ NĂNG CẤU HÌNH VÀ MỞ RỘNG LINH HOẠT**

Tự động mở rộng với Kubernetes: CN-Series Firewall tận dụng khả năng tự động mở rộng của Kubernetes mang tới khả năng bảo vệ toàn diện ngay cả trong môi trường có sự thay đổi liên tục.

Thiết lập policy dựa theo tag-based: CN-Series Firewall cho phép sử dụng nhóm địa chỉ động trong đó các member được chọn dựa trên tag-based, tag-based này ứng với tên của các object thuộc Kubernetes như pod name, service name. Khi sử dụng tag-based, người quản trị không cần quan tâm địa chỉ IP của pod/service là gì (trong K8S/OCP mỗi khi pod restart sẽ nhận IP address mới), mà chỉ cần dựa vào tên của pod/sevice.



Thân thiện trong triển khai và cấu hình: CN-Series Firewall được triển khai thông qua YAML file nên có thể nhanh chóng tính hợp với hạ tầng hệ thống sử dụng container.

Cộng đồng hỗ trợ Kubenetes Helm chart: Với cộng đồng các nhà phát triển sử dụng công cụ Helm để quản lý ứng dụng Kubernetes, Palo Alto đã xây dựng một repository giúp cho việc triển khai và quản lý firewall được dễ dàng và thuận tiện hơn.

1. **TRIỂN KHAI VÀ TÍCH HỢP DỄ DÀNG**

Đa dạng mô hình triển khai: Có thể triển khai CN-Series Firewall theo mô hình phân tán hoặc cluster tùy theo môi trường triển khai và khả năng tài chính.

Tích hợp dễ dàng: CN-Series hỗ trợ nhiều Container Network Interface (CNI) plugins để phù hợp với nhiểu mô hình triển khai Kubernetes khác nhau.

1. **HỖ TRỢ TRIỂN KHAI TRÊN CẢ MÔI TRƯỜNG ON-PREMISES VÀ CLOUD**

Môi trường Public Cloud: CN-Series Firewall có thể được triển khai trên môi trường container như GKE, AKS, Amazon EKS và Red Hat OpenShift.

Môi trường On-premises: CN-Series Firewall có thể được triển khai trên môi trường Kubernetes tại hạ tầng nội bộ của tổ chức.

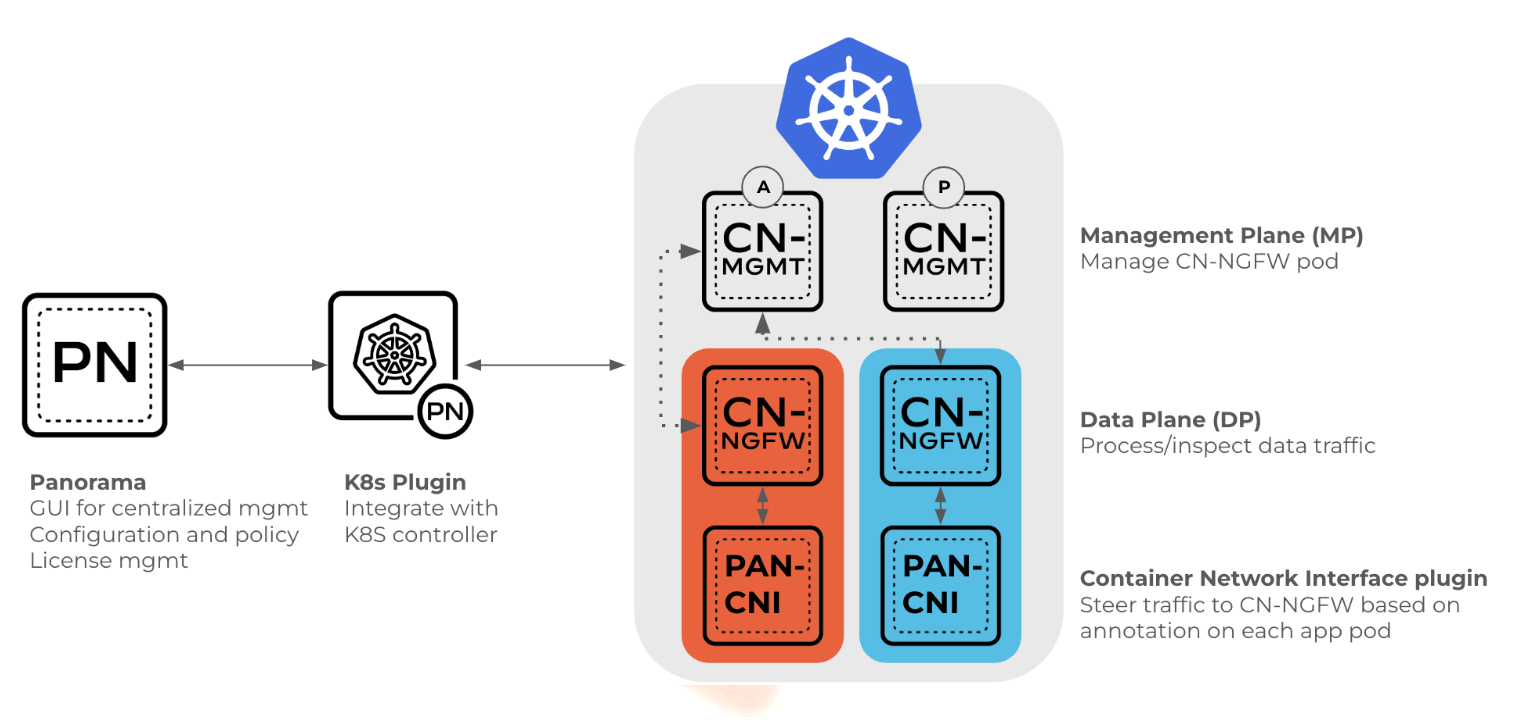
1. **QUẢN TRỊ TẬP TRUNG**

Quản trị tập trung: Palo Alto cung cấp giải pháp Panorama để quản trị tập trung các CN-Series Firewall.

Hỗ trợ đa dạng Plugin cho các môi trường khác nhau: Giải pháp Panorama có các plugin riêng cho các môi trường GKE, AKS, Amazon EKS và OpenShift cho phép tích hợp và quản trị các CN-Series một cách dễ dàng.

Quản trị log tập trung: Toàn bộ log của hệ thống CN-Series Firewall sẽ được tập trung về Panorama giúp quản trị viên có thể dễ dàng theo dõi và vận hành.

1. **THÀNH PHẦN GIẢI PHÁP CN-SERIES CONTAINER FIREWALL**

****

**Panorama:** hệ thống quản trị tập trung, quản lý cấu hình, chính sách (policy), tạo các báo cáo dựa trên template có sẵn hoặc người dùng có thể tự định nghĩa. Có thể triển khai dưới dạng thiết bị vật lý hoặc ảo hóa. Panorama quản trị, cấu hình CN-Series Firewall trong cụm OCP thông qua K8S plugin. Panorama phải được kết nối với CN-MGMT để có thể cấp license cho CN-NGFW, đẩy các cấu hình và security policy.

**K8S Plugin:** Quản lý license CN-Series Firewall. K8S Plugin kết nối đến cụm OCP thông qua service account. Thu thập các thông tin về namespaces, services, pods, lables trong cụm OCP sau đó tạo tag-based một cách tự động hoặc người quản trị có thể tự tạo thủ công cho các object liên quan. Các tag này sau đó được sử dụng để tạo các chính sách bảo mật (security policies) cho cụm OCP.

**CN-MGMT:** Đóng vai trò điều khiển (Management Plane). CN-MGMT pod được triển khai theo cặp sử dụng StatefulSet Configure để đảm bảo Volume persistence và cấu hình được đồng bộ.

Đồng thời cung cấp khả năng dự phòng khi một pod CN-MGMT bị lỗi hoặc khởi động lại. Tại một thời điểm chỉ có duy nhất một pod CN-MGMT quản lý các pods CN-NGFW (Data Plane). CN-MGMT đầu tiên sẽ kết nối với CN-MGMT thứ hai thông qua Kubernetes API. Trong quá trình triển khai CN-MGMT, một pod bổ sung CN-MGMT-INIT được sử dụng để sinh certificate xác thực. Pod CN-MGMT-INIT sẽ bị xóa khi quá trình tạo certificate xác thực hoàn thành.

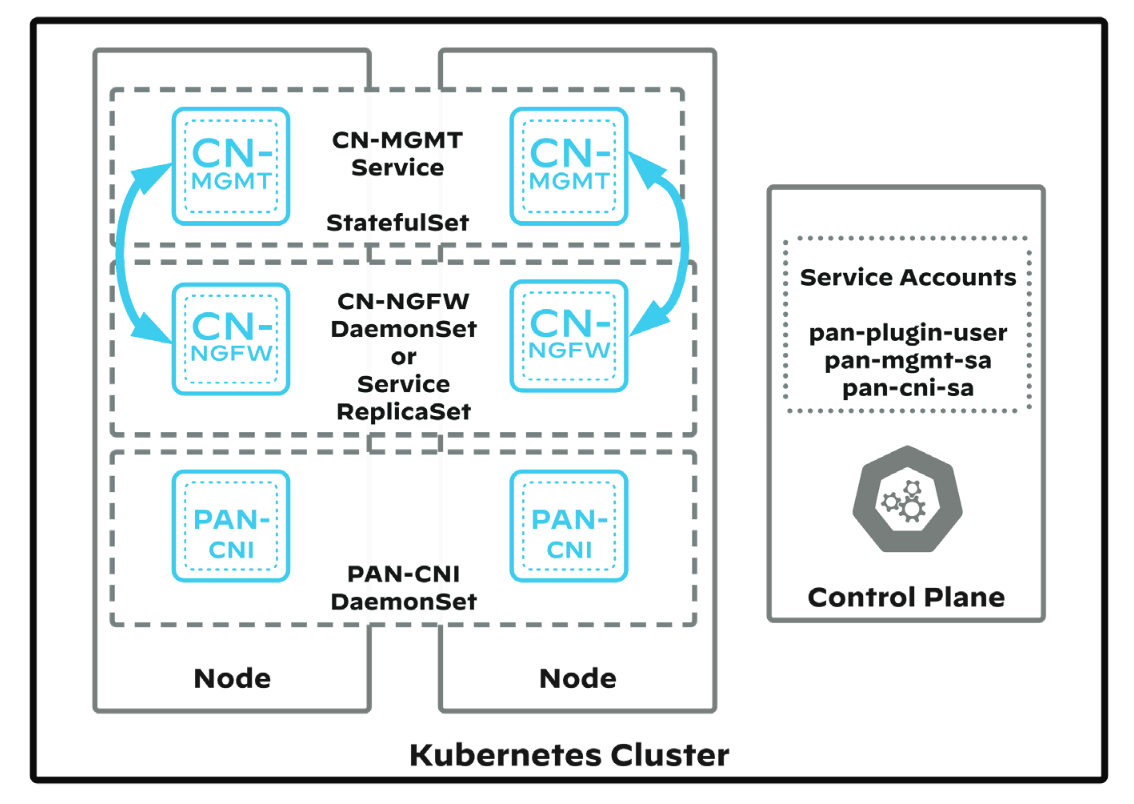
**CN-NGFW:** CN-NGFW (Data Plane) có thể được triển khai theo dạng DaemonSet hoặc Kubernetes Service (ReplicaSet deployment). Đóng vai trò tường lửa trực tiếp xử lý luồng dữ liệu giữa các pod/service trong cụm OCP. CN-NGFW nhận lệnh điều khiển trực tiếp từ CN-MGMT. Mỗi một CN-NGFW pod sẽ kết nối tới một CN-MGMT pod qua tunnel IPSec xác thực bằng certificate. Tất cả các kết nối giữa CN-NGFW và CN-MGMT sẽ thông qua tunnel này.

**CN-CNI:** Triển khai dưới dạng DaemonSet, nhúng PAN-CNI plugin vào chuỗi CNI plugin trên mỗi node trong cluster. Plugin CN-CNI thực hiện lái luồng dữ liệu từ các pod ứng dụng qua CN-NGFW để kiểm soát, chặn lọc.

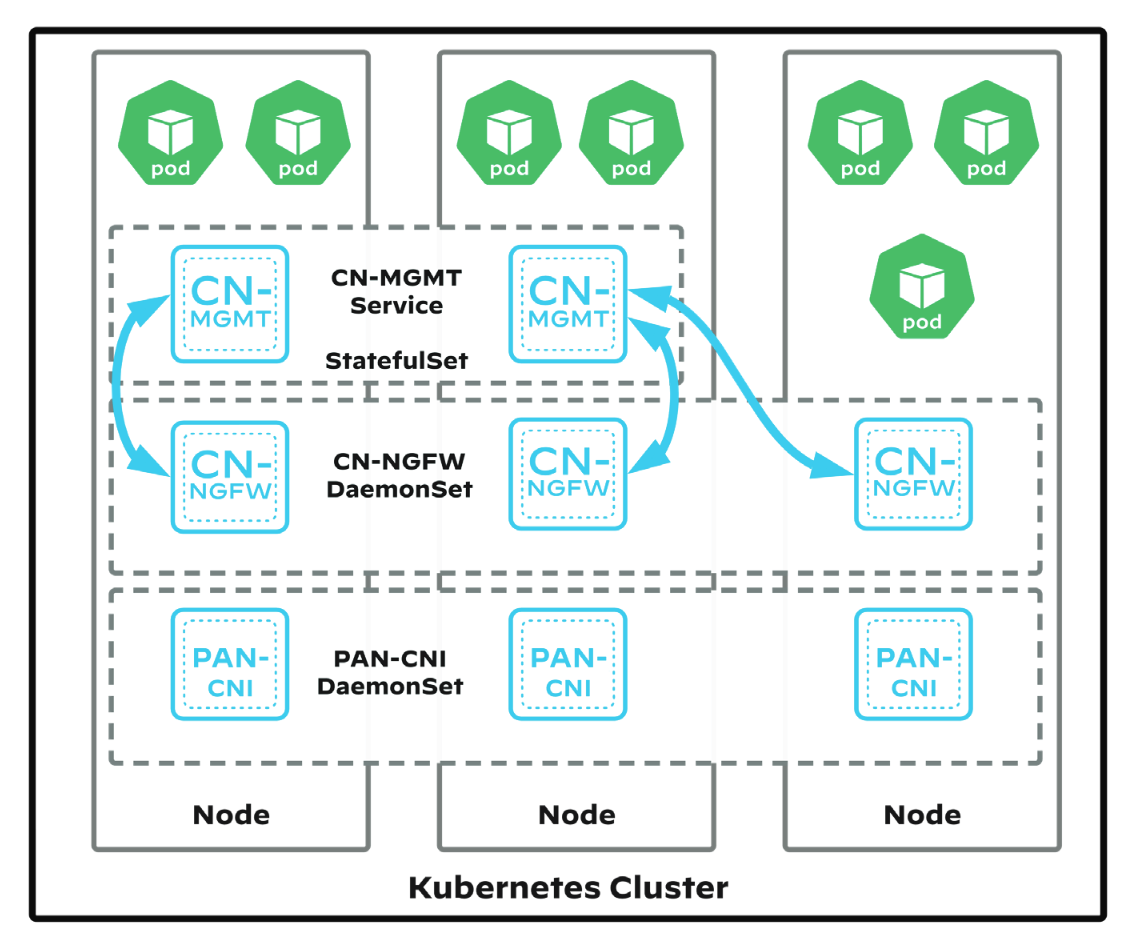
Plugin CN-CNI sẽ đọc annotation trên mỗi pod ứng dụng (khi pod ứng dụng khởi động) để xác định xem có bật tính năng bảo mật hay điều hướng inbound, outbound luồng dữ liệu từ pod ứng dụng tới CN-NGFW pod hay không.

Ngoài các thành phần chính trên, CN-Series Firewall yêu cầu tạo ra 03 tài khoản Kubernetes service account với quyền tối thiểu cho phép giao tiếp với các tài nguyên của OCP cluster. Cụ thể như sau:

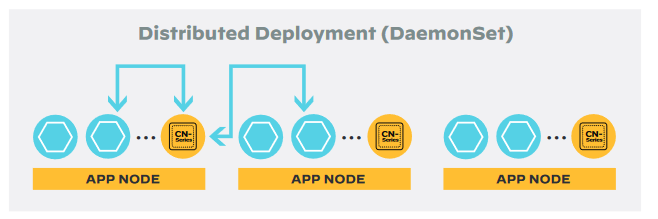
* pan-plugin-user: xác thực giữa plugin trên Panorama với Kubernetes cluster để lấy thông tin metadata của pod.
* pan-mgmt-sa: xác thực giữa các CN-MGMT pod đảm bảo cơ chế dự phòng của CN-MGMT hoạt động ổn định.
* pan-cni-sa: xác thực giữa CN-MGMT pod và CN-NGFW pod



1. **MÔ HÌNH TRIỂN KHAI**
2. **MÔ HÌNH TRIỂN KHAI PHÂN TÁN (DEAMONSET)**

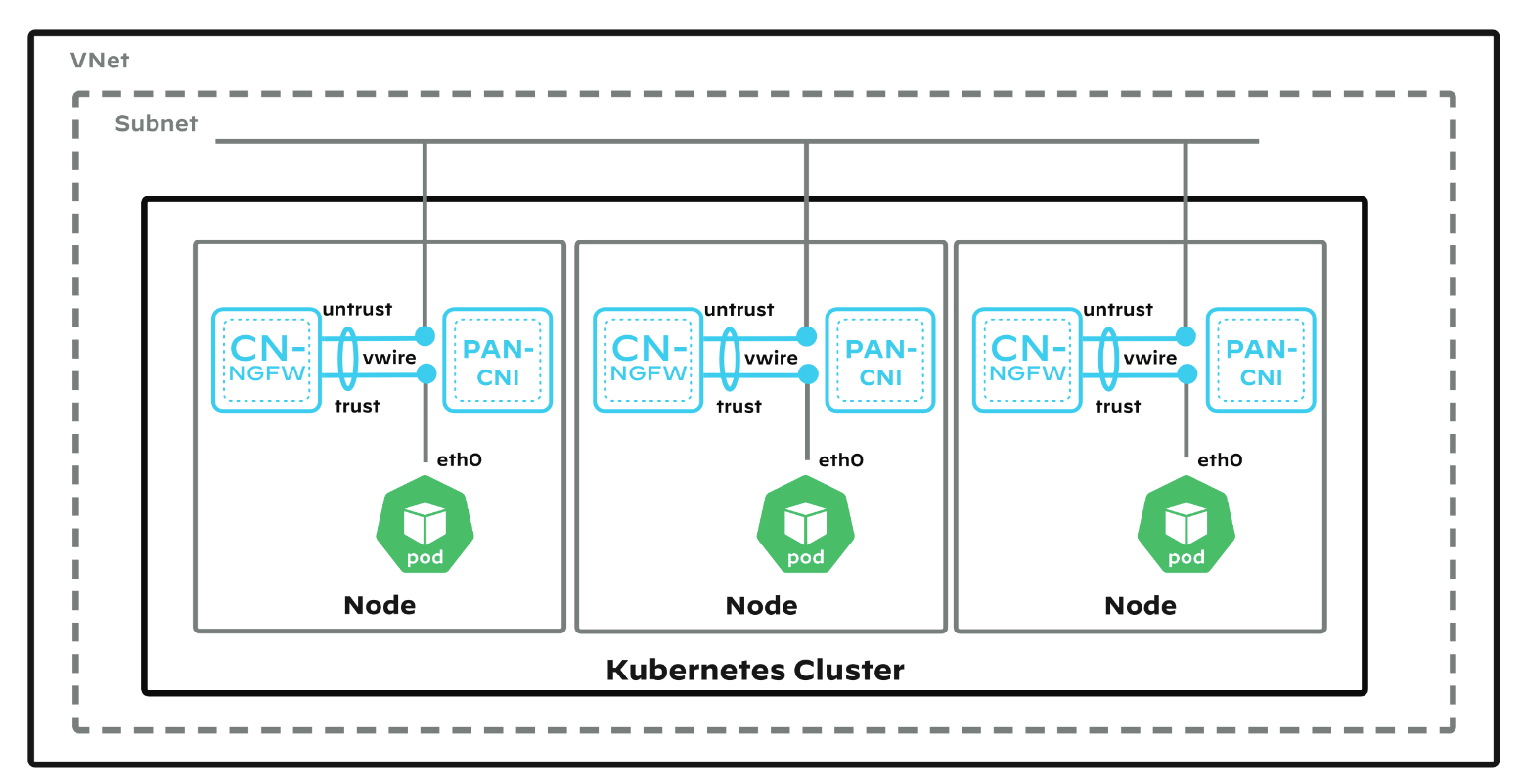


Trong mô hình triển khai này, một CN-NGFW pod sẽ nằm trên mỗi node trong OCP cluster. Với mô hình triển khai này với mỗi node được triển khai thêm sẽ tự động tạo thêm một CN-NGFW pod. Pod CN-NGFW được triển khai cùng với pod ứng dụng, cũng như bất kỳ thành phần nào của CN-Series trên cùng một node nên yêu cầu tài nguyên (CPU, RAM, DISK) trên mỗi node phải đủ lớn.



Mô hình triển khai DaemonSet sử dụng cấu hình virtual-wire để kiểm soát luồng dữ liệu đến và đi từ pod ứng dụng, không thay đổi mô hình mạng vật lý hay logic. Plugin PAN-CNI sẽ chịu trách nhiệm điều hướng traffic từ pod ứng dụng tới CN-NGFW pod thông qua việc đọc annotation được gán cho pod ứng dụng.

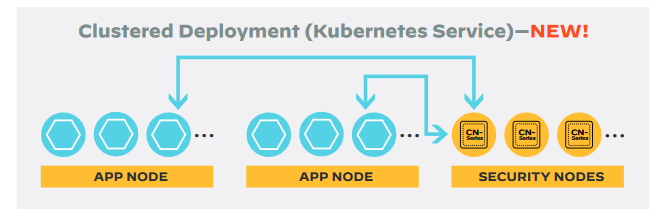
Cho phép tạo ra 02 vùng mạng “trust” và “untrust” cho mỗi pod ứng dụng. Mỗi CN-NGFW pod có thể kiểm soát truy cập được tối đa 30 pod ứng dụng (một pod ứng dụng cho mỗi một virtual-wire) trong một node. Plugin PAN-CNI chỉ kiểm soát luồng dữ liệu trên interface mặc định “eth0” của pod ứng dụng.



Ưu điểm của mô hình triển khai này là luồng dữ liệu trao đổi có độ trễ thấp, do pod ứng dụng nằm chung với CN-NGFW pod trên cùng một node.

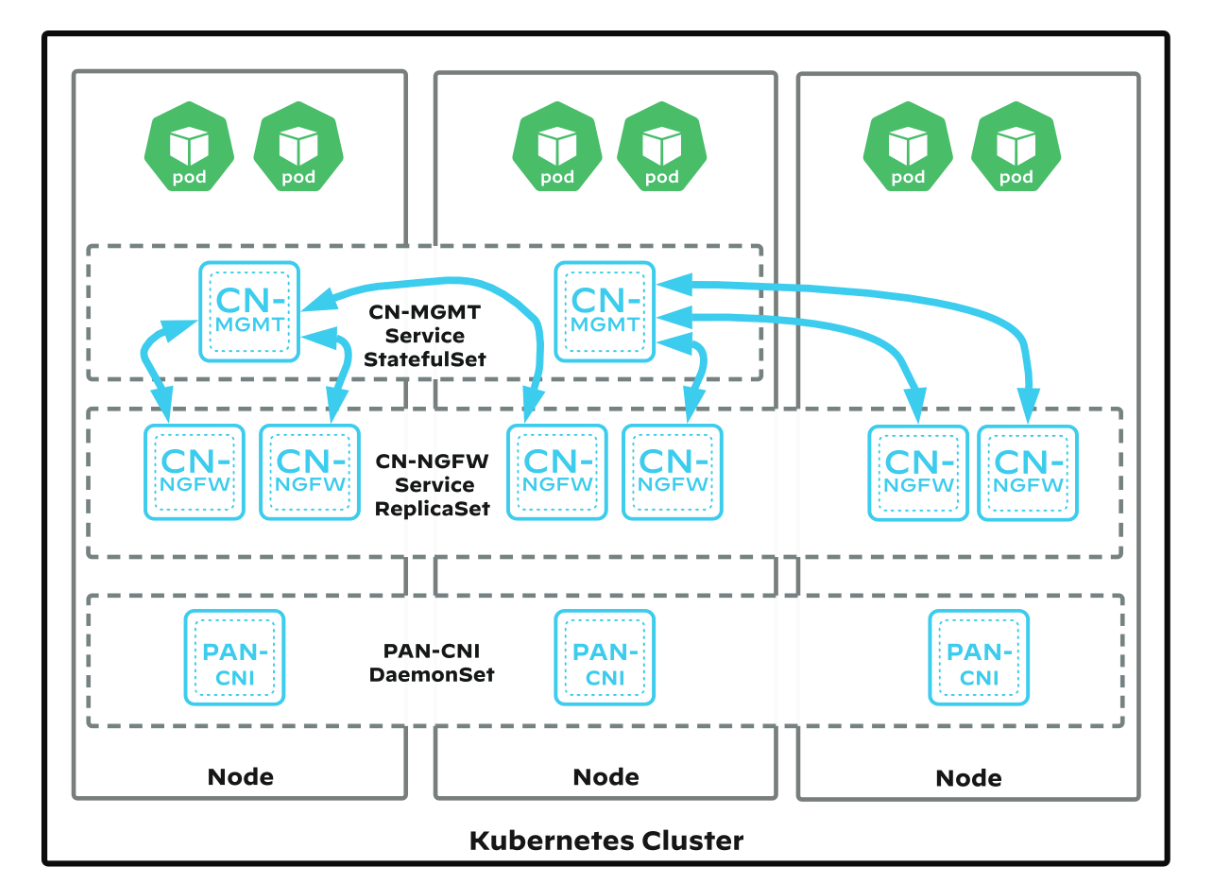
Nhược điểm của mô hình này là giới hạn mỗi node chỉ có thể triển khai tối đa 30 pod ứng dụng.

1. **MÔ HÌNH TRIỂN KHAI CLUSTER (KUBERNETES SERVICE)**



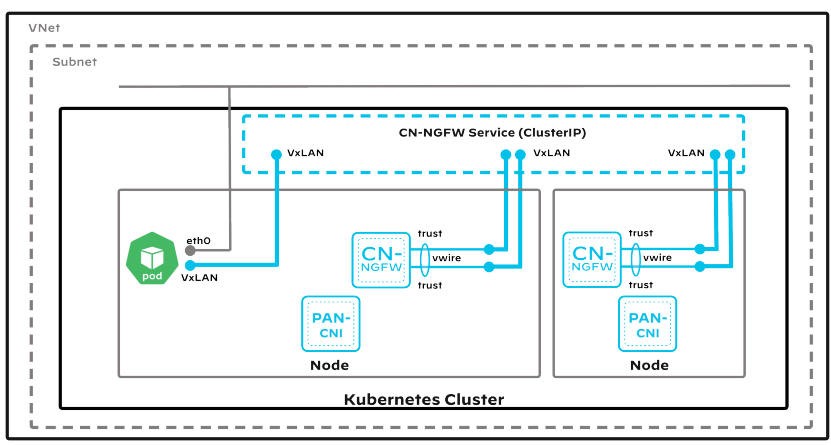
Trong mô hình này, CN-NGFW sẽ được triển khai sử dụng ReplicaSet deployment. Cách thức triển khai này cho phép tạo và quản lý pod CN-MGMT và pod CN-NGFW ở bất kì node nào trong cluster hoặc trong một nhóm các node gọi là node pool. Điều này giúp tách biệt được tài nguyên sử dụng cho ứng dụng và tài nguyên sử dụng cho CN-Series Firewall.

Quản trị viên có thể dễ dàng sizing số lượng node dựa theo số lượng CN-NGFW pod triển khai trên node hoặc tùy chỉnh tài nguyên của CN-NGFW pod sao cho phù hợp với tài nguyên của node.



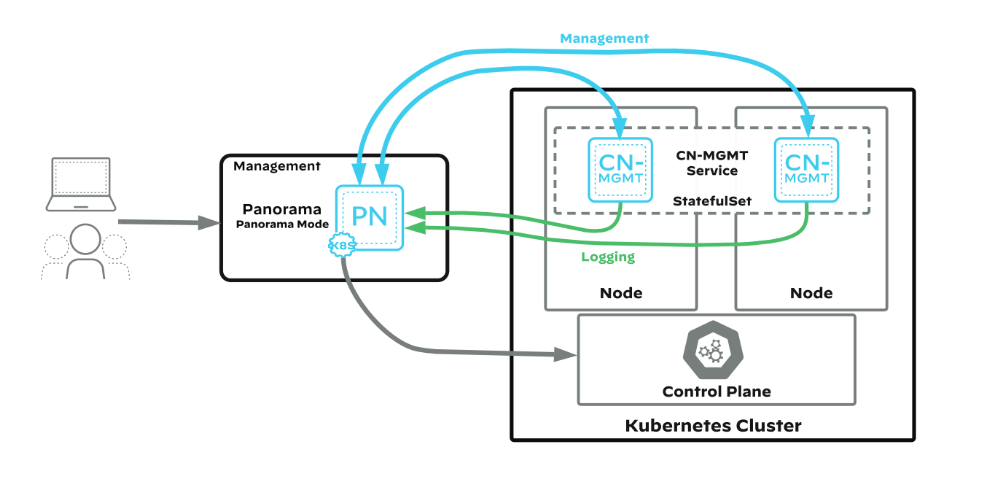
Cũng tương tự như triển khai DaemonSet, triển khai theo Kubernetes Services cũng sử dụng cấu hình virtual-wire nhưng chi sử dụng duy nhất một virtual-wire trên node mà CN-NGFW pod được deploy. CN-Series Firewall sử dụng VxLAN tunnel để điều hướng traffic từ pod ứng dụng tới pod CN-NGFW.

Plugin PAN-CNI sẽ tạo VxLAN tunnel giữa interface của pod ứng dụng tới CN-NGFW service ClusterIP. PAN-CNI plugin sẽ chỉ kiểm soát luồng dữ liệu trên interface mặc định “eth0” của pod ứng dụng. PAN-CNI plugin sẽ đọc annotation được gắn vào pod ứng dụng sau đó tạo VxLAN tunnel từ pod ứng dụng tới CN-NGFW Service, định tuyến lại luồng cho pod ứng dụng và thay đổi gateway đến VxLAN tunnel.



Trong mô hình triển khai này, số lượng pod ứng dụng mà một pod CN-NGFW có thể kiểm soát là không giới hạn và chỉ phụ thuộc vào hiệu năng và tài nguyên cấp cho CN-NGFW pod. Ngoài ra có thể mở rộng pod CN-NGFW thủ công hoặc tự động mở rộng pod theo chiều ngang trong trường hợp lưu lượng ứng dụng tăng lên.

1. QUẢN TRỊ TẬP TRUNG CẤU HÌNH POLICY FIREWALL VÀ THU THẬP LOG VỚI PANORAMA



Trong các mô hình triển khai của giải pháp Palo Alto CN-Series, Panorama được sử dụng cho 02 mục đích quản trị tập trung CN-Series Firewall và lưu trữ log tập trung.

Tính năng quản trị tập trung và thu thập log tập trung có thể được triển khai trên một Panorama appliance duy nhất hoặc có thể tách biệt riêng việc thu thập log ra một Panorama appliance khác (đặc biệt trong các hệ thống lớn).