



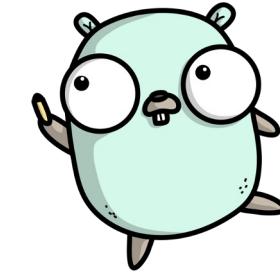
HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Giới thiệu về Kubernetes

Việt Trung Trần
(chuyển thể từ CNCF Kubernetes - Giới thiệu)

Trước khi chúng tôi bắt đầu

- Yêu cầu:
 - Minikube:
<https://github.com/kubernetes/minikube>
 - kubectl:
<https://kubernetes.io/docs/tasks/tools/install-kubectl/>
 - kho lưu trữ hướng dẫn giới thiệu k8s: <https://github.com/mrbobbytables/k8s-intro-tutorials>



Hộp đựng

Container - Tên có gì?

- Xuất thân từ ngành vận tải biển



Vận chuyển tới phần mềm

Vận chuyển container

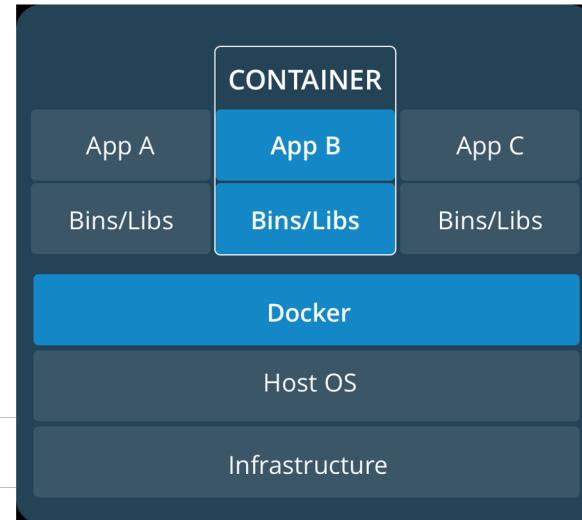
- Tính di động - có thể được sử dụng trên bất kỳ loại tàu nào được hỗ trợ
- Loại hàng hóa đa dạng có thể đóng gói bên trong
- Kích thước tiêu chuẩn - phụ kiện tiêu chuẩn trên tàu
- Nhiều container trên tàu
- Cách ly hàng hóa với nhau

Dịch sang phần mềm

- Tính di động - có thể được sử dụng trên bất kỳ hệ thống được hỗ trợ nào (hệ thống có khả năng thực thi vùng chứa môi trường)
- Phần mềm đa dạng có thể được gói gọn bên trong
- Định dạng chuẩn
- Nhiều vùng chứa cho một nút vật lý
- Tách biệt việc thực thi vùng chứa này với vùng chứa khác

Thùng chứa là gì?

- Cách kết hợp mã và phần phụ thuộc với nhau
- Có thể chạy mọi nơi
- Thực thi nhiều container trên một máy vật lý



Âm thanh quen thuộc?

- Khái niệm tương tự như máy ảo
- Đóng gói hệ điều hành và phần mềm lại với nhau để chạy trong các trường hợp riêng biệt
- Có thể chạy ở bất kỳ nơi nào mà trình ảo hóa cụ thể chạy
- Nhiều máy ảo vào một máy vật lý

VM hoạt động như thế nào?

- Hypervisor = lớp giữa VM và kernel
- Mô phỏng các cuộc gọi hệ thống
- Cho phép chạy nhiều loại hệ điều hành trên một máy (Windows trên Linux)
- Chi phí cho phần mềm ảo hóa

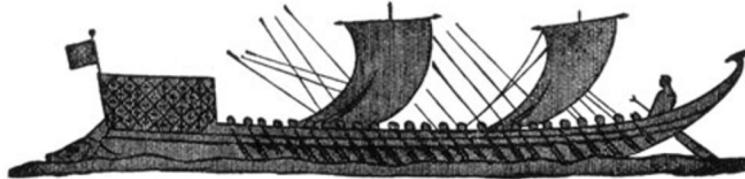
Mặt khác, container...

- Chỉ chứa các thư viện và khung công tác liên quan đến ứng dụng và ứng dụng chạy trên nhân của máy chủ
- Nhỏ hơn
- Chi phí thấp hơn
- Sự khác biệt trong phân phối và phụ thuộc hệ điều hành được trừu tượng hóa - cùng một hạt nhân

Kubernetes – Cái gì? Tại sao? Làm sao?

Kubernetes là gì?

"Kubernetes" = Tiếng Hy Lạp có nghĩa là thống đốc, người cầm lái, thuyền trưởng hệ thống điều phối container nguồn mở ban đầu được thiết kế bởi Google, được CNCF duy trì nhằm mục đích cung cấp "nền tảng để tự động hóa việc triển khai, mở rộng quy mô và vận hành các vùng chứa ứng dụng trên các cụm máy chủ"



kubernetes

Kubernetes làm gì?

- Được biết đến như hạt nhân linux của hệ thống phân tán.
- Tóm tắt phần cứng cơ bản của các nút và cung cấp giao diện thống nhất cho khối lượng công việc vừa được triển khai vừa sử dụng nhóm tài nguyên được chia sẻ.
- Hoạt động như một công cụ để giải quyết trạng thái bằng cách hội tụ trạng thái thực tế và trạng thái mong muốn của hệ thống.

Tách rời cơ sở hạ tầng và mở rộng quy mô

- Tất cả các dịch vụ trong Kubernetes đều được Cân bằng tải.
- Có thể tăng giảm quy mô linh hoạt.
- Được sử dụng cả hai để cho phép tự phục hồi và nâng cấp hoặc khôi phục ứng dụng một cách liền mạch.

Tự chữa bệnh

- Kubernetes sẽ LUÔN cố gắng điều khiển cụm về trạng thái mong muốn.
- Tôi: “Tôi muốn 3 phiên bản redis khỏe mạnh luôn chạy.”
- Kubernetes: “Được rồi, tôi sẽ đảm bảo luôn có 3 phiên bản hoạt động.”
- Kubernetes: “Ôi nhìn kìa, một người đã chết. Tôi đang cố gắng tạo ra một cái mới.”

Kubernetes THỰC SỰ có thể làm được gì?

- Tự động chia tỷ lệ khối lượng công việc
- Triển khai Xanh lam/Xanh lục
- Sa thải công việc và cronjob theo lịch trình
- Quản lý các ứng dụng có trạng thái và phi trạng thái
- Cung cấp các phương pháp khám phá dịch vụ gốc
- Dễ dàng tích hợp và hỗ trợ các ứng dụng của bên thứ 3

Quan trọng nhất...

- Sử dụng API CÙNG
trên kim loại trần và MỌI nhà cung cấp đám mây!!!



Ai “Quản lý” Kubernetes?



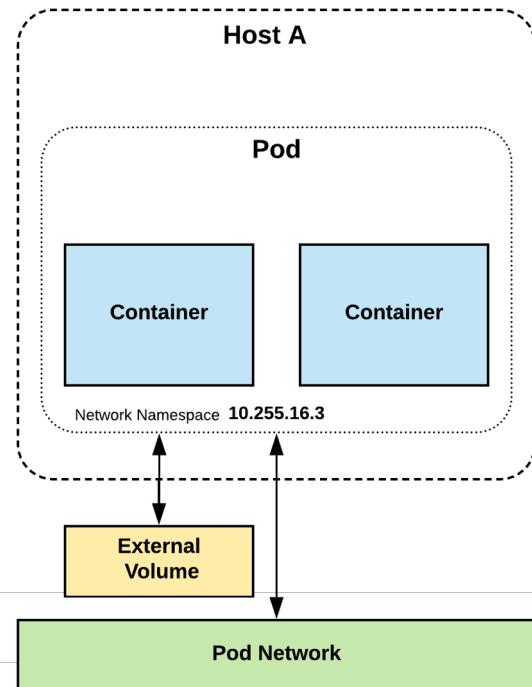
**CLOUD NATIVE
COMPUTING FOUNDATION**

- CNCF là một thực thể con của Linux Foundation và hoạt động như một nhóm quản trị trung lập với nhà cung cấp.

Một cặp vợ chồng
Ý chính...

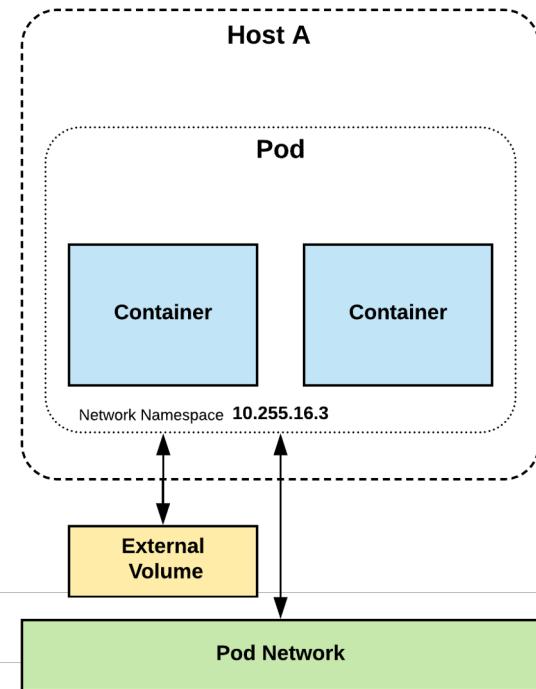
Vỏ

- Đơn vị nguyên tử hay “đơn vị công việc” nhỏ nhất của Kubernetes.
- Nhóm là một hoặc NHIỀU vùng chứa chia sẻ các khối, không gian tên mạng và là một phần của một ngữ cảnh.



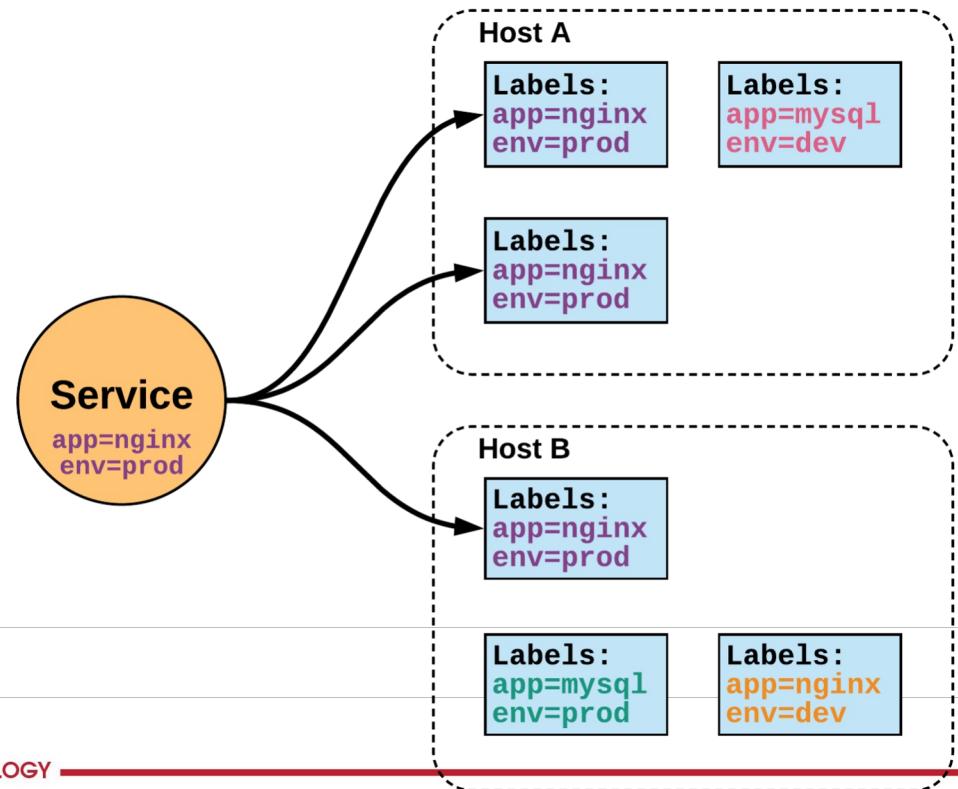
Vở

- Chúng cũng là phù du!



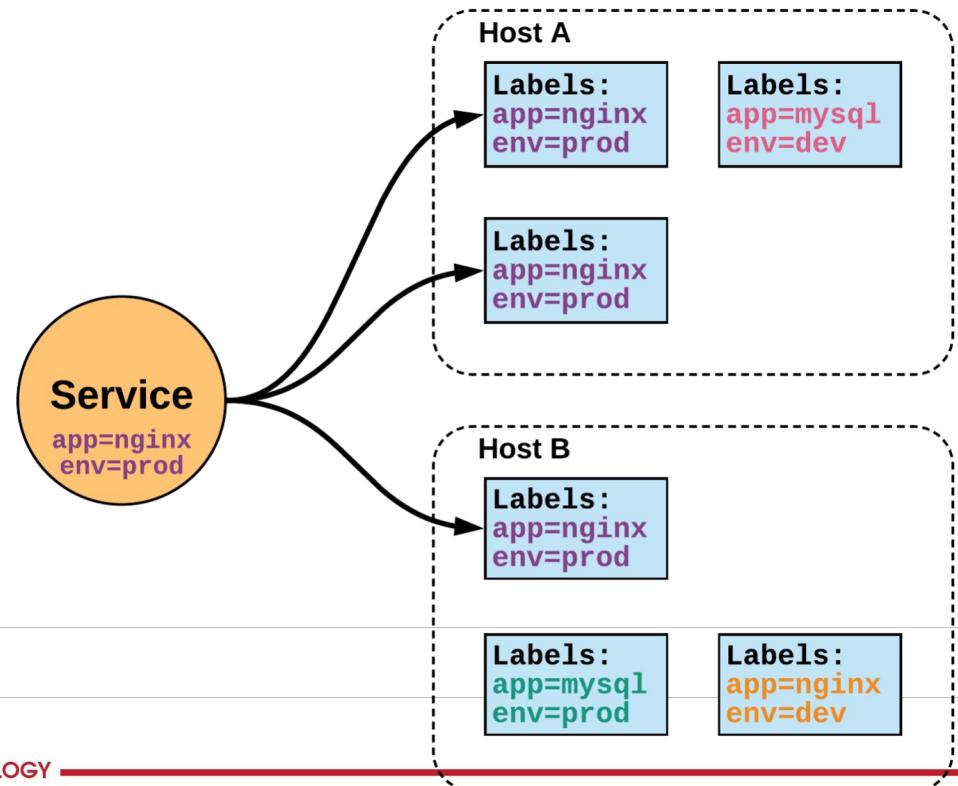
Dịch vụ

- Phương pháp thống nhất để truy cập khối lượng công việc của Pod.
- Tài nguyên bền vững
 - IP cụm tĩnh
 - tên DNS được đặt tên tĩnh

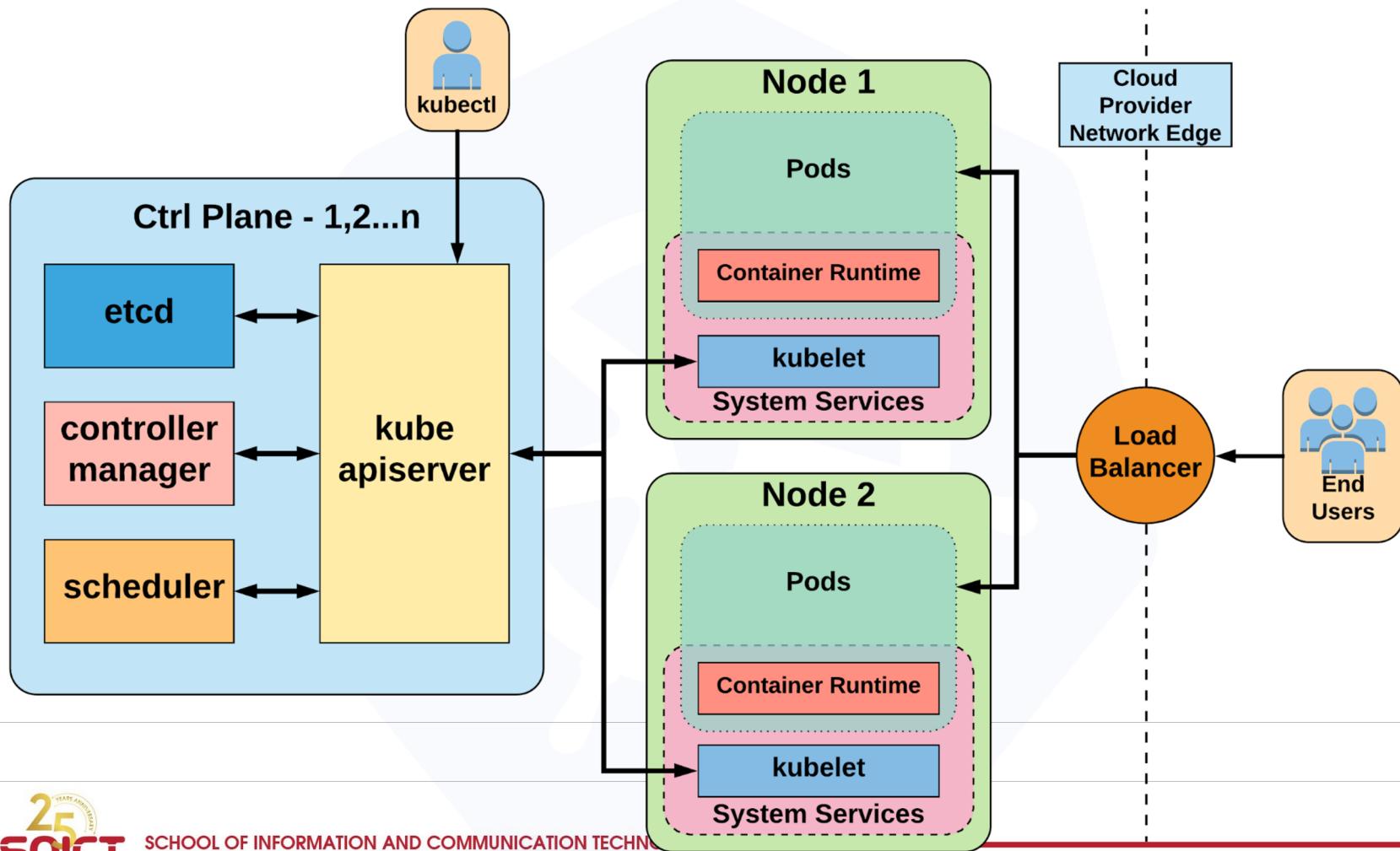


Dịch vụ

- Phương pháp thống nhất để truy cập khối lượng công việc của Pod.
- Tài nguyên bền vững
 - IP cụm tĩnh
 - tên DNS được đặt tên tĩnh
- **KHÔNG** phù du!



Tổng quan về kiến trúc

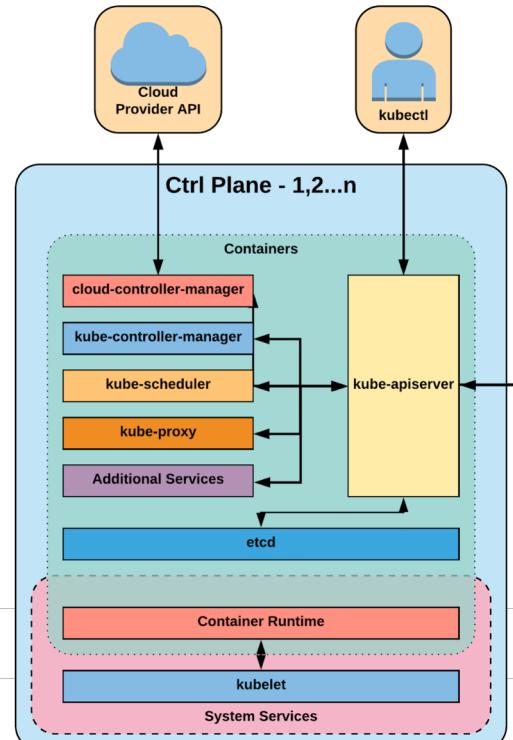


Thành phần mặt phẳng điều khiển

Tổng quan về kiến trúc

Thành phần mặt phẳng điều khiển

- máy chủ kube-apiserver
- V.V.
- trình quản lý bộ điều khiển kube
- bộ lập lịch kube



máy chủ kube

- Cung cấp giao diện REST hướng về phía trước vào mặt phẳng điều khiển kubernetes và kho dữ liệu.
- Tất cả các máy khách và ứng dụng khác đều tương tác với kubernetes một cách nghiêm ngặt thông qua Máy chủ API.
- Đóng vai trò là người gác cổng cho cụm bằng cách xử lý xác thực và ủy quyền, xác thực yêu cầu, đột biến và kiểm soát nhập học ngoài vai trò là giao diện người dùng cho kho dữ liệu hỗ trợ.

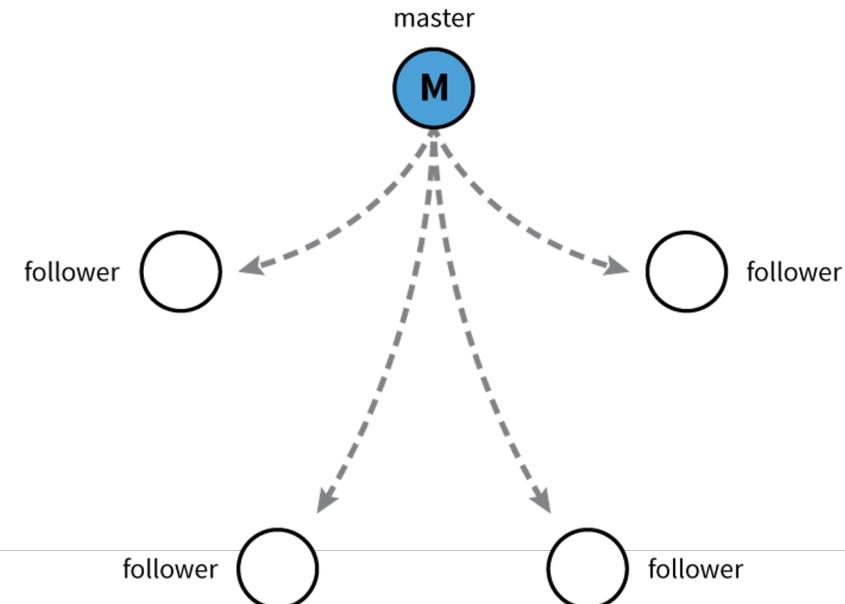
V.V.

- etcd hoạt động như kho dữ liệu cụm.
- Mục đích liên quan đến Kubernetes là cung cấp một kho lưu trữ khóa-giá trị mạnh mẽ, nhất quán và có tính sẵn sàng cao cho trạng thái cụm liên tục.
- Lưu trữ các đối tượng và thông tin cấu hình.



V.V.

- Sử dụng “Sự đồng thuận bè” giữa một số hệ thống để tạo ra một “quan điểm” nhất quán về khả năng chịu lỗi về
- <https://raft.github.io/>



[Nguồn hình ảnh](#)

quản lý bộ điều khiển kube

- Phục vụ như daemon chính quản lý tất cả các vòng điều khiển thành phần cốt lõi.
- Giám sát trạng thái cụm thông qua máy chủ apiserver và điều khiển cụm theo trạng thái mong muốn.
- Danh sách các bộ điều khiển lõi: <https://github.com/kubernetes/kubernetes/blob/master/cmd/kube-controller-manager/app/controllermanager.go#L344>

lập lịch kube

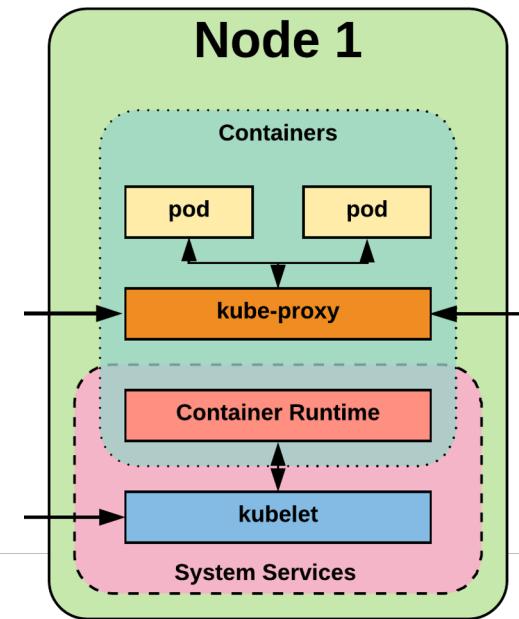
- Công cụ giàu chính sách chi tiết đánh giá các yêu cầu về khối lượng công việc và cố gắng đặt khối lượng công việc đó vào tài nguyên phù hợp.
- Bộ lập lịch mặc định sử dụng việc đóng gói thùng rác.
- Yêu cầu về khối lượng công việc có thể bao gồm: yêu cầu chung về phần cứng, mối quan hệ/chống mối quan hệ, nhãn và các yêu cầu tài nguyên tùy chỉnh khác nhau.

Thành phần nút

Tổng quan về kiến trúc

Thành phần nút

- kubelet
- proxy kube
- Công cụ thời gian chạy vùng chứa



kubelet

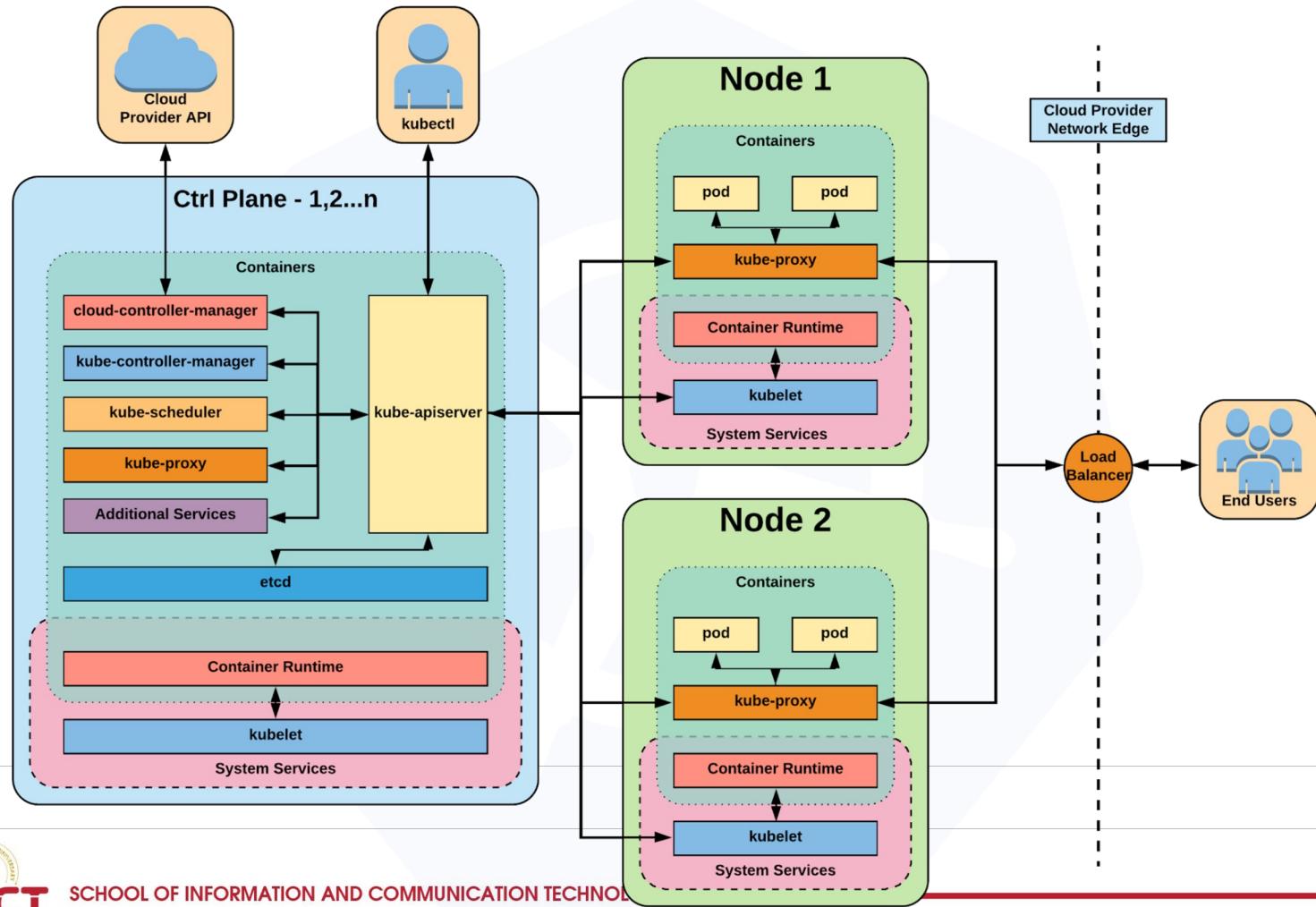
- Đóng vai trò là tác nhân nút chịu trách nhiệm quản lý vòng đời của mọi nhóm trên máy chủ của nó.
- Kubelet hiểu các bảng kê khai vùng chứa YAML có thể đọc từ nhiều nguồn:
 - đường dẫn tập tin
 - Điểm cuối HTTP
 - etcd xem hành động theo bất kỳ thay đổi nào
 - Chế độ Máy chủ HTTP chấp nhận các bảng kê khai vùng chứa qua một API đơn giản.

proxy kube

- Quản lý các quy tắc mạng trên mỗi nút.
- Thực hiện chuyển tiếp kết nối hoặc cân bằng tải cho các dịch vụ cụm Kubernetes.
- Các chế độ proxy có sẵn:
 - Không gian người dùng
 - iptables
 - ipvs (mặc định nếu được hỗ trợ)

Công cụ thời gian chạy container

- Thời gian chạy vùng chứa là một ứng dụng tương thích với CRI (Giao diện thời gian chạy vùng chứa) để thực thi và quản lý vùng chứa.
 - Container (docker)
 - Cri-o
 - Rkt
 - Kata (trước đây rõ ràng và siêu)
 - Virtlet (thời gian chạy tương thích VM CRI)



Mạng

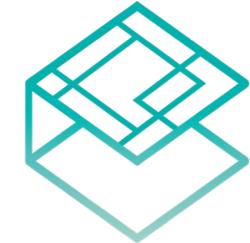
Tổng quan về kiến trúc

Mạng Kubernetes

- Mạng nhóm
 - Mạng toàn cụm được sử dụng để liên lạc giữa các nhóm được quản lý bởi plugin CNI (Giao diện mạng vùng chứa).
- Mạng lưới dịch vụ
 - Phạm vi IP ảo trên toàn cụm được quản lý bởi kube-proxy để khám phá dịch vụ.

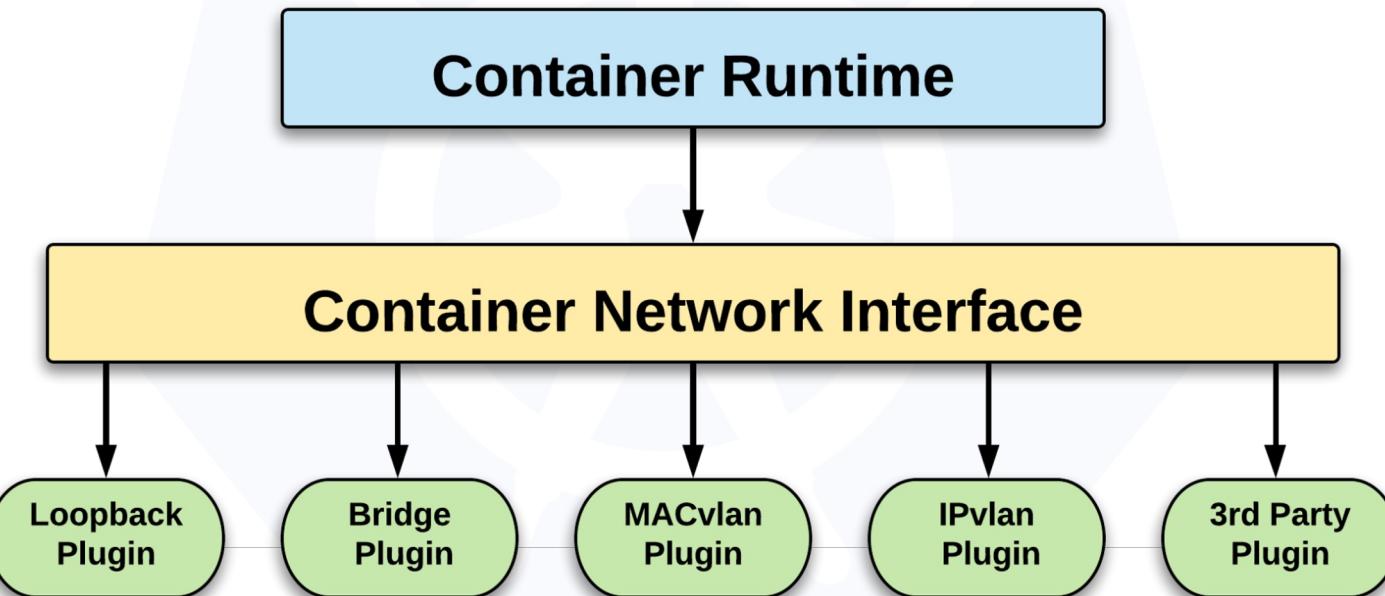
Giao diện mạng container (CNI)

- Mạng Pod trong Kubernetes được kết nối thông qua Giao diện Mạng Container (CNI).
- Chức năng như một giao diện giữa thời gian chạy vùng chứa và plugin triển khai mạng.
- Dự án CNCF
- Sử dụng Lược đồ JSON đơn giản.

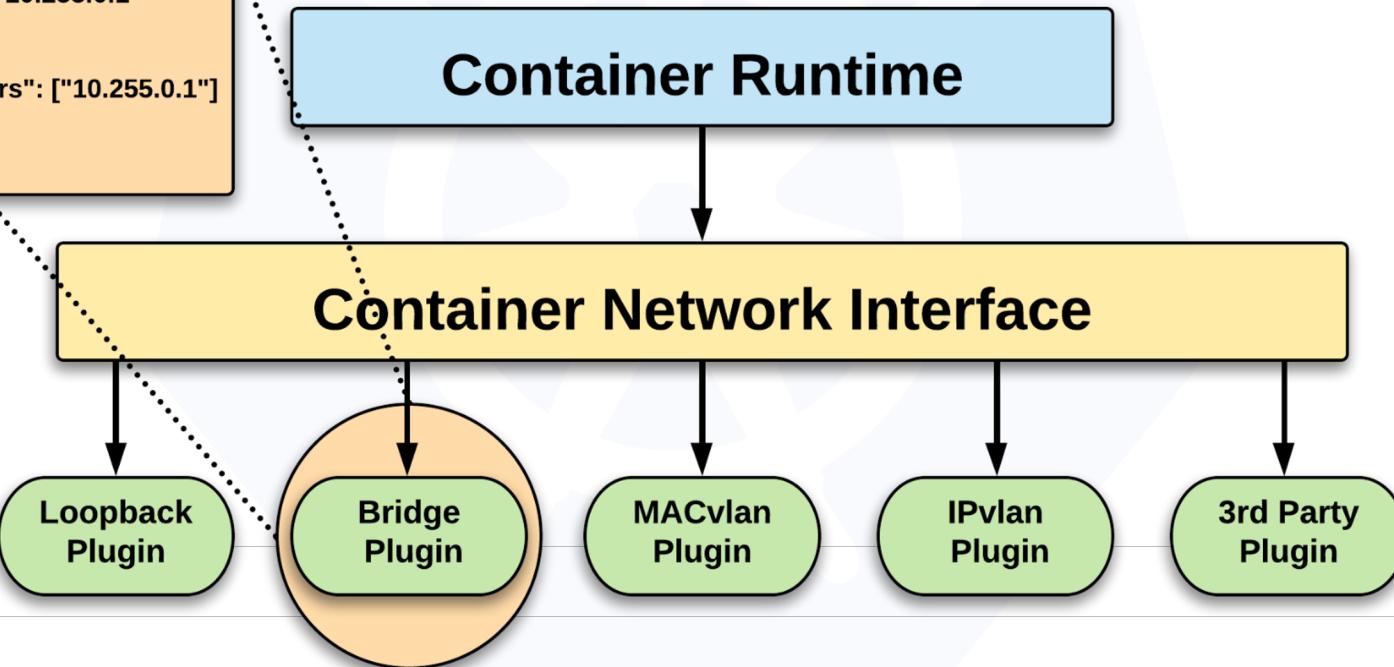


C N I

Tổng quan về CNI



```
{  
    "cniVersion": "0.3.1",  
    "name": "examplenet",  
    "type": "bridge",  
    "bridge": "cni0",  
    "ipam": {  
        "type": "host-local",  
        "subnet": "10.255.0.0/16",  
        "gateway": "10.255.0.1"  
    },  
    "dns": {  
        "nameservers": ["10.255.0.1"]  
    }  
}
```



Plugin CNI

- Amazon ECS
- Calico
- Cillium
- Tiếp tục
- vệt tương phản
- Vải nỉ



- GCE
- bộ định tuyến kube
- Đa nhân
- OpenVSwitch
- La Mã
- Dệt



Quy tắc mạng cơ bản

- Tất cả các container trong một nhóm có thể giao tiếp với nhau mà không bị cản trở.
- Tất cả các Pod có thể giao tiếp với tất cả các Pod khác mà không cần NAT.
- Tất cả các nút có thể giao tiếp với tất cả các Pod (và ngược lại) mà không cần NAT.
- IP mà Pod tự coi đó là IP mà người khác coi đó là IP.

Nguyên tắc cơ bản được áp dụng

- Từ container tới container
 - Các vùng chứa trong một nhóm tồn tại trong cùng một không gian tên mạng và chia sẻ một IP.
 - Cho phép giao tiếp nội bộ qua localhost.
- Pod-to-Pod
 - Phân bổ IP duy nhất của cụm trong suốt vòng đời của nó.
 - Bản thân vỏ quả là phù du.

Nguyên tắc cơ bản được áp dụng

- Pod-to-Service
 - được quản lý bởi kube-proxy và được cấp IP duy nhất của cụm liên tục
 - tồn tại ngoài vòng đời của Pod.
- Từ bên ngoài đến dịch vụ
 - Được xử lý bởi kube-proxy.
 - Hợp tác với nhà cung cấp đám mây hoặc tổ chức bên ngoài khác (bộ cân bằng tải).

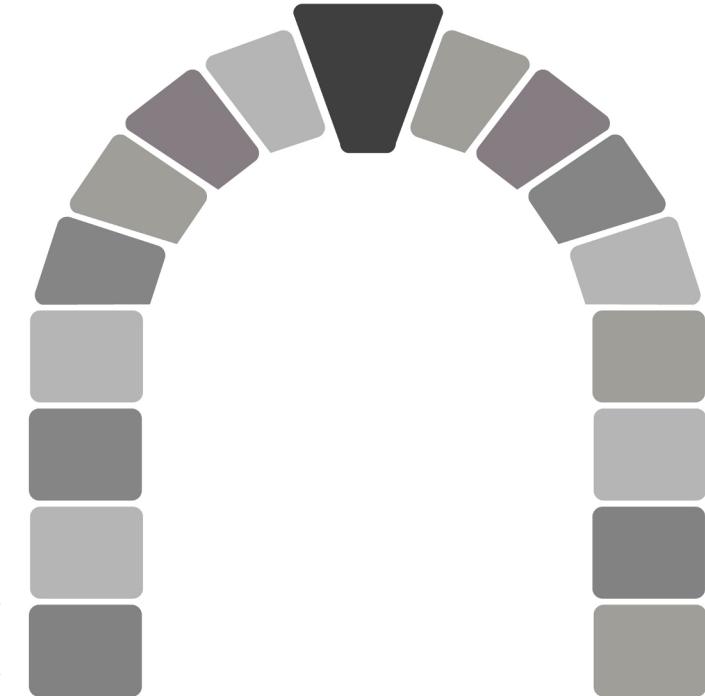
Khái niệm và tài nguyên

API và mô hình đối tượng

Khái niệm và tài nguyên

Tổng quan về API

- API REST là nền tảng thực sự của Kubernetes.
- Mọi thứ trong Kubernetes đều là Đối tượng API.



[Ng nguồn hình ảnh](#)

Nhóm API

- Được thiết kế để làm cho việc sử dụng cực kỳ đơn giản **vừa hiểu vừa mở rộng**.
- Nhóm API là một đường dẫn tương thích với REST, đóng vai trò là bộ mô tả kiểu cho đối tượng Kubernetes.
- Được tham chiếu bên trong một đối tượng dưới dạng apiVersion và loại.

Định dạng:

/apis/<nhóm>/<phiên bản>/<tài nguyên>

Ví dụ:

/apis/**Ứng dụng**/v1/trieu khai /
apis/**lô hàng**/v1beta1/cronjob