



TÀI LIỆU HƯỚNG DẪN SỬ DỤNG DỊCH VỤ

VIETTEL OPEN KUBERNETES SERVICE

DÀNH CHO KHÁCH HÀNG

MỤC LỤC

1. Tổng quan dịch vụ.....	6
1.1 Một số khái niệm cơ bản về Kubernetes	6
1.1.1 Kubernetes & kiến trúc của 1 cụm tài nguyên Kubernetes	6
1.1.2 Control Plane (Master Node)	7
1.1.3 Node (Worker Node)	8
1.1.4 Kubectl	8
1.1.5 Pod.....	8
1.1.6 Image.....	9
1.1.7 Deployment	9
1.1.8 ReplicaSet	9
1.1.9 Service	9
1.1.10 Labels	9
1.1.11 Taints.....	10
1.1.12 Worker Autoscale.....	11
1.2 Giới thiệu dịch vụ Viettel Open Kubernetes Service (VOKS).....	12
1.2.1 VOKS là gì?.....	12
1.2.2 VOKS có những tính năng gì?	12
1.2.3 VOKS tính tiền như thế nào?	14
2. Cluster	15
2.1 Cluster là gì?.....	15
2.2 Cluster của dịch vụ vOKS	15
2.3 Khởi tạo cluster.....	17
2.3.1 Khởi tạo 01 cluster dùng thử.....	17
2.3.2 Khởi tạo 01 cluster Saving Plan	19
2.4 Quản trị cluster.....	23

2.5 Nâng cấp phiên bản của cụm Kubernetes (Upgrade Kubernetes cluster version)	24
2.6 Best practice khi nâng cấp phiên bản cho cụm Kubernetes	25
2.6.1. Chuẩn bị trước khi nâng cấp:	25
2.6.2. Thực hiện nâng cấp.....	26
2.6.3. Sau khi nâng cấp	26
2.7 Xóa cluster	26
3. Quản lý truy cập.....	27
3.1 Sử dụng tính năng Identity Access Management (IAM)	27
3.2 Các chính sách (Policy) phân quyền của dịch vụ vOKS.....	28
4. Quản lý tài nguyên	28
4.1 Tạo Node group	29
4.1.1 Tạo thêm Node group cho cluster Saving Plan	29
4.1.2 Tạo thêm Node group & Node cho gói dịch vụ PAYG	30
4.2 Xem thông tin Node group & các node bên trong	33
4.3 Sửa thông tin Node group.....	34
4.3.1 Sửa tên Node group.....	34
4.3.2 Sửa cấu hình Worker Autoscale theo Node Group	35
4.3.3 Sửa cấu hình auto-repair theo Node Group	36
4.4 Xóa Node group.....	37
5. Lưu trữ dữ liệu ứng dụng	38
5.1 Giới thiệu Volume & NFS Node	38
5.2 Các dạng kết nối lưu trữ trên NFS node	40
5.3 Kiến trúc NFS node	40
5.4 Thêm dung lượng NFS Node	41

5.4.1 Thêm dung lượng NFS Node cho cluster Saving Plan.....	41
5.4.2 Thêm dung lượng NFS Node cho cluster PAYG.....	42
5.5 Hướng dẫn cấu hình sử dụng NFS node	43
6. Thiết lập mạng.....	44
6.1 Kết nối private với các dịch vụ khác trong cùng 1 VPC Network.....	44
6.2 Kết nối dịch vụ Viettel Open Virtual Private Cloud (vOVPC)	45
6.3 Kết nối với dịch vụ Viettel Database Service (vDBS).....	47
6.4. CNF	48
6.5. Cấu hình DNS của Viettel IDC	49
7. Triển khai ứng dụng trên cụm	51
7.1 Cài đặt môi trường điều khiển phía người dùng	51
7.2 Hướng dẫn tải file Kube-config & sử dụng Kubernetes Dashboard	54
7.2.1 Tải file Kube-config.....	54
7.2.2 Sử dụng Kubernetes Dashboard	54
7.3 Tính năng Horizontal Pod Autoscaling.....	56
7.3.1 Tổng quan	56
7.3.2 Cơ chế hoạt động.....	57
7.3.3 Hướng dẫn cấu hình các metrics HPA	58
8. Quản lý chi phí sử dụng	62
8.1 Kiểm tra thời gian hoạt động của các Worker Node	62
8.2 Xem lịch sử thông báo cước & lịch sử thanh toán cước.....	63
8.3 Xem & tải lịch sử các hành động với cluster	65
9. Theo dõi cụm	65
9.1 Sử dụng Prometheus, Grafana	65
9.1.1 Prometheus	66

9.1.2 Grafana	67
9.2 Sử dụng công cụ Viettel Cloud Watch (coming soon).....	67
10. Các câu hỏi thường gặp (FAQ).....	67
10.1 Sự khác nhau cơ bản giữa Container, Docker và Kubernetes là gì?	67
10.2 Dịch vụ vOKS cung cấp gì cho khách hàng?.....	68
10.3 Khác biệt giữa Node Storage và Persistent Storage trong 01 cụm K8s?	68
10.4 vOKS có tương thích với các ứng dụng dạng container đang chạy trên Cloud của các nhà cung cấp khác?.....	68

1. Tổng quan dịch vụ

1.1 Một số khái niệm cơ bản về Kubernetes

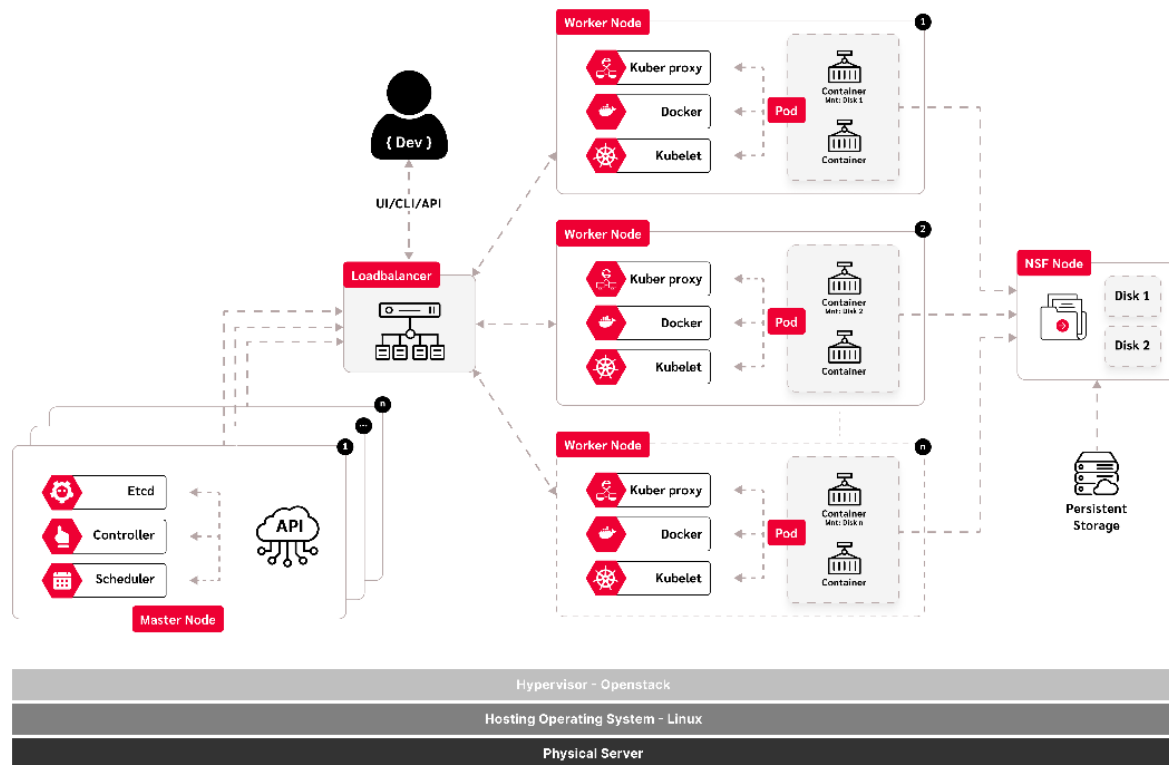
1.1.1 Kubernetes & kiến trúc của 1 cụm tài nguyên Kubernetes

[Kubernetes](#) là một nền tảng nguồn mở, có khả năng chuyển dịch nhanh chóng, có thể mở rộng để quản lý các ứng dụng được đóng gói và các service, giúp thuận lợi trong việc cấu hình và tự động hoá việc triển khai ứng dụng.

Các thành phần cơ bản của Kubernetes gồm:

- [Control Plane Components](#) (Master Node): gồm các thành phần giúp quản lý trạng thái chung của cụm
- [Node Components](#) (Worker Node): gồm các thành phần chạy trên mọi worker node, duy trì các pod đang chạy và cung cấp môi trường thời gian (runtime environment) chạy Kubernetes
- [Addons](#): sử dụng tài nguyên Kubernetes (DaemonSet, Deployment, v.v.) giúp mở rộng chức năng của Kubernetes và triển khai các tính năng của cụm. Ví dụ: DNS, Web UI (Dashboard), Container Resource Monitoring, Cluster-level Logging

Kiến trúc cơ bản của 1 cụm tài nguyên Kubernetes (Cluster Kubernetes) mà Viettel IDC đang cung cấp như sau:



1.1.2 Control Plane (Master Node)

Control Plane (Master Node) là server điều khiển các Worker Node chạy các Pod chứa ứng dụng. Master Node bao gồm 04 thành phần chính:

- Kube API Server: là thành phần giúp các thành phần khác liên lạc nói chuyện với nhau. Lập trình viên khi triển khai ứng dụng sẽ gọi API Server này;
- Kube Scheduler: Thành phần này lập lịch triển khai cho các ứng dụng, ứng dụng được đặt vào Worker Node nào để chạy;
- Kube Controller Manager: Thành phần đảm nhiệm phần quản lý các Worker Nodes, kiểm tra trạng thái các Worker “sống hay chết” (Node Controller), đảm nhận việc nhân bản ứng dụng, v.v.;
- Etcd: là cơ sở dữ liệu của Kubernetes, tất cả các thông tin của Kubernetes được lưu trữ cố định tại đây;

Ngoài 4 thành phần chính trên, control plane có thể tùy chọn tích hợp với nền tảng của nhà cung cấp đám mây thông qua thành phần cloud-controller-manager

1.1.3 Node (Worker Node)

[Node](#) (Worker Node) là server chạy các Pod chứa ứng dụng trên đó, bao gồm 03 thành phần chính:

- [Container runtime](#): thành phần giúp chạy các ứng dụng dưới dạng Container. Kubernetes hỗ trợ nhiều loại Container runtime như Docker, containerd, CRI-O;
- [Kubelet](#): thành phần giao tiếp với Kubernetes API Server, đồng thời đóng vai trò quản lý các container;
- [Kubernetes Service Proxy](#) (tùy chọn): đảm nhận việc phân tải giữa các ứng dụng;

Các node được tập hợp thành từng nhóm có cấu hình giống nhau được gọi là Node Group. Node group có thể tăng thêm hoặc giảm đi một cách linh hoạt và phân chia thành nhiều nhóm trong 1 cụm tài nguyên Kubernetes để đáp ứng nhu cầu triển khai ứng dụng. Người dùng có thể cấu hình Worker Autoscale theo từng Node Group giúp tối ưu tài nguyên và hiệu suất của ứng dụng.

1.1.4 Kubectl

[Kubectl](#) là agent chạy trên từng worker node trong cụm Kubernetes cluster. Kubelet hoạt động dưới dạng PodSpec. Kubelet lấy một tập hợp các PodSpec được cung cấp thông qua các cơ chế khác nhau và đảm bảo rằng các container được mô tả trong các PodSpec đó đang chạy và khỏe mạnh. Kubelet không quản lý các container không được tạo bởi Kubernetes.

1.1.5 Pod

[Pod](#) là đơn vị tính toán nhỏ nhất có thể triển khai mà khách hàng có thể tạo và quản lý trong Kubernetes. Một Pod bao gồm một hoặc nhiều containers, với tài nguyên mạng/vùng lưu trữ (storage) được chia sẻ và một đặc điểm kỹ thuật về cách chạy các containers. Nội dung của Pod luôn được định vị và đồng thời lên lịch, và chạy trong bối cảnh chung. Pod mô hình hóa một "máy chủ logic" dành riêng cho ứng dụng: nó chứa một hoặc nhiều containers ứng dụng được kết hợp tương đối chặt chẽ. Trong bối cảnh không phải đám mây, các ứng dụng được thực thi trên cùng một máy vật lý hoặc máy ảo tương tự như

các ứng dụng đám mây được thực thi trên cùng một máy chủ logic. Cũng như container ứng dụng, Pod có thể chứa init container chạy trong quá trình khởi động Pod.

1.1.6 Image

[Image](#) đại diện cho dữ liệu nhị phân bao bọc một ứng dụng và tất cả các gói phần mềm phụ thuộc (dependencies) của nó. Image có thể chạy độc lập và đưa ra các giả định được xác định rất rõ ràng về môi trường thời gian chạy của chúng. Image này thông thường được lưu trữ ở một repository tập trung, chẳng hạn Docker Hub là nơi chứa Images chính thức của nhiều ứng dụng phổ biến như nginx, mysql, wordpress, redis, v.v.

1.1.7 Deployment

Một [Deployment](#) cung cấp bản cập nhật khai báo cho Pods, Replica Set. Người dùng mô tả trạng thái mong muốn trong một Deployment và Deployment Controller thay đổi trạng thái thực tế thành trạng thái mong muốn với tốc độ được kiểm soát. Ngoài ra, người dùng có thể xác định các Deployment để tạo các ReplicaSets mới hoặc xóa các Deployment hiện có và sử dụng tất cả các tài nguyên của chúng với Deployment mới.

1.1.8 ReplicaSet

[ReplicaSet](#) được tạo ra nhằm duy trì một tập hợp gồm các bản sao Pod ổn định đang chạy ở bất cứ thời điểm nào. ReplicaSet thường được sử dụng để đảm bảo sự sẵn sàng của một nhóm các Pods, dễ dàng nhân bản thêm hay giảm số lượng Pod.

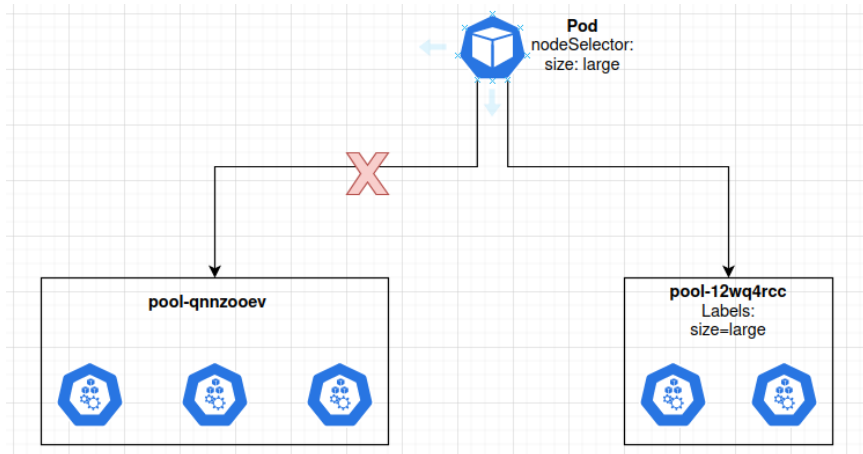
1.1.9 Service

[Service](#) là một cách trừu tượng để expose một ứng dụng đang chạy trên một tập hợp các Pod như một dịch vụ mạng. Với Kubernetes, người dùng không cần phải sửa đổi ứng dụng của mình để sử dụng cơ chế khám phá dịch vụ không quen thuộc. Kubernetes cung cấp cho các Pods địa chỉ IP của riêng chúng và một tên DNS duy nhất cho một nhóm các Pod và có thể cân bằng tải trên các nhóm đó.

1.1.10 Labels

[Labels](#) là các cặp key/value được gắn vào các đối tượng, chẳng hạn như Pod. Các label nhằm mục đích sử dụng để xác định các thuộc tính nhận dạng của các đối tượng có ý nghĩa

và phù hợp với người dùng. Labels có thể được sử dụng để tổ chức và chọn các tập hợp con của các đối tượng (VD: gán labels theo chức năng frontend, backend, chạy trên môi trường dev, test, staging, production). Các labels có thể được gán vào các đối tượng tại thời điểm tạo và sau đó được thêm vào và sửa đổi bất kỳ lúc nào. Mỗi đối tượng có thể có một tập hợp các key/value labels được xác định. Mỗi key phải là duy nhất cho một đối tượng nhất định.



Để sử dụng label ta dùng đến nodeSelector. Ý tưởng của việc sử dụng tham số nodeSelector này đó là mong muốn cấu hình các Pod chỉ chạy trên một tập các node có label nhất định.

Ví dụ như hình, các pod gán **nodeSelector** với cặp key-value là **size:large** sẽ được scheduling vào node pool có label **size=large**

1.1.11 Taints

[Taint](#) là một loại nhãn đặc biệt dùng để đánh dấu node có những đặc trưng nào đó, xác định bởi một bộ key-value và một action khi Pod không thỏa mãn điều kiện key-value đó.

Gán Taint vào một node giống như việc chúng ta khóa một node bằng một cái ổ khóa (coi taint là ổ khóa), và (ổ khóa) này đặc trưng bởi một label key=value và một action kèm theo (gọi là effect nếu không có (chìa khóa)).

Action này có thể là:

- NoSchedule ==> Không có (chìa khóa) thì sẽ không được lên lịch chạy trên node này.
- NoExecute ==> Không có (chìa khóa) thì sẽ có thể được lên lịch trên node này nhưng

Pod sẽ không được thực thi.

- PreferNoSchedule ==> Kubernetes sẽ cố né tránh nhất schedule pod k có tolerations lên node bị taints

Có (khóa) thì phải có (chìa). Để mở khóa (Taint) cho phép Pod có thể được chạy trên Node có Taint, thì Pod đó phải khai báo Tolerations (chìa khóa) đặc trưng với một label key=value và một Action kèm theo tương tự như với Taint.

Lưu ý: Một node có thể có nhiều Taint, khi đó Pod được gán vào Node nếu có tất cả các Toleration tương ứng với các Taint của node.

1.1.12 Worker Autoscale

Worker Autoscale là tính năng tự động tăng giảm tài nguyên Worker Node theo Node Group dựa trên nguyên lý hoạt động cơ bản của Kubernetes là tính năng [Cluster Autoscaler](#) với các Node và

Cluster Autoscaler là công cụ tự động điều chỉnh kích thước của cụm Kubernetes khi đáp ứng một trong các điều kiện sau:

- Có các pod không chạy được trong cụm do thiếu tài nguyên dựa trên [Horizontal Pod Autoscaling \(HPA\)](#) với các pod trong Node.
- Có các Nodes trong cụm đã không được sử dụng hết trong một thời gian dài và các pod của chúng có thể được đặt trên các Nodes hiện có khác.

Ví dụ người dùng config mức tài nguyên CPU sử dụng của pod đạt trên 70% hoặc Memory sử dụng đạt 80% qua file yaml, thì sẽ tiến hành tự động sinh thêm Pod mới khi đạt điều kiện. Khi các pod mới được sinh ra không đủ tài nguyên để hoạt động trong số Nodes đang có, 1 Worker Node mới sẽ được sinh ra theo cấu hình Node Group được chỉ định label theo nhu cầu của người dùng.

Trường hợp Node Group không được chỉ định label hoặc có label giống nhau, các Worker Node được tăng ngẫu nhiên vào một trong những cấu hình đang có. Người dùng có thể theo dõi thông tin Worker Node được tăng thêm trên giao diện quản trị do Viettel IDC phát triển.

1.2 Giới thiệu dịch vụ Viettel Open Kubernetes Service (VOKS)

1.2.1 VOKS là gì?

Viettel Open Kubernetes Service (vOKS) là một dịch vụ ở lớp nền tảng (PaaS) cung cấp cho các Nhà phát triển phần mềm một môi trường ảo hóa ở lớp OS (Operation System) hoàn chỉnh, hỗ trợ người dùng phát triển, kiểm thử, triển khai và phân phối các ứng dụng trên nền tảng Kubernetes.

vOKS được phát triển và cung cấp trên nền tảng Public Cloud Openstack mạnh mẽ và ổn định của Viettel IDC, nhờ đó dịch vụ được triển khai và cung cấp hoàn toàn tự động, nhanh chóng và thuận tiện cho khách hàng. vOKS đáp ứng nhiều tiêu chuẩn và chứng chỉ dành cho dịch vụ Cloud của Việt Nam (G-Cloud) cũng như thế giới (ISO 9001:2015 - Hệ thống quản lý Chất lượng, ISO 27001:2013 - Hệ thống quản lý An toàn thông tin, ISO 27017:2015 - Kiểm soát bảo mật an toàn thông tin cho các dịch vụ đám mây, v.v.).

1.2.2 VOKS có những tính năng gì?

VOKS đem đến cho khách hàng toàn bộ [các tính năng hữu ích nhất của nền tảng Kubernetes](#) như sau:

- Service discovery và cân bằng tải: Kubernetes có thể expose một container sử dụng DNS hoặc địa chỉ IP của riêng nó. Nếu lượng traffic truy cập đến một container cao, Kubernetes có thể cân bằng tải và phân phối lưu lượng mạng (network traffic) để việc triển khai được ổn định.
- Điều phối bộ nhớ: Kubernetes cho phép tự động mount một hệ thống lưu trữ mà lập trình viên chọn, như local storages, public cloud providers, v.v.
- Tự động rollouts và rollbacks: Lập trình viên có thể mô tả trạng thái mong muốn cho các container được triển khai dùng Kubernetes và nó có thể thay đổi trạng thái thực tế sang trạng thái mong muốn với tần suất được kiểm soát. Ví dụ, có thể tự động hoá Kubernetes để tạo mới các container cho việc triển khai ứng dụng, xóa các container hiện có và áp dụng tất cả các resource của chúng vào container mới.
- Đóng gói tự động: lập trình viên cung cấp cho Kubernetes một cluster gồm các node mà

nó có thể sử dụng để chạy các tác vụ được đóng gói (containerized task), cho Kubernetes biết mỗi container cần bao nhiêu CPU và bộ nhớ (RAM). Kubernetes có thể điều phối các container đến các node để tận dụng tốt nhất các resource.

- Tự phục hồi: Kubernetes khởi động lại các containers bị lỗi, thay thế các container, xóa các container không phản hồi lại cấu hình health check do người dùng xác định và không cho các client biết đến chúng cho đến khi chúng sẵn sàng hoạt động.
- Quản lý cấu hình và bảo mật: Kubernetes cho phép lưu trữ và quản lý các thông tin nhạy cảm như: password, OAuth token và SSH key. Lập trình viên có thể triển khai và cập nhật lại secret và cấu hình ứng dụng mà không cần build lại các container image và không để lộ secret trong cấu hình stack của bạn.

Bên cạnh những tính năng gốc của nền tảng Kubernetes, Viettel IDC cũng phát triển thêm các tính năng trên trang quản trị dịch vụ tập trung với những tính năng hữu ích giúp người dùng quản trị cụm nhanh chóng và dễ dàng hơn, cụ thể:

Cung cấp Node Group

Tính năng Node Group cho phép nhà phát triển tạo nhóm, tùy chỉnh và quản lý các worker nodes có liên quan hoặc phục vụ đa dạng nhu cầu sử dụng hạ tầng trong cùng 1 cụm tài nguyên Kubernetes.

Quản trị các Nodes tập trung

Quản trị các Worker Nodes theo label, taints, cấu hình autoscale theo cùng nhóm cấu hình Node group; có thể thêm và xóa các Node groups nhanh chóng, dễ dàng; quản lý chi phí sử dụng tập trung bằng self-service portal của Viettel IDC.

Nâng cấp phiên bản mới

Hỗ trợ các version mới nhất và nâng cấp version liên tục, mang lại những cải thiện về hiệu năng, bảo mật nhằm hỗ trợ khách hàng sử dụng những tính năng mới nhất của nền tảng Kubernetes.

Lưu trữ Persistent Volume với giải pháp NFS

Tích hợp 01 NFS Node giúp lưu trữ dữ liệu dài hạn, tự động cấu hình chia sẻ giữa các

Node. Khách hàng có thể chủ động bổ sung thêm dung lượng khi cần trên trang quản trị dịch vụ.

Tự động hoá & phục hồi Pod

Kubernetes giúp tự động tạo Pod mới, xoá Pod cũ và phân phối tài nguyên của Pod cũ vào Pod mới, không gây downtime. Ngoài ra, tự động reboot các Pod bị lỗi, xoá hoặc thay thế các Pod không phản hồi healthcheck.

Cài đặt và sử dụng các công cụ giám sát nhanh chóng

Khởi tạo các công cụ hỗ trợ như Kubernetes Dashboard, Prometheus và Grafana chỉ bằng 01 nút click ngay trên giao diện quản trị dịch vụ do Viettel IDC phát triển.

Kiểm soát tác động tới cluster

Kiểm soát mọi thay đổi với tính năng xem lại lịch sử thao tác trên cụm tài nguyên Kubernetes; bao gồm tạo, sửa Node group, cài đặt công cụ hỗ trợ, bổ sung dung lượng NFS Storage,...

Tích hợp Viettel CloudWatch (coming soon)

Dễ dàng thu thập log, giám sát các thành phần như Pods, Volumes, Nodes trong Kubernetes thông qua công cụ Viettel CloudWatch của Viettel IDC.

1.2.3 VOKS tính tiền như thế nào?

Người dùng có thể lựa chọn giữa 2 hình thức thanh toán gồm:

- Saving Plan: là hình thức thanh toán khi người dùng đăng ký & trả tiền trước theo tháng, giúp tiết kiệm về ngân sách và phù hợp với mục đích sử dụng lâu dài, ổn định
- Pay As You Go (PAYG): là hình thức thanh toán khi người dùng sử dụng trước dịch vụ và trả sau, chi phí tài nguyên Worker Node và các thành phần khác như Master Node, Persistent Volume, IP và băng thông được tính theo giờ căn cứ trên thời gian tạo và xóa của các Worker Node và các thành phần khác. Chi phí sử dụng được tổng hợp và thanh toán hàng ngày từ số dư tài khoản online của khách hàng. danh sách gói đăng ký dịch vụ thanh toán theo số lượng tài nguyên sử dụng, [dựa trên tính năng Worker Node Autoscale](#).

Chi tiết bảng giá của từng hình thức thanh toán xem tại [trang thông tin dịch vụ](#)

2. Cluster

2.1 Cluster là gì?

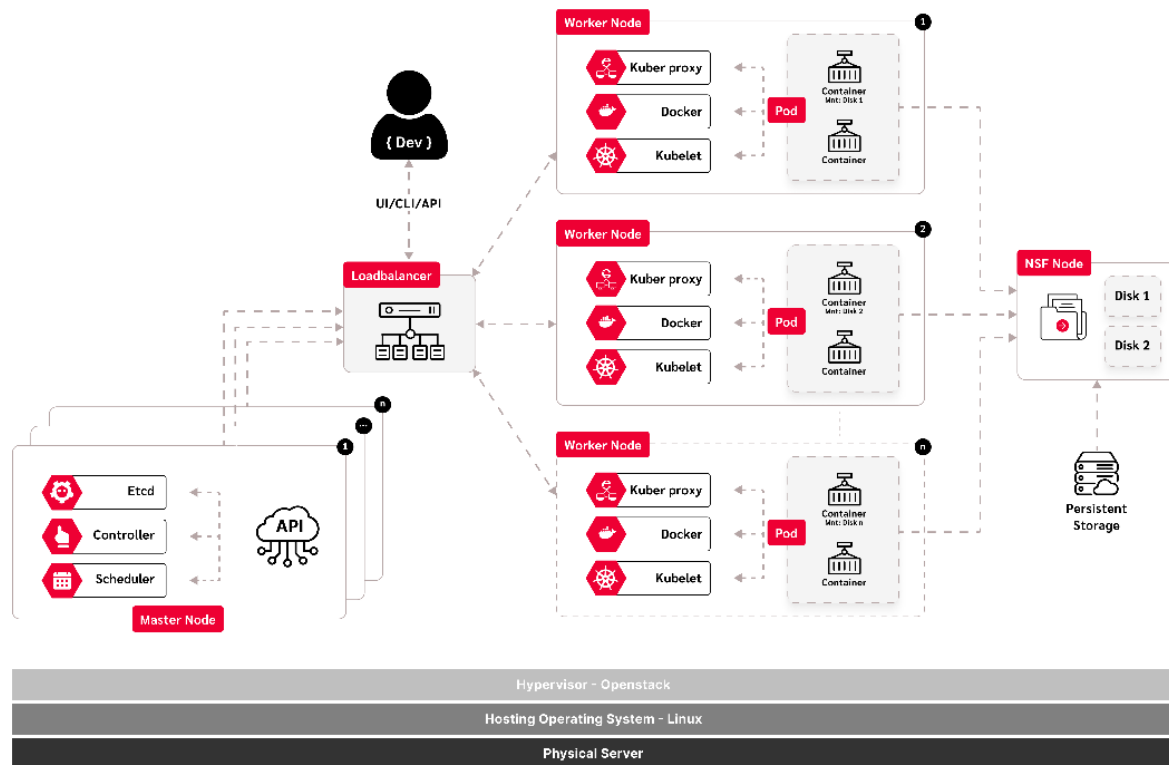
Một cụm Kubernetes ([Cluster](#)) bao gồm một [Control plane](#) (một hoặc nhiều Master Nodes); một tập hợp các Worker Nodes (còn được gọi là các [Nodes](#)), chạy các ứng dụng được chứa trong container. Mỗi cụm cần ít nhất một Worker Nodes để chạy [Pod](#).

[Worker Node\(s\)](#) lưu trữ các [Pod](#) là các thành phần của tải ứng dụng. [Control plane](#) quản lý các worker node(s) và các [Pod](#) trong cụm. Trên môi trường production (môi trường thực tế cho người dùng sử dụng), [control plane](#) thường chạy trên nhiều máy tính và một cụm thường chạy nhiều [Nodes](#), cung cấp khả năng chịu lỗi và tính khả dụng cao.

2.2 Cluster của dịch vụ vOKS

Dịch vụ vOKS mà Viettel IDC đang cung cấp là 01 [cụm tài nguyên Kubernetes](#) bao gồm [Master Nodes](#) và [Worker Node\(s\)](#), trong khi các tính năng và thành phần khác như Autoscale, service LB (Ingress Controller), Upgrade version, Persistent volume... được hỗ trợ đầy đủ.

Kiến trúc cơ bản của 01 [cụm tài nguyên Kubernetes](#) mà Viettel IDC đang cung cấp như sau:



Tùy theo hình thức thanh toán mà KH lựa chọn từ đầu, [cụm tài nguyên Kubernetes](#) có thể được khởi tạo với số lượng các thành phần khác nhau, cụ thể:

- Với hình thức Saving Plan, 01 [cụm tài nguyên Kubernetes](#) bao gồm:
 - o 01 hoặc 03 Master Nodes để đảm bảo tính sẵn sàng cao (High Availavility) cho cụm
 - o Tối thiểu 3 Worker Node trong 01 [Node group](#) với cấu hình mà người dùng chọn
 - o Dung lượng NFS Storage để sử dụng [Persistent Volume](#)
 - o 2 IP cho [API Server](#) và [Kubernetes Dashboard](#).
 - o Người dùng có thể tùy chọn mua thêm các Node group với cấu hình khác nhau để phục vụ nhu cầu phù hợp với dịch vụ chạy trên cụm; mua thêm Ingress Controller để sử dụng như Ingress giúp hiển thị các định tuyến HTTP và HTTPS từ bên ngoài cụm tới các dịch vụ trong cụm
- Với hình thức PAYG, 01 [cụm tài nguyên Kubernetes](#) bao gồm:
 - o 01 hoặc 03 Master Nodes để đảm bảo tính sẵn sàng cao (High Availavility) cho cụm

- Tối thiểu 01 Worker Node trong 01 [Node group](#) với cấu hình mà người dùng chọn. Người dùng có thể thay đổi cấu hình [Worker Autoscale](#) các Node group bằng cách điều chỉnh số lượng min node, max node trên giao diện quản trị của Viettel IDC để đáp ứng nhu cầu về tài nguyên node khi triển khai ứng dụng
- Dung lượng NFS Storage để sử dụng [Persistent Volume](#)
- 2 IP cho [API Server](#) và [Kubernetes Dashboard](#).
- Ingress Controller để sử dụng như Ingress giúp hiển thị các định tuyến HTTP và HTTPS từ bên ngoài cụm tới các dịch vụ trong cụm

2.3 Khởi tạo cluster

Người dùng có thể lựa chọn khởi tạo cluster với danh sách gói trên trang thông tin dịch vụ. Để tạo cluster phù hợp, người dùng cần cân nhắc một vài lựa chọn:

- Dùng thử để trải nghiệm dịch vụ hay đăng ký mới trả tiền trước?
- Lựa chọn hình thức thanh toán Saving Plan hay PAYG?
- Lựa chọn region nào để sử dụng dịch vụ, miền Bắc hay miền Nam?
- Lựa chọn cấu hình nào cho ứng dụng được deploy trên Kubernetes?
- Lựa chọn phiên bản Kubernetes phù hợp với mục đích sử dụng.

2.3.1 Khởi tạo 01 cluster dùng thử

Bước 1: Trên [trang thông tin dịch vụ](#) Viettel Open Kubernetes Service (vOKS), Người dùng kéo xuống mục “Các gói dịch vụ”, click “Dùng thử” ở gói dịch vụ muốn dùng thử. Hệ thống sẽ chuyển sang màn hình Đăng ký dùng thử

Lưu ý: Trong vòng 01 tháng Người dùng chỉ được phép dùng thử 01 gói dịch vụ. Nếu có nhu cầu dùng thử đặc biệt, hãy liên hệ với chúng tôi để được hỗ trợ

Bước 2: Trên màn hình Đăng ký dùng thử, Người dùng có thể:

- Lựa chọn phiên bản cho cụm tài nguyên Kubernetes theo nhu cầu

- Lựa chọn **Tạo mới VPC Network** hoặc **Chọn VPC Network đã tạo** để kết nối với các dịch vụ như Viettel Database Service (vDBS), Viettel Open Virtual Private Cloud (vOVPC)

Click chọn **Mua ngay** để sang bước Xác nhận đơn hàng dùng thử. Nếu có nhu cầu chọn lại gói dùng thử, Người dùng click chọn “Quay lại”.

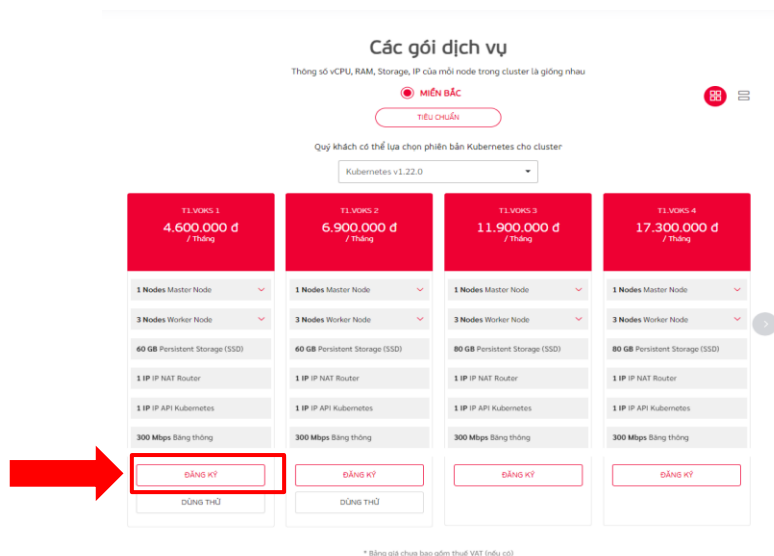
Bước 3: Trên màn hình Xác nhận, Người dùng tick vào ô đồng ý với các điều khoản sử dụng dịch vụ của Viettel IDC, sau đó click chọn “Xác nhận” để hoàn thành tạo đơn hàng Dùng thử.

Bước 4: Hệ thống hiển thị thông báo tạo đơn hàng Dừng thử thành công. Người dùng sẽ nhận được email chứa thông tin gói dịch vụ vừa đăng ký.

Trên trang chủ Viettel IDC, Người dùng click vào góc phải màn hình phần Tài khoản → click chọn **Quản lý dịch vụ** để xem trạng thái khởi tạo, các thông tin chi tiết về quản trị và sử dụng dịch vụ.

2.3.2 Khởi tạo 01 cluster Saving Plan

Bước 1: Trên trang thông tin dịch vụ Viettel Open Kubernetes Service (vOKS), Người dùng kéo xuống mục **Các gói dịch vụ**, click **Đăng ký** ở gói dịch vụ muốn đăng ký dùng trả phí. Hệ thống sẽ chuyển sang màn hình Đăng ký.



Bước 2: Trên màn hình Đăng ký:

Người dùng nhập Số lượng gói dịch vụ, Thời gian sử dụng, Khu vực

- Lựa chọn phiên bản cho cụm tài nguyên Kubernetes theo nhu cầu
- Người dùng có thể lựa chọn **Tạo mới VPC Network** hoặc **Chọn VPC Network đã tạo** để kết nối với các dịch vụ trong hệ sinh thái Viettel Cloud (ví dụ như Viettel Database Service (vDBS), Viettel Open Virtual Private Cloud (vOVPC)...)
 - Gói dịch vụ Saving Plan có tối thiểu 01 Node Group (Base) với 3 Worker Node cùng cấu hình của gói người dùng đã chọn ở bước 1. Người dùng không thể remove Node

group này và có thể thêm các Node Group & số lượng Worker Node phù hợp với nhu cầu sử dụng của mình.

- Người dùng tùy chọn mua add-ons thêm các thành phần:

o High Availability (HA): bổ sung thêm 02 Master Node & 01 Loadbalancer để chạy Cluster theo mô hình Multi Master Node, giúp tăng mức độ sẵn sàng cho dịch vụ.

Lưu ý: K8s Cluster đã khởi tạo theo mô hình Multi Master Node sẽ không thể chuyển về mô hình Single Master Node. Option HA không thể mua bổ sung sau khi Cluster đã được khởi tạo.

o Ingress Controller: Người dùng chọn mua thêm Ingress Controller để định tuyến cấu hình Loadbalancer HTTP cho các ứng dụng chạy trên Kubernetes

o Băng thông: Băng thông mặc định 300Mbps, Người dùng có thể mua thêm nếu cần;

The screenshot displays the Viettel IDC web interface for configuring a service. The top navigation bar includes the Viettel IDC logo, links for 'Dịch vụ', 'Giải pháp', 'Bảng giá', 'Tin tức', 'Đối tác & Đại lý', 'Về Viettel IDC', and 'Q&A'. A progress bar indicates the steps: 1. Đăng ký, 2. Giỏ hàng, 3. Xác nhận, 4. Thanh toán.

The main content area is divided into two columns:

- Thông tin gói (Package Information):**
 - T1.vOKS 2:** 7.000.000đ/Tháng
 - Số lượng:** 1
 - Thời gian:** 1 Tháng
 - Khu vực:** Miền Nam
 - Phiên bản:** Kubernetes v1.30.5
 - CNI:** Calico
- Thông tin dịch vụ (Service Information):**
 - T1.vOKS 2:** 7.000.000 đ
 - Master Node 2:** 1 Node, Persistent Storage (SSD): 60 GB, IP Public: 2 IP, T1.vOKS 2: 1 Nodes, Băng thông: 300 Mbps, vCPU/Node: 4 vCPU, RAM/Node: 8 GB, SSD/Node: 50 GB
 - Số lượng:** 1
 - Thời gian:** 1 Tháng
 - Phí khởi tạo:** 0 đ
 - Tiền chưa VAT:** 7.000.000 đ
 - Tổng:** 7.700.000 đ

Below the package information, there is a section for VPC Network configuration:

Chọn VPC Network đã tạo hoặc tạo VPC Network mới để đăng ký dịch vụ (Các dịch vụ được tạo chung VPC Network có thể kết nối Private)

VPC Network *

Tạo VPC Network mới

Chọn VPC Network cho gói dịch vụ

At the bottom right, there are two buttons: 'Quay lại' (Back) and 'MUA NGAY' (Buy Now).

- Người dùng click chọn Mua ngay để sang màn hình Giỏ hàng.

Bước 3: Trên màn hình Giỏ hàng: Người dùng click chọn “Xác nhận” để sang màn hình Xác nhận đơn hàng:

1 Đăng ký —
 2 Giỏ hàng —
 3 Xác nhận —
 4 Thanh toán

DỊCH VỤ THANH TOÁN NGAY

DỊCH VỤ	SỐ LƯỢNG	THỜI GIAN	CHƯA BAO GỒM PHÍ VAT	TỔNG TIỀN SAU VAT	...
T1.vOKS 1 (OpenKubernetesService)			4.600.000 đ	5.060.000 đ	
Worker Node 1: 3 Nodes, Master Node 1: 1 Node, Persistent Storage (SSD): 60 GB, IP NAT Router: 1 IP, IP API Kubernetes: 1 IP, Băng thông: 300 Mbps, vCPU/Node: 2 vCPU, RAM/Node: 4 GB, SSD/Node: 50 GB, vCPU/Node: 2 vCPU, RAM/Node: 4 GB, SSD/Node: 50 GB	1	1 Tháng	4.600.000 đ	5.060.000 đ	🗑
Tổng tiền			4.600.000 đ	5.060.000 đ	

MUA THÊM DỊCH VỤ KHÁC

XÁC NHẬN

Bước 4: Trên màn hình Xác nhận:

- Người dùng tick vào ô Xuất hoá đơn GTGT, nhập các thông tin liên quan để nhận hoá đơn;
- Người dùng tick vào ô Xác nhận đồng ý với các điều khoản sử dụng dịch vụ của Viettel IDC;
- Người dùng nhập thông tin voucher (mã ưu đãi, giảm giá) nếu có;
- Người dùng click chọn “Thanh toán” để sang màn hình Thanh toán cho đơn hàng

1 Đăng ký —
 2 Giỏ hàng —
 3 Xác nhận —
 4 Thanh toán

Thông tin cá nhân

HO VÀ TÊN: EMAIL: CMND/HỘ CHIẾU:

Xuất hoá đơn GTGT

☒ Xuất Hóa đơn GTGT cho Đơn hàng này! Viettel IDC sẽ xuất HÓA ĐƠN ĐIỆN TỬ

Vui lòng điền ĐÚNG và ĐỦ thông tin xuất hóa đơn theo Giấy phép đăng ký kinh doanh của tổ chức/doanh nghiệp, Viettel IDC sẽ sử dụng thông tin quý khách đã nhập để xuất hóa đơn. Trường hợp quý khách không nhập thông tin hóa đơn, Viettel IDC sẽ xuất hóa đơn theo thông tin cá nhân người mua hàng.

☒ Tôi xác nhận mọi thông tin đăng ký đã đúng, tôi đồng ý với các điều khoản quy định của Công ty Viettel IDC

Lưu ý: Kể từ ngày 01/05/2023 Viettel IDC có quy định mới với số dư trong tài khoản khuyến mại, giới hạn thanh toán cho 1 số dịch vụ như Viettel Domain, Viettel SSL..., thông tin chi tiết thể hiện tại điều khoản 6.4 trong "Thỏa thuận sử dụng dịch vụ trên Website" tại [đây](#)

Thông tin đơn hàng

Mã voucher: ÁP DỤNG

Sử dụng được nhiều voucher. Mỗi voucher áp dụng cách nhau bằng dấu ", ".

Tổng trước VAT: 4.600.000 đ

Tổng sau VAT: 5.060.000 đ

Tổng khấu trừ: - 0 đ

Tổng: 5.060.000 đ

Quay lại
Thanh toán

Bước 5: Trên màn hình Thanh toán:

- Người dùng lựa chọn hình thức thanh toán phù hợp và thực hiện Thanh toán.
- Người dùng click chọn “Xác nhận” để hoàn thành đơn hàng Đăng ký dùng trả phí.

The screenshot displays the checkout interface with a progress bar at the top: 1 Đăng ký, 2 Giỏ hàng, 3 Xác nhận, 4 Thanh toán. The main content is divided into three sections:

- Thanh toán bằng tài khoản của tôi:** Shows a masked card number with a "NẠP NGAY" button. A note states: "Hệ thống mặc định trừ tiền trong tài khoản của quý khách để thanh toán cho đơn hàng này! Nếu số dư tài khoản không đủ, quý khách cần nạp thêm hoặc tiếp tục lựa chọn phương thức thanh toán dưới đây."
- Phương thức thanh toán:**
 - ☒ Thanh toán Online: Giao dịch thực hiện tiện lợi, nhanh chóng, an toàn.
 - ☒ Thẻ ATM nội địa (có Internet Banking)
 - ☐ Thẻ Visa, Master, JCB, UnionPay
 - ☐ Ví MoMo
 - ☐ Quét QR Code (App ngân hàng, Viettel Money)
 - ☐ Thanh toán sau, chuyển khoản ngân hàng
- Thông tin đơn hàng:**
 - Tạm tính: 5.060.000 đ
 - Tiền khuyến mại: 0 đ
 - Tổng khấu trừ: 0 đ
 - Tổng: 5.060.000 đ

At the bottom right, there are two buttons: "Quay lại" and "Xác nhận", with a red arrow pointing to the "Xác nhận" button.

Ảnh 2.3.3 Khởi tạo 01 cluster Pay-as-you-go (PAYG)

The screenshot shows the "Các gói dịch vụ" (Service Packages) page for the "Pay as you go" model. It includes a navigation bar with links like "Dịch vụ", "Giải pháp", "Bảng giá", "Tin tức", "Hỗ trợ", "Liên hệ", and "Về Viettel IDC".

Pay as you go (Phản ứng nhu cầu sử dụng máy chủ linh hoạt theo giờ)

Thông số vCPU, RAM, Storage, IP của mỗi node trong cluster là giống nhau.

☒ MIỄN BẮC ☐ MIỄN NAM

☐ TIÊU CHUẨN ☐ CAO GẤP

Lựa chọn phiên bản Kubernetes cho Cluster: Kubernetes v1.22.0

Lựa chọn loại HA cho cluster: 1 Master node

TL WORKS 1	TL WORKS 2	TL WORKS 3	TL WORKS 4
4.200 đ / giờ	5.300 đ / giờ	7.000 đ / giờ	8.900 đ / giờ
1 Nodes Master Node	1 Nodes Master Node	1 Nodes Master Node	1 Nodes Master Node
1 Nodes Worker Node	1 Nodes Worker Node	1 Nodes Worker Node	1 Nodes Worker Node
60 GB Persistent Storage (SSD)	60 GB Persistent Storage (SSD)	80 GB Persistent Storage (SSD)	80 GB Persistent Storage (SSD)
2 IP IP Public	2 IP IP Public	2 IP IP Public	2 IP IP Public
300 Mbps băng thông	300 Mbps băng thông	300 Mbps băng thông	300 Mbps băng thông
Giá Thành Phần Tham khảo	Giá Thành Phần Tham khảo	Giá Thành Phần Tham khảo	Giá Thành Phần Tham khảo
✓ Giá cluster (gồm Master Node, Persistent Storage, IP băng thông): 3.000đ/giờ	✓ Giá cluster (gồm Master Node, Persistent Storage, IP băng thông): 3.500đ/giờ	✓ Giá cluster (gồm Master Node, Persistent Storage, IP băng thông): 4.000đ/giờ	✓ Giá cluster (gồm Master Node, Persistent Storage, IP băng thông): 4.500đ/giờ

At the bottom right, there is a chat bubble asking: "Xin chào Viettel IDC có thể hỗ trợ gì cho Quý khách?"

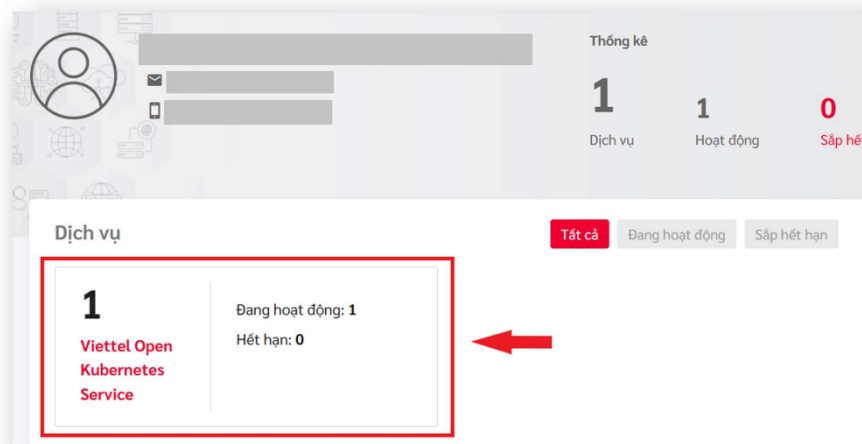
- Pay-as-you-go là hình thức thanh toán dùng trước trả sau, chi phí tài nguyên Worker

Node và các thành phần khác như Master Node, Persistent Volume, IP và băng thông được tính theo giờ căn cứ trên thời gian tạo và xóa của các Worker Node và các thành phần khác. Chi phí sử dụng được tổng hợp và thanh toán hàng ngày từ số dư tài khoản online của khách hàng.

- Người dùng thực hiện đăng ký gói PAYG theo 4 bước tương tự cluster Saving Plan, cước phí sẽ được tính toán & trừ vào tài khoản trên website của Người dùng theo ngày. Cách kiểm tra cước phí sử dụng dịch vụ vOKS PAYG được hướng dẫn cụ thể ở phần [3.2 Sử dụng dịch vụ](#)

2.4 Quản trị cluster

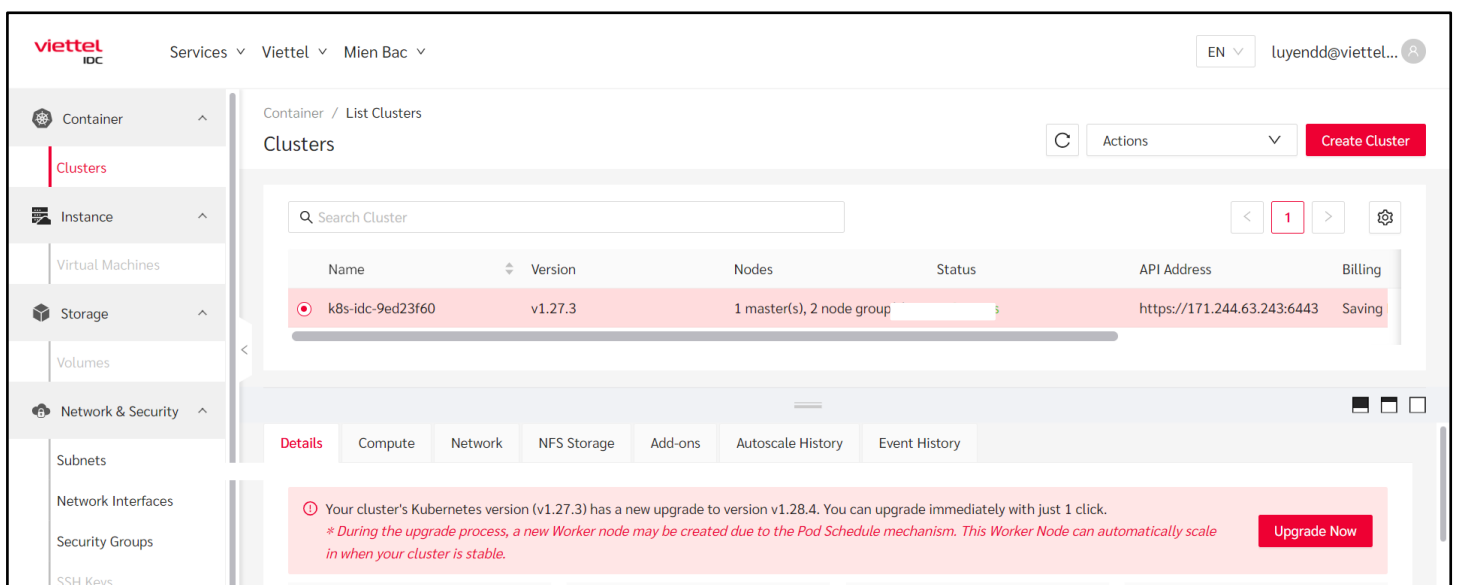
Bước 1: Trên màn hình Quản trị dịch vụ, Người dùng click chọn “Viettel Open Kubernetes Service” để truy cập vào portal Quản trị dịch vụ (Cloud Management Platform – CMP).



Bước 2: Người dùng được chuyển đến giao diện CMP quản lý dịch vụ

- Phần **List Cluster**: hiển thị danh sách các K8s cluster người dùng đang có
- Tab **Details**: hiển thị các thông tin chi tiết của K8s cluster: tên cluster, mã đơn hàng, trạng thái, API Address, số lượng Node Groups, hình thức Billing, tên VPC Network ngày bắt đầu – ngày kết thúc của gói dịch vụ
- Tab **Compute**: hiển thị thông tin chi tiết của các Node (Master, NFS) và Node group trong cluster

- Tab **Network**: hiển thị các thông tin network, vd: Security Group
- Tab **NFS Storage**: hiển thị thông tin NFS Storage – Persistent Storage
- Tab **Activity History**: hiển thị lịch sử scale out – scale in khi tăng giảm các node bằng autoscale
- Tab **Add-ons**: cho phép người dùng cài đặt các công cụ quản trị và giám sát hiệu suất của cluster như Dashboard, Prometheus, Grafana, Viettel Cloud Watch
- Tab **Event History**: hiển thị lịch sử tác động tới cụm Kubernetes



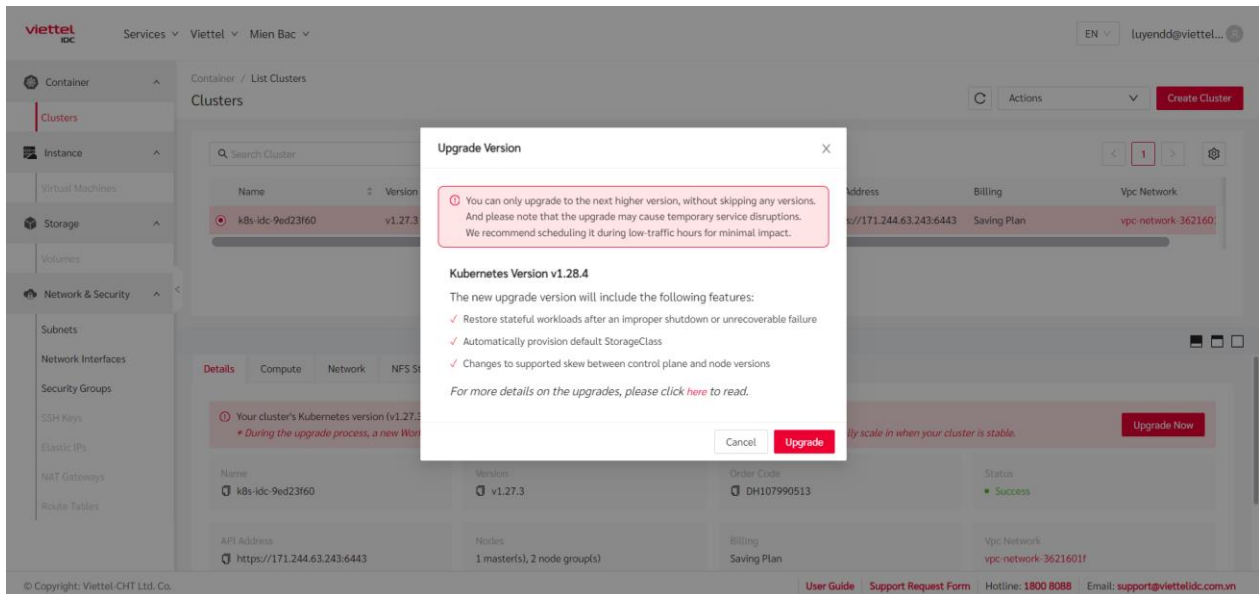
2.5 Nâng cấp phiên bản của cụm Kubernetes (Upgrade Kubernetes cluster version)

Viettel IDC hỗ trợ người dùng nâng cấp cụm Kubernetes trực tiếp trên trang quản trị dịch vụ. Người dùng có thể click **Upgrade Now** để tiến hành nâng cấp cụm.

Lưu ý:

- Người dùng chỉ có thể nâng cấp lên phiên bản cao hơn gần kề. VD: nâng cấp từ phiên bản 1.27 lên 1.28
- Việc nâng cấp có thể gây ra gián đoạn dịch vụ tạm thời. Viettel IDC khuyến nghị nên lên lịch vào những giờ ít lưu lượng truy cập để giảm thiểu tác động downtime.

- Trong quá trình nâng cấp, một Worker Node mới có thể được tạo do cơ chế [Pod Schedule](#). Worker node này có thể tự động giảm xuống khi cụm Kubernetes ổn định.



2.6 Best practice khi nâng cấp phiên bản cho cụm Kubernetes

Để có trải nghiệm mượt mà và hài lòng nhất khi nâng cấp cụm cluster, dưới đây là một số gợi ý giúp khách hàng cân nhắc xuyên suốt quá trình nâng cấp phiên bản mới.

2.6.1. Chuẩn bị trước khi nâng cấp:

- Thực hiện nâng cấp trong thời điểm hệ thống tải thấp (sau giờ hành chính hoặc cuối tuần).
- Trước khi upgrade môi trường prod nên kiểm thử các ứng dụng có tương thích với K8S version mới không ở môi trường dev/uat trước, đặc biệt là các API K8S version mới.
- Backup config của cluster trước khi upgrade: sử dụng các tool CD hoặc backup manifest/helm chart
- Kiểm tra phiên bản hiện tại và phiên bản mục tiêu, đảm bảo chúng nằm trong phạm vi hỗ trợ nâng cấp trực tiếp.
- Trong trường hợp lưu data trong các nodes thì phải backup dữ liệu để tránh bị mất data.
- Sử dụng cluster HA master (03 master) và HA worker (từ 02 worker trở lên) để đảm

bảo không downtime.

- Triển khai ứng dụng dạng Workload Controller (Deployment, Replicaset, DaemonSet) với replica ≥ 2 , tránh sử dụng Pod đơn lẻ.

2.6.2. Thực hiện nâng cấp

- Nâng cấp lần lượt qua từng phiên bản (ví dụ: 1.28 → 1.29 → 1.30).
- Viettel IDC khuyến cáo khách hàng nên nâng cấp phiên bản mới nhất theo danh sách phiên bản Kubernetes mà Viettel IDC hỗ trợ.
- Không nên triển khai ứng dụng sử dụng local storage (empty dir, hostpath ..) mà nên sử dụng CSI
- Trong quá trình nâng cấp, Viettel IDC đã thực hiện thiết kế sizing tài nguyên cho ứng dụng đảm bảo có đủ tài nguyên (trong quá trình nâng cấp cụm có thể scale thêm 01 worker làm buffer trước khi upgrade version cluster).

2.6.3. Sau khi nâng cấp

- Kiểm tra trạng thái cluster & các node chạy trên version mới hoạt động ổn định.
- Lên kế hoạch định kỳ cho việc nâng cấp phiên bản cluster để tránh sử dụng các phiên bản hết hạn hỗ trợ
- Duy trì hệ thống giám sát và cảnh báo để theo dõi hiệu suất và trạng thái cluster sau khi nâng cấp

2.7 Xóa cluster

Hiện tại dịch vụ vOKS chỉ hỗ trợ người dùng gửi yêu cầu xóa cluster thông qua ticket với hình thức PAYG. Với hình thức Saving plan, người dùng không thể thực hiện xóa cluster do chính sách về hóa đơn và thanh toán trước dịch vụ

Bước 1: Trên màn hình Quản trị dịch vụ, Người dùng click chọn “Viettel Open Kubernetes Service” để truy cập vào màn hình quản lý gói cước, click tên gói cước để truy cập trang gửi yêu cầu xóa cluster

Dịch vụ Pay as You go



6

Viettel Open
Kubernetes
Service

Đang hoạt động: **6**

Hết hạn: **0**

[Click vào đây để quản trị dịch vụ](#)

KẾT QUẢ TÌM KIẾM

STT	Gói dịch vụ	Tên máy	Thông tin gói dịch vụ	Loại gói	Mã đơn hàng	Trạng thái máy chủ	Trạng thái gói dịch vụ	Ngày bắt đầu	Phí duy trì (đ)	Quản trị
	T2.vOKS 1		Master Node 1: 1 Nodes, vCPU/Node: 2 vCPU, RAM/Node: 4 GB, SSD/Node: 50 GB, Ingress Controller: 1, IP Public: 2 IP, Cluster 1: 1, Persistent Volume: 100 GB, Băng thông: 300 Mbps	Cao cấp	PG107990501	Không xác định	Đang xử lý	02/10/2024	5.300	

Bước 2: Người dùng click Gửi yêu cầu xóa dịch vụ, điền thông tin qua mẫu để Viettel IDC liên lạc hỗ trợ xóa cluster.

Quản trị 29.011.648 đ Nạp tiền

Trang chủ / Quản lý dịch vụ / Viettel Open Kubernetes

T2.vOKS 1 Đang hoạt động

Mã đơn hàng	Trạng thái máy chủ
PG107990524	Máy bất

 Hướng dẫn cấu hình
 Hủy dịch vụ

LỊCH SỬ THÔNG BÁO CƯỚC

STT	Thời gian ghi nhận cước	Giá
Không có dữ liệu		

Yêu cầu hủy dịch vụ Paygo

Họ và tên:

Số điện thoại: Email:

Tiêu đề:

Nội dung *

Tôi muốn hủy máy chủ có thông tin như sau: Mã đơn hàng - PG107990524, Tên VM - k8s-ic-e77f3594-machine-template-mtrf, IP - 10.7.10.59

Mã xác thực: 

Thời gian hoạt động: Từ 10:24 31/10/2024

Persistent Volume (SSD)	Số lượng Node Group
	1

Đơn thanh toán (bao gồm VAT)	Hình thức thanh toán	Trạng thái thanh toán	Thời gian thanh toán
Không có dữ liệu			

3. Quản lý truy cập

3.1 Sử dụng tính năng Identity Access Management (IAM)

Quản lý danh tính và truy cập (Identity Access Management – IAM) là tính năng do

Viettel IDC phát triển và vận hành. Quản trị viên có thể quản lý toàn bộ quyền truy cập của các người dùng khác thông qua các tính năng cơ bản của IAM như:

- IAM User
- IAM Group
- IAM Policy
- IAM Session
- Monitor
- Account setting

Để nắm được cách sử dụng toàn bộ tính năng IAM của Viettel IDC, người dùng có thể [xem thêm tại đây](#)

3.2 Các chính sách (Policy) phân quyền của dịch vụ vOKS

Để hỗ trợ người dùng dễ dàng quản lý, dịch vụ vOKS đang cung cấp các phân quyền như sau:

- List: cho phép xem thông tin trên giao diện quản trị
- Read: cho phép xem chi tiết tài nguyên
- Write: cho phép người dùng tạo, cập nhật hoặc xóa tài nguyên

Với từng quyền, người dùng có thể tương tác với các thành phần khác nhau của cụm trên giao diện quản trị của Viettel cung cấp, cụ thể:

- Cluster
- Cluster Instance
- Network
- Add-ons

4. Quản lý tài nguyên

Một node Kubernetes là một máy ảo có các tài nguyên chạy các ứng dụng được chứa trong container. Mỗi Node có các thành phần cơ bản sau:

- [Container runtime](#) – Phần mềm chịu trách nhiệm chạy các container.

- [Kubelet](#) – Đảm bảo rằng các container hoạt động bình thường và chạy trong Pod được liên kết của chúng.
- [Kube-proxy](#) – Duy trì các quy tắc mạng cho phép giao tiếp với các Pod.

Để biết thêm thông tin, xem [Nodes](#) trên trang tài liệu Kubernetes.

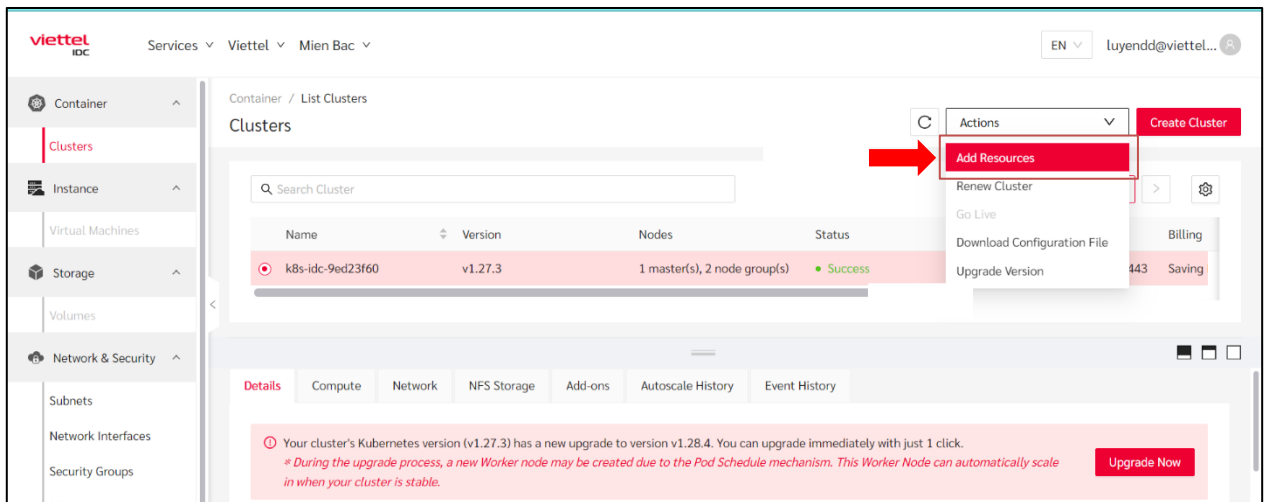
Node Group là một nhóm các node có cấu hình giống nhau được tập hợp để quản trị cấu hình nhanh chóng, dễ dàng. Node group có thể tăng thêm hoặc giảm đi một cách linh hoạt và phân chia thành nhiều nhóm trong 1 cụm tài nguyên Kubernetes để đáp ứng nhu cầu triển khai ứng dụng.

Người dùng có thể tạo thêm Node group, xem thông tin của Node group, sửa cấu hình Worker Autoscale theo từng Node Group giúp tối ưu tài nguyên và hiệu suất của ứng dụng.

4.1 Tạo Node group

4.1.1 Tạo thêm Node group cho cluster Saving Plan

Bước 1: Trên giao diện CMP, Người dùng click chọn Cluster cần mua bổ sung → chọn Action Add Resources



Bước 2: Hệ thống chuyển sang màn hình **Mua bổ sung:** Người dùng lựa chọn mua thêm Node group hoặc mua bổ sung thêm Worker Node cho Node Group:

- Click button  để tăng thêm Node group, thay đổi sizing của Node group và số lượng node mặc định của Node group. [Số lượng Node có thể thay đổi theo tính năng Autoscale của từng Node group](#)

- Tick chọn Worker Node, nhập/kéo thả số lượng Node cần mua bổ sung click chọn “Mua ngay”.

The screenshot shows the Viettel IDC service configuration interface. It includes a progress bar at the top with three steps: 1. Mua bổ sung dịch vụ (Buy additional services), 2. Xác nhận (Confirm), and 3. Thanh toán (Payment). The main content area is divided into two columns. The left column contains the 'Thông tin gói' (Package info) and 'Cấu hình Worker Node Group' (Worker Node Group configuration). The right column shows the 'Thông tin dịch vụ' (Service info) and a summary of costs. A red arrow points to the 'Thêm Node' (Add Node) button in the bottom right corner.

Bước 3: Xác nhận đơn hàng: *Thao tác tương tự như khi Đăng ký dùng trả phí.*

Bước 4: Thanh toán đơn hàng: *Thao tác tương tự như khi Đăng ký dùng trả phí.*

4.1.2 Tạo thêm Node group & Node cho gói dịch vụ PAYG

Khi đăng ký gói dịch vụ PAYG, khách hàng sẽ có 01 cluster với 01 Node group để sử dụng. Khách hàng có thể cấu hình [tùy chỉnh thông số min node, max node để đáp ứng nhu cầu autoscale](#) cấu hình theo nhu cầu

Với nhu cầu cần sử dụng thêm Node group, khách hàng thực hiện như sau:

Bước 1: Trên màn hình **List Clusters**, Chọn cluster cần thêm node group, chọn tab **Compute**

The screenshot shows the Viettel Cloud console interface. The left sidebar contains navigation links for Container, Clusters, Instance, Storage, and Network & Security. The main area displays a table of clusters. The cluster 'k8s-ids-84cx75ba' is highlighted. Below the table, there are tabs for Details, Compute, Network, NFS Storage, Add-ons, Autoscale History, and Event History. The 'Compute' tab is active, showing 'Node Groups (3)' and 'Nodes (6)'. A red arrow points to the 'Add node group' button in the top right corner of the 'Node Groups' section.

Bước 2: Trên màn hình tạo Node group – bước 1 Configure Node group, khách hàng có thể **Điền tên Node Group** theo nhu cầu, **Điền thông tin Labels, Taints** cần thêm cho node group, chọn Next

The screenshot shows the 'Add node group' configuration page in the Viettel Cloud console. The page has a progress bar with three steps: 1. Configure node group, 2. Set compute and scaling config, and 3. Review and create. The first step is active. The 'Name' field is highlighted with a red box. Below it, the 'Kubernetes labels' section is highlighted with a red box. At the bottom, the 'Kubernetes taints' section is highlighted with a red box. A red arrow points to the 'Add node group' button at the bottom right.

Lưu ý:

- Phần Key của Labels phải bắt đầu và kết thúc bằng kí tự số hoặc chữ, không dài hơn 63 kí tự, chỉ được chứa 2 kí tự đặc biệt là chấm (.) và gạch ngang (-)
- Phần Value của Labels phải bắt đầu và kết thúc bằng kí tự số hoặc chữ, không dài hơn 63 kí tự, chỉ được chứa 2 kí tự đặc biệt là chấm (.) và gạch ngang (-)
- Phần Key của Taints phải bắt đầu và kết thúc bằng kí tự số hoặc chữ, không dài hơn

253 kí tự, chỉ được chứa 4 kí tự đặc biệt là chấm (.), gạch chéo (/), gạch ngang (-) và gạch dưới (_)

- Phần Value của Taints phải bắt đầu và kết thúc bằng kí tự số hoặc chữ, không dài hơn 63 kí tự, chỉ được chứa 4 kí tự đặc biệt là chấm (.), gạch chéo (/), gạch ngang (-) và gạch dưới (_)
- o Với tùy chọn Effect là hành động khi Pods không tolerate taint, Khách hàng có thể lựa chọn 1 trong 3 loại như sau:
 - NoSchedule: Không có chìa khóa thì sẽ không được lên lịch trên Node này.
 - NoExecute: Không có chìa khóa thì sẽ không được lên lịch node này và những pod đang chạy trên node này không có toleration tương ứng sẽ bị remove khỏi node này.
 - PrefectNoSchedule: Kubernetes sẽ cố né tránh nhất schedule pod không có tolerations bị Taints

Bước 3: Trên màn hình tạo Node group – bước 2 Set compute & scaling config, khách hàng có thể **Lựa chọn cấu hình cho Node Group theo nhu cầu, tick enable/disable tính năng Worker Autoscale & điền số min node, max node**, chọn Next

The screenshot displays the 'Set compute and scaling config' step in the Viettel Cloud console. The interface is divided into two main sections: 'Node group compute configuration' and 'Node group scaling configuration'. In the compute section, the 'Resource Type' is set to 'T1.0KS 3 (2 vCPU - 3GB RAM - 60GB SSD)' and the 'Pricing' is set to '1.2000\$/h'. In the scaling section, 'Worker Node Autoscale' is enabled, 'Min Node' is set to 2, and 'Max Node' is set to 4. Red arrows point to the 'Configure node group' step indicator, the resource type dropdown, the pricing field, and the autoscale toggle/inputs. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Bước 4: Trên màn hình Review tạo Node group – bước 3 Review & Create, khách hàng có thể **Kiểm tra lại thông tin Node Group cần tạo thêm**, chọn Create để khởi tạo thêm Node Group

The screenshot displays the 'Review and create' step in the Viettel Cloud console. The breadcrumb trail at the top indicates the path: Container / List Clusters / Add node group. The main content area is divided into several sections:

- Node group configuration:** Shows the Name 'node-group-30-07'.
- Kubernetes labels (1):** A table with one label: Key 'kubernetes.io', Value 'test'.
- Kubernetes taints (1):** A table with one taint: Key 'system', Value 'locked', Effect 'PreferNoSchedule'.
- Set compute and scaling config:** Shows Resource Type 'T1-VKS 3 (2 vCPU - 2GB RAM - 40GB SSD)' and Pricing '1.2000/h'. It also includes a toggle for 'Worker Node Autoscale' and input fields for 'Min Nodes' (2) and 'Max Nodes' (4).

At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Create'. A red arrow points to the 'Create' button.

4.2 Xem thông tin Node group & các node bên trong

Người dùng có thể xem thông tin các Node groups tại tab Compute, phần Node Group. Bảng Node group sẽ bao gồm các Worker node đang có trong Node group, bảng Nodes có số lượng các nodes và các trường thông tin cụ thể như sau:

- **Node Name:** tên Node
- **Type:** loại node, gồm 3 loại là Master, Worker và NFS Node để sử dụng làm Persistent Volume
- **Resource Type:** cấu hình của Node
- **Node Group:** là nhóm cấu hình của node
- **Start Time:** thời gian node bắt đầu hoạt động
- **End Time:** thời gian node ngừng hoạt động
- **Status:** trạng thái của node, gồm 2 trạng thái là Powered On (đang hoạt động) và **Terminated** (đã ngừng hoạt động và không thể khôi phục)
- **IP Private:** địa chỉ IP riêng tư của Node

- **Lifetime:** thời gian node hoạt động giúp KH theo dõi được thông tin thời gian để tính cước PAYG

The screenshot shows the Viettel Cloud console interface. The left sidebar contains navigation options like Container, Clusters, Instance, Virtual Machines, Storage, Volumes, Network & Security, Subnets, Network Interfaces, Security Groups, SSH Keys, Elastic IPs, NAT Gateways, and Route Tables. The main area displays the 'Clusters' page with a search bar and a table of clusters. The 'Compute' tab is selected, showing 'Node Groups (2)' and 'Nodes (17)'. The 'Nodes' table has columns: Node Name, Type, Resource Type, Node Group, Start Time, End Time, Status, IP Private, and Lifetime. A red box highlights the 'Lifetime' column header.

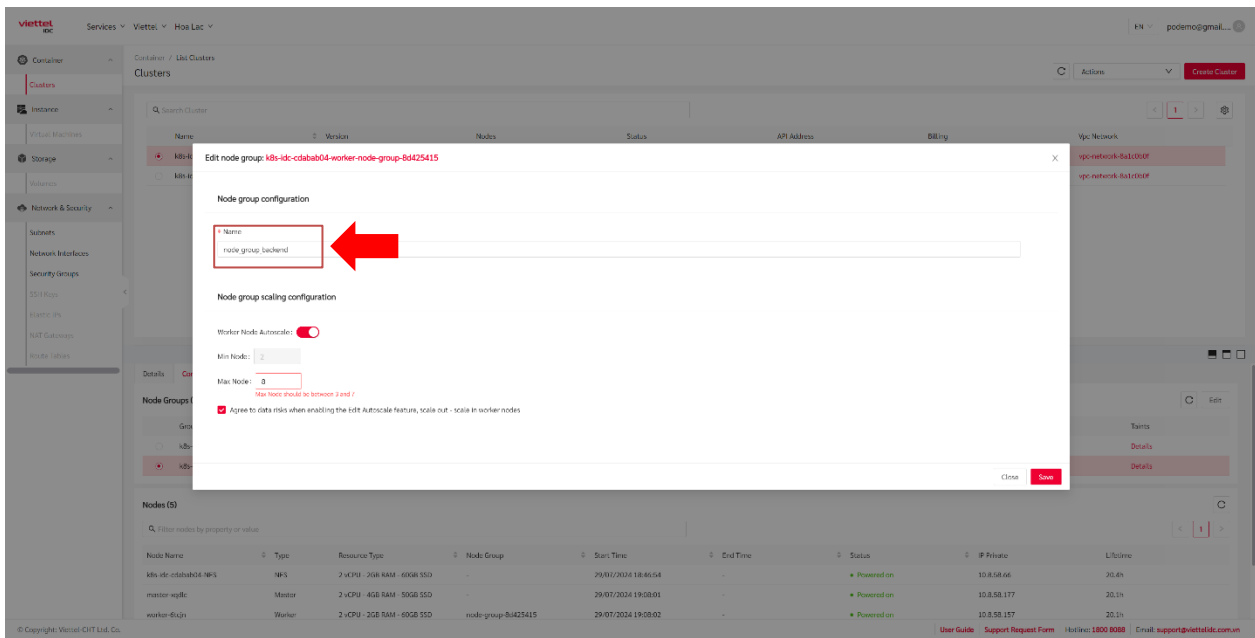
4.3 Sửa thông tin Node group

4.3.1 Sửa tên Node group

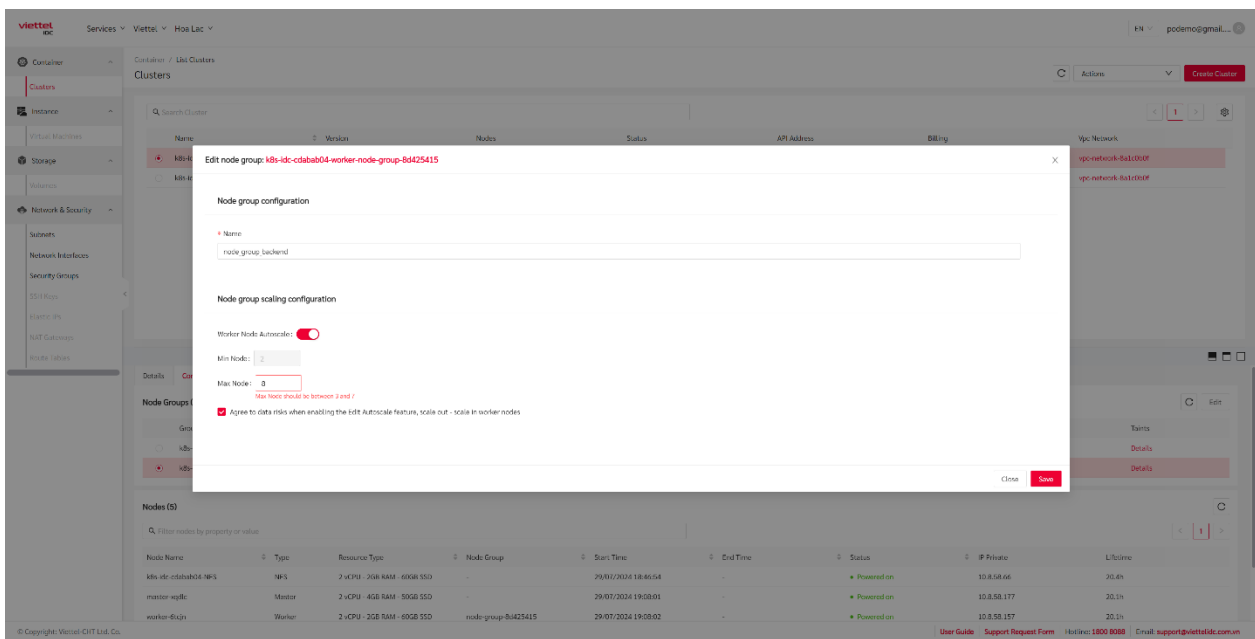
Bước 1: Người dùng chọn cluster, chọn Node group cần sửa tên và Click **Edit**

The screenshot shows the Viettel Cloud console interface. The left sidebar contains navigation options like Container, Clusters, Instance, Virtual Machines, Storage, Volumes, Network & Security, Subnets, Network Interfaces, Security Groups, SSH Keys, Elastic IPs, NAT Gateways, and Route Tables. The main area displays the 'Clusters' page with a search bar and a table of clusters. The 'Compute' tab is selected, showing 'Node Groups (2)' and 'Nodes (5)'. The 'Node Groups' table has columns: Group Name, Resource Type, Auto Scale, Min Node, Max Node, Status, Labels, and Taints. A red arrow points to the 'Edit' button in the top right corner of the Node Groups table.

Bước 2: Người dùng có thể update tên theo mong muốn, Click **Save** để lưu thông tin



4.3.2 Sửa cấu hình Worker Autoscale theo Node Group



Người dùng có thể cấu hình tính năng Worker Autoscale theo từng Node Group để đáp ứng nhu cầu về tài nguyên. Hiện tại Viettel IDC đang hỗ trợ tính năng Autoscale với cả 2 hình thức Saving Plan và PAYG, cụ thể:

- Với hình thức Saving Plan, Người dùng có thể cấu hình tối đa thêm 5 Worker Node so với số Worker Node tối thiểu người dùng đã mua cho 1 Node Group

- Với hình thức PAYG, Người dùng có thể cấu hình linh hoạt Autoscale lên tới 10 Worker Node để đáp ứng nhu cầu tăng giảm tài nguyên Worker để đáp ứng nhu cầu tài nguyên cho ứng dụng.

Bước 1: Người dùng chọn cluster, chọn Node group cần sửa tên và Click **Edit**

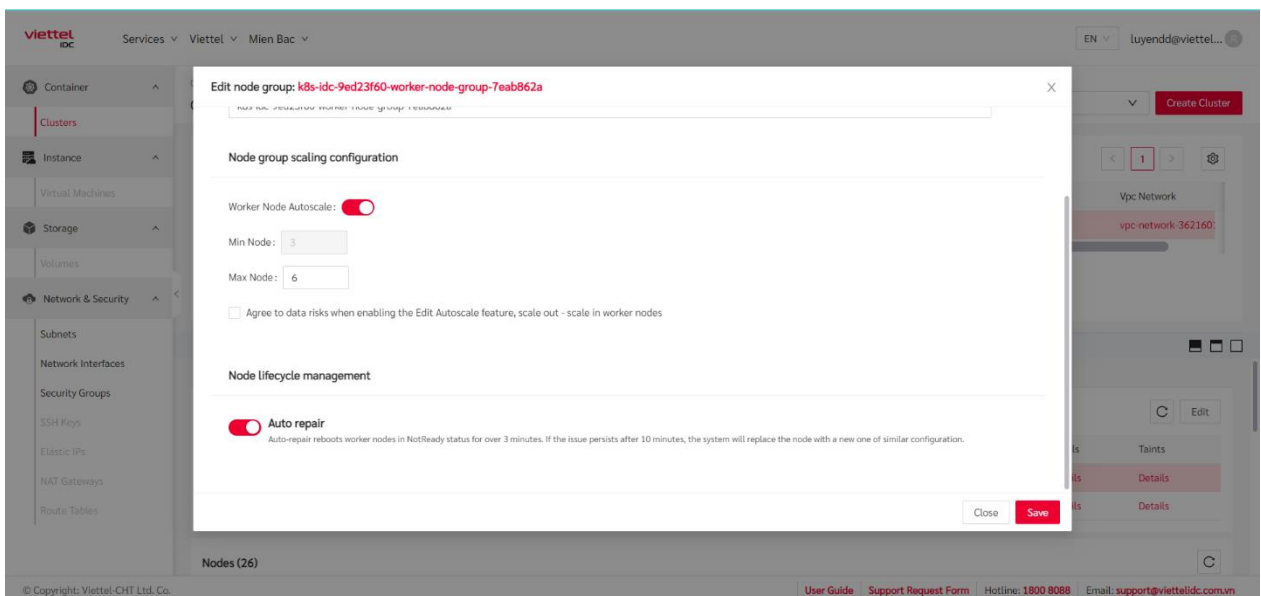
Bước 2: Click **Enable** và điền số lượng Min Node, Max node để đáp ứng ngưỡng tài nguyên cho Cluster. Min node là số lượng node còn lại khi scale in (giảm node); Max node là số lượng node tối đa có thể scale out (tăng node)

Bước 3: Click **Agree to data risks when enabling the Edit Worker Autoscale feature, scale out - scale in worker nodes** để đồng ý với rủi ro về dữ liệu khi thực hiện scale out – scale in

Bước 4: Click **Update** để cập nhật config cho tính năng Worker Autoscale.

Khi cluster của KH vận hành đến ngưỡng và cần các pod mới để chạy dịch vụ sẽ sinh ra Worker Node mới với số node nằm trong khoảng từ min node đến max node. Nếu số node tăng đến max node mà dịch vụ vẫn cần pod, hệ thống sẽ không sinh ra thêm Worker Node mà Người dùng cần update config để tiếp tục có thêm Worker Node cho dịch vụ.

4.3.3 Sửa cấu hình auto-repair theo Node Group



Bên cạnh tính năng Worker Autoscale, dịch vụ vOKS của Viettel IDC cung cấp tính năng

Auto-repair theo Node group mang lại nhiều lợi ích quan trọng, giúp đảm bảo tính sẵn sàng và độ tin cậy cao cho ứng dụng. Tính năng Auto-repair có những điểm nổi bật sau:

- Tự động phát hiện lỗi: Kubernetes có thể tự động phát hiện các node bị lỗi hoặc gặp sự cố thông qua việc theo dõi trạng thái của các node. Một số dấu hiệu cho thấy node bị lỗi bao gồm: node báo cáo trạng thái "NotReady", node không thể ping được, node bị lỗi phần cứng, v.v.
- Tự động khởi động lại node: Khi một node bị phát hiện lỗi, Kubernetes sẽ tự động khởi động lại node. Việc khởi động lại node có thể giúp khắc phục các lỗi tạm thời và đưa node trở lại trạng thái hoạt động bình thường.
- Giảm thiểu sự can thiệp thủ công: Auto-repair giúp giảm thiểu sự can thiệp thủ công của người quản trị viên hệ thống, tiết kiệm thời gian và công sức.
- Cải thiện hiệu quả hoạt động: Auto Healing giúp cải thiện hiệu quả hoạt động của hệ thống bằng cách đảm bảo rằng các node luôn hoạt động bình thường.

Nếu sau quá trình auto repair mà node không thể trở lại trạng thái Ready thì sau 10 phút hệ thống sẽ thay thế node notReady đó bằng một node mới có cấu hình tương tự. Tính năng này mặc định disable cho tất cả các Node group. Người dùng có thể lựa chọn enable hoặc disable tính năng này cho các worker group khác trong cluster.

Bước 1: Người dùng chọn cluster, chọn Node group cần đổi trạng thái và Click **Edit**

Bước 2: Click **Enable** để bật tính năng auto-repair cho Node group, **Disable** để tắt tính năng auto-repair cho Node group.

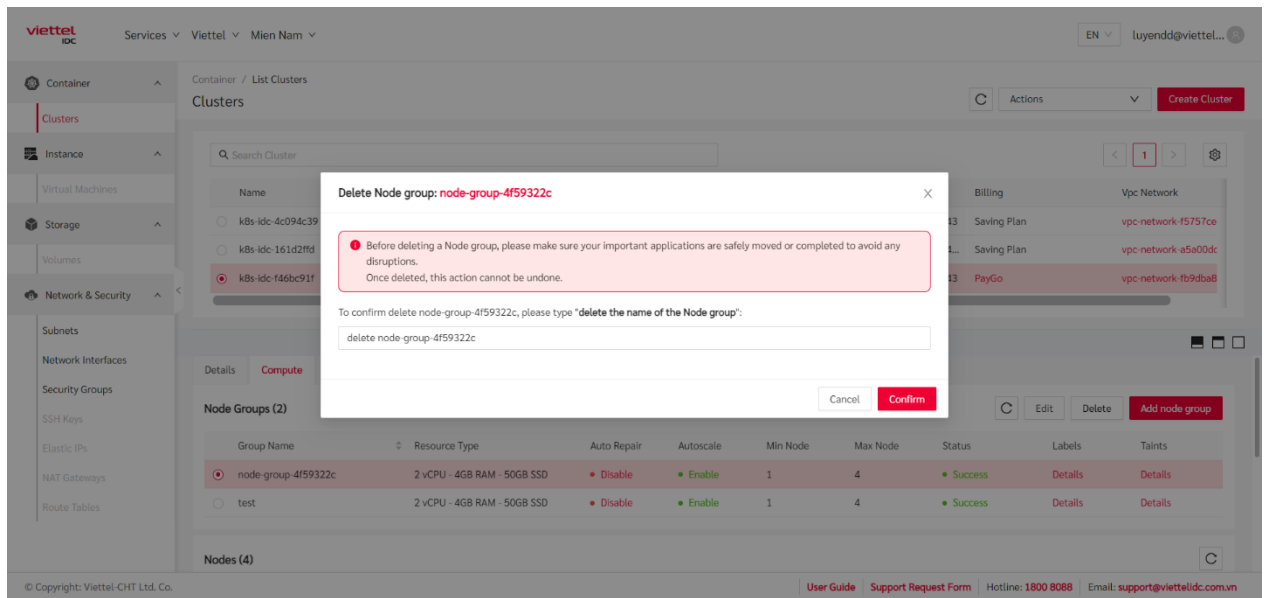
Bước 3: Click **Update** để cập nhật config cho tính năng auto-repair.

4.4 Xóa Node group

Với gói cước sử dụng hình thức thanh toán PAYG, người dùng có thể dễ dàng điều chỉnh số lượng các Node group có trong cụm thông qua chức năng xóa node group, cụ thể:

Bước 1: Người dùng chọn cluster, chọn Node group cần sửa tên và Click **Delete**

Bước 2: Người dùng nhập câu lệnh theo cú pháp **delete + tên Node group**, click confirm để xóa node group



Lưu ý:

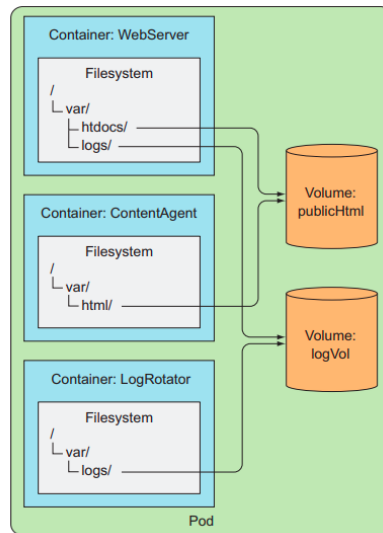
- Cluster phải có tối thiểu 1 node group sau khi xóa. Để xóa toàn bộ các node group, [người dùng buộc phải xóa cluster thông qua luồng ticket](#)
- Trước khi xóa nhóm Node, người dùng cần đảm bảo các ứng dụng quan trọng được di chuyển hoặc hoàn tất an toàn để tránh sự gián đoạn về dịch vụ.
- Sau khi xóa, hành động này không thể hoàn tác.

5. Lưu trữ dữ liệu ứng dụng

5.1 Giới thiệu Volume & NFS Node

Hệ thống Kubernetes sẽ tạo ra Pods mới thay thế khi một Pods bị lỗi, chết hay crash. Vậy dữ liệu được lưu trong Pods cũ sẽ ra sao? Pods mới có lấy lại được dữ liệu của Pods cũ đã mất để tiếp tục sử dụng không? [Volumes](#) sẽ giúp giải quyết các vấn đề trên.

[Volumes](#) là thành phần trực thuộc Pods. Volumes được định nghĩa trong cấu hình file .yaml khi khởi tạo các Pods. Các container có thể thực hiện mount dữ liệu bên trong container đến đối tượng volumes thuộc cùng Pods.



Hình: Các containers trong Pod mount đến 2 volumes để share data với nhau

Các loại volumes:

- EmptyDir
- HostPath
- GitRepo
- NFS
- Cloud volumes: gcePersistentDisk (Google Compute Engine Persistent Disk), awsElasticBlockStore (Amazon Web Services Elastic Block Store Volume), azureDisk (Microsoft Azure Disk Volume)
- Cinder, cephfs, iscsi, flocker, glusterfs, quobyte, rbd, flexVolume, sphereVolume, photonPersistentDisk, scaleIO
- ConfigMap, secret, downwardAPI
- PersistentVolumeClaim

Như đã thấy, có rất nhiều loại volumes khác nhau. Tuy nhiên, với dịch vụ vOKS tập trung vào cung cấp giải pháp Persistent Volume (PV) – NFS node.

vOKS có thể tự động thêm các node NFS vào cấu hình Kubernetes khi tạo một cụm mới. Quản trị viên có thể sử dụng (các) NFS node để triển khai lưu trữ giữa nhiều container, cho phép triển khai các ứng dụng dạng stateful tốt hơn khi dữ liệu luôn được đảm bảo lưu trữ

ở phân vùng cố định.

Phần này sẽ mô tả việc quản lý NFS node và một ví dụ hướng dẫn thiết lập lưu trữ trên NFS, yêu cầu khách hàng cần nắm được lý thuyết cơ bản về lưu trữ Kubernetes cũng như quản trị NFS.

5.2 Các dạng kết nối lưu trữ trên NFS node

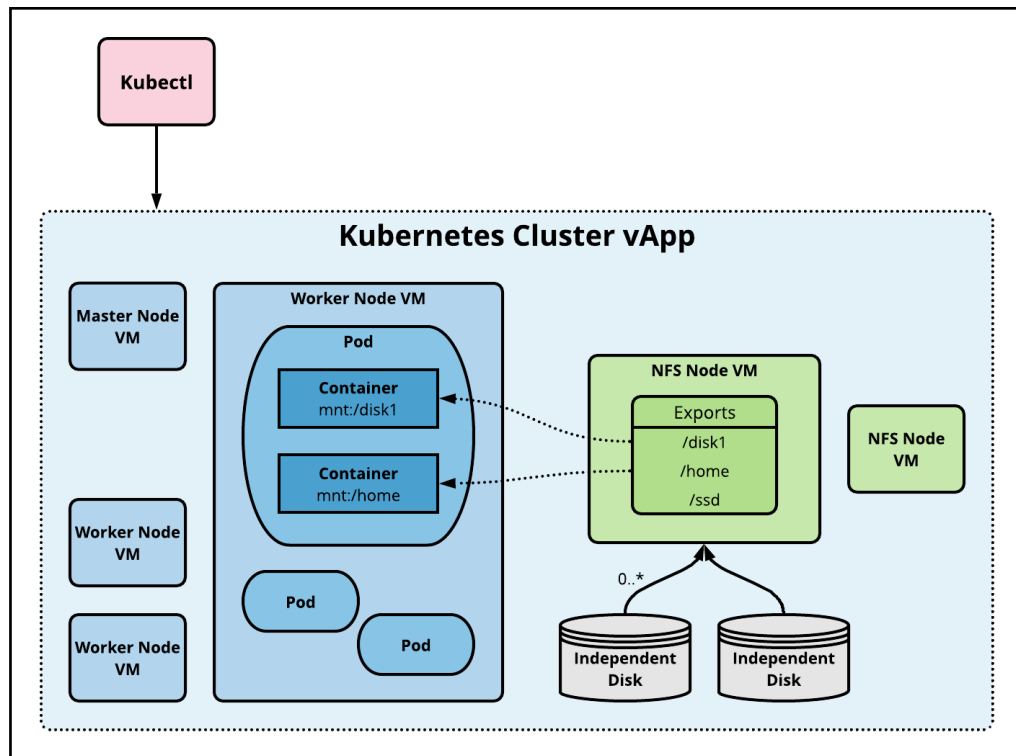
Lưu trữ tĩnh được quản trị viên cluster cấp phát trước. Chúng mang thông tin chi tiết về dung lượng lưu trữ thực tế có sẵn để người dùng sử dụng. Chúng tồn tại trong Kubernetes API và sẵn sàng để sử dụng. Người dùng có thể phân bổ dung lượng lưu trữ tĩnh bằng cách tạo một xác nhận dung lượng lưu trữ cho cùng một phân vùng lưu trữ hoặc tùy biến theo yêu cầu sử dụng trên các container. vOKS hỗ trợ dung lượng lưu trữ tĩnh cấp phát trước trên node NFS.

Lưu trữ động không được quản trị viên cluster cấp phép trước. Hiện tại, vOKS chưa hỗ trợ cung cấp dung lượng lưu trữ liên tục động.

5.3 Kiến trúc NFS node

NFS Node cho phép chia sẻ một hoặc nhiều phân vùng lưu trữ trên một hoặc nhiều cluster node khác nhau. Khi (các) cluster bị xóa, dữ liệu lưu trữ trên NFS được giữ nguyên không bị mất và các kết nối từ container chỉ bị ngắt kết nối vào phân vùng lưu trữ. Điều này có nghĩa là dữ liệu lưu trữ từ NFS node có thể được chia sẻ, sử dụng đồng thời giữa các cluster khác nhau ngay cả khi có một node nào đó trong cluster bị xóa hoặc lỗi. Bên cạnh đó, NFS có thể phục vụ cho nhiều nguồn ghi đồng thời.

Sơ đồ kiến trúc triển khai sử dụng NFS node:

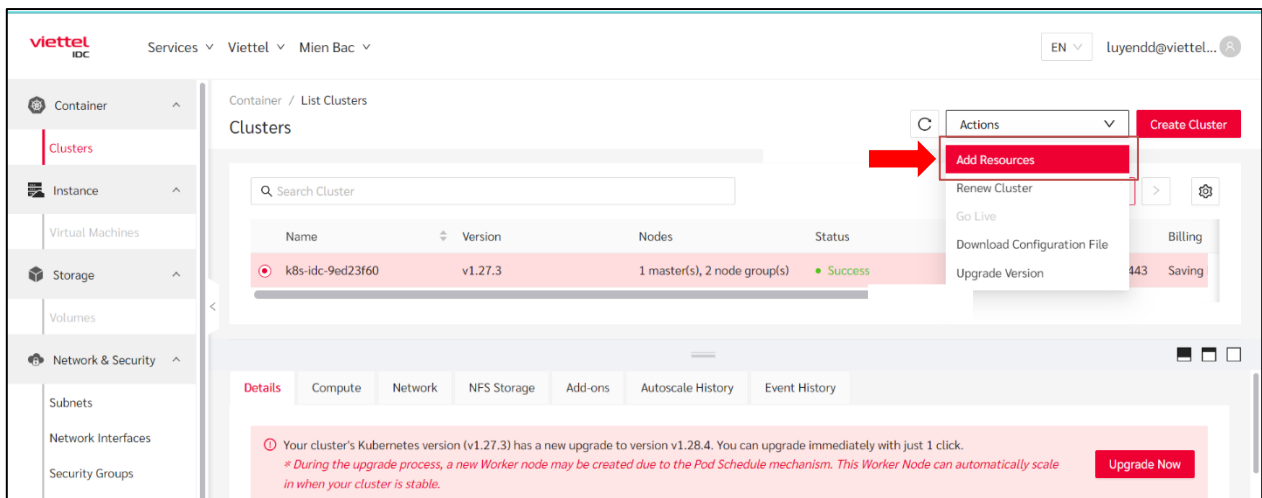


5.4 Thêm dung lượng NFS Node

Người dùng có thể dễ dàng bổ sung dung lượng cho NFS Node trên giao diện quản trị của Viettel IDC phát triển

5.4.1 Thêm dung lượng NFS Node cho cluster Saving Plan

Bước 1: Trên giao diện CMP, Người dùng click chọn Cluster cần mua bổ sung → chọn Action Add Resources



Bước 2: Hệ thống chuyển sang màn hình **Mua bổ sung:**

- Người dùng lựa chọn **Persistent Volume** để mua thêm dung lượng NFS
- Nhập/kéo thả dung lượng cần mua bổ sung
- Click chọn **Mua ngay**.

The screenshot displays the Viettel IDC 'Mua bổ sung' (Buy additional) interface. The main configuration area shows two node groups: 'T1.vOKS 1 (2 vCPU, 4 GB RAM, 50 GB SSD)' and 'Worker Node Group 2'. Under 'Worker Node Group 2', there is a section for 'Chọn tài nguyên, dịch vụ bổ sung' (Select additional resources/services). A red arrow points to the 'Persistent Volume' checkbox, which is checked. Below this, a slider indicates the volume size, ranging from 10GB to 1000GB, with a value of 24 selected. The price for 24 units is 23,760 VND. The right sidebar shows a summary of the order, including the total price of 26,136 VND and a 'MUA NGAY' (Buy Now) button.

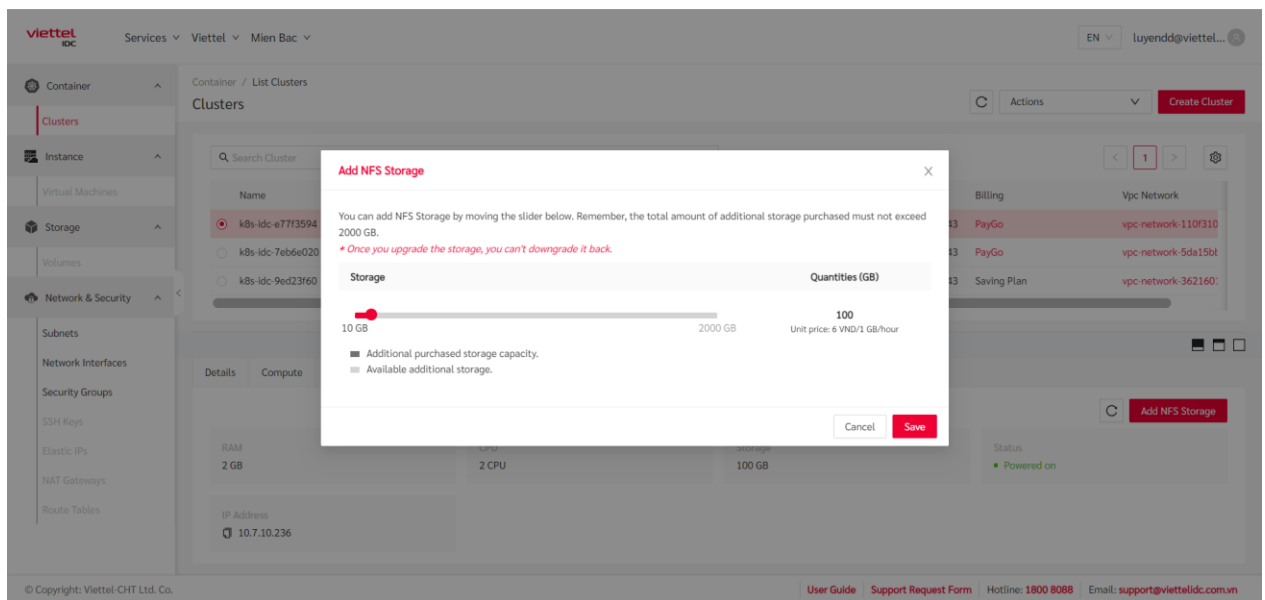
Bước 3: Xác nhận đơn hàng: *Thao tác tương tự như bước [tạo thêm node group](#)*

Bước 4: Thanh toán đơn hàng: *Thao tác tương tự như bước [tạo thêm node group](#)*

5.4.2 Thêm dung lượng NFS Node cho cluster PAYG

Với nhu cầu cần sử dụng thêm dung lượng NFS Node, khách hàng thực hiện như sau:

Bước 1: Trên màn hình **List Clusters**, Chọn cluster cần thêm dung lượng, chọn tab **NFS Storage**



Bước 2: Người dùng kéo thả dung lượng NFS Storage cần mua thêm cho cụm. Thông tin sử dụng tài nguyên bổ sung sẽ hiển thị tại [màn hình quản lý cước](#)

5.5 Hướng dẫn cấu hình sử dụng NFS node

* Tạo NFS Node: Hệ thống sẽ tự động tạo NFS node, format file system ổ đĩa theo định dạng ext4, khởi tạo thư mục chia sẻ mount point mặc định và thiết lập cơ chế tự động mount thư mục chia sẻ vào file /etc/fstab trong trường hợp NFS node khởi động lại.

* Cấu hình sử dụng phân vùng lưu trữ trên Kubernetes:

Hiện tại NFS Node được cung cấp 2 phân vùng volume (root disk và shared data storage). Thư mục /data/mydata (mặc định) sẽ được mount sẵn với phân vùng volume shared data và được chia sẻ chung giữa các NFS clients.

Lưu ý: tên đường dẫn /data/mydata khi tạo các persistent volume phải ghi chính xác để đảm bảo persistent volume sử dụng được tài nguyên được cấp phát trong NFS node, nếu không sẽ gặp lỗi khi Kubernetes cố gắng sử dụng các node trong cluster.

- **Bước 1:** Cài đặt helm-chart csi-driver-nfs từ phía client:

```
helm repo add csi-driver-nfs https://raw.githubusercontent.com/kubernetes-csi/csi-driver-nfs/master/charts
```

```
helm install csi-driver-nfs csi-driver-nfs/csi-driver-nfs --namespace kube-system --
version v4.6.0
```

- **Bước 2:** Cài đặt StorageClass mặc định

File cấu hình StorageClass (default-sc.yaml):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-driver-nfs-default-sc
  annotations:
    storageclass.kubernetes.io/is-default-class: 'true'
provisioner: nfs.csi.k8s.io
parameters:
  server: 10.14.53.175 # IP of nfs server
  share: /data/mydata
reclaimPolicy: Delete
mountOptions:
  - nfsvers=4.1
volumeBindingMode: Immediate
```

Khởi tạo StorageClass:

```
kubectl create -f default-sc.yaml
```

Kiểm tra dịch vụ:

```
kubectl get sc -A -owide
kubectl get po -A
```

6. Thiết lập mạng

6.1 Kết nối private với các dịch vụ khác trong cùng 1 VPC Network

Khi lựa chọn tạo mới VPC Network hoặc chọn VPC Network đã tạo cho cluster ở [bước khởi tạo cluster](#), Người dùng có thể dễ dàng kết nối riêng tư với các dịch vụ khác trong cùng hệ sinh thái sản phẩm của Viettel IDC như vOKS, vOVPC với những tính năng nổi bật như

bằng thông tốc độ cao, bảo mật dữ liệu và kết nối, giúp hệ thống công nghệ thông tin được đảm bảo an toàn thông tin & các tiêu chuẩn bảo mật cao nhất.

6.2 Kết nối dịch vụ Viettel Open Virtual Private Cloud (vOVPC)

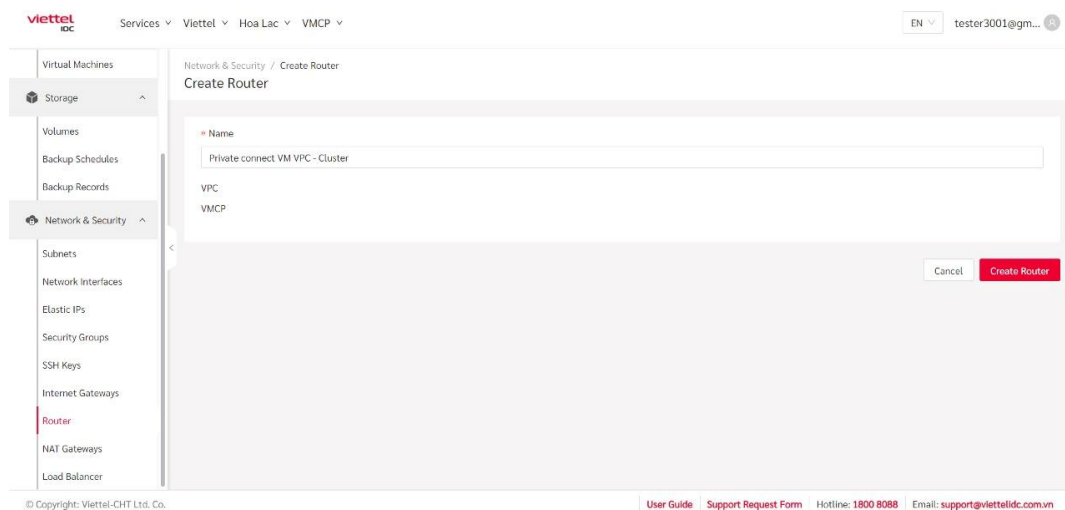
Trường hợp 1: VM dịch vụ vOVPC sử dụng Public Subnet

Khi đó, cluster K8s của dịch vụ vOKS & VM của dịch vụ vOVPC sẽ mặc định kết nối với nhau.

Trường hợp 2: VM dịch vụ vOVPC sử dụng Private Subnet để tương tác với cluster K8s của dịch vụ vOKS, Người dùng cần thực hiện theo các bước sau:

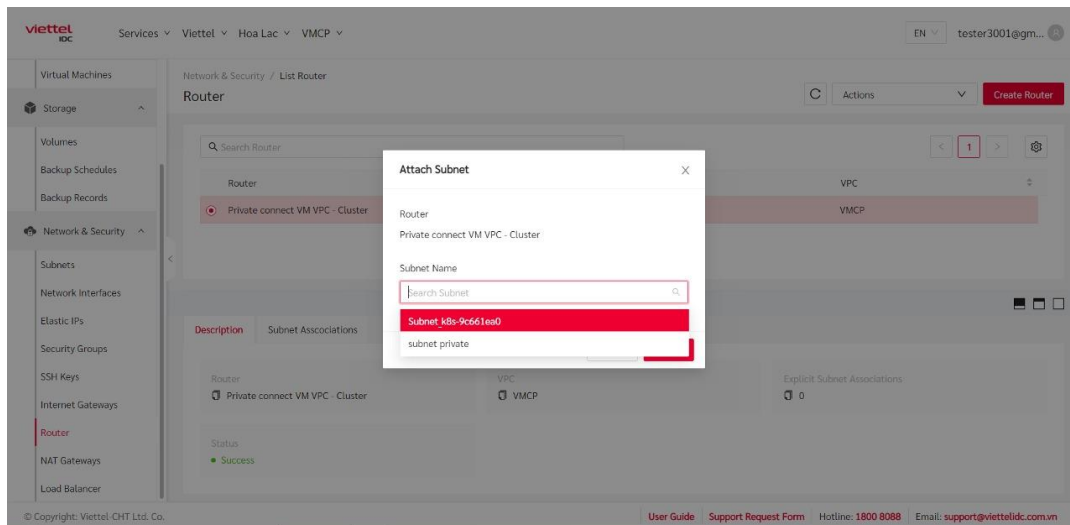
Bước 1: Tạo router trên giao diện quản trị Network & Security

Người dùng vào trang quản trị dịch vụ vOVPC -> tab **Network & Security** -> **Router** -> chọn **Create Router**, nhập tên Router



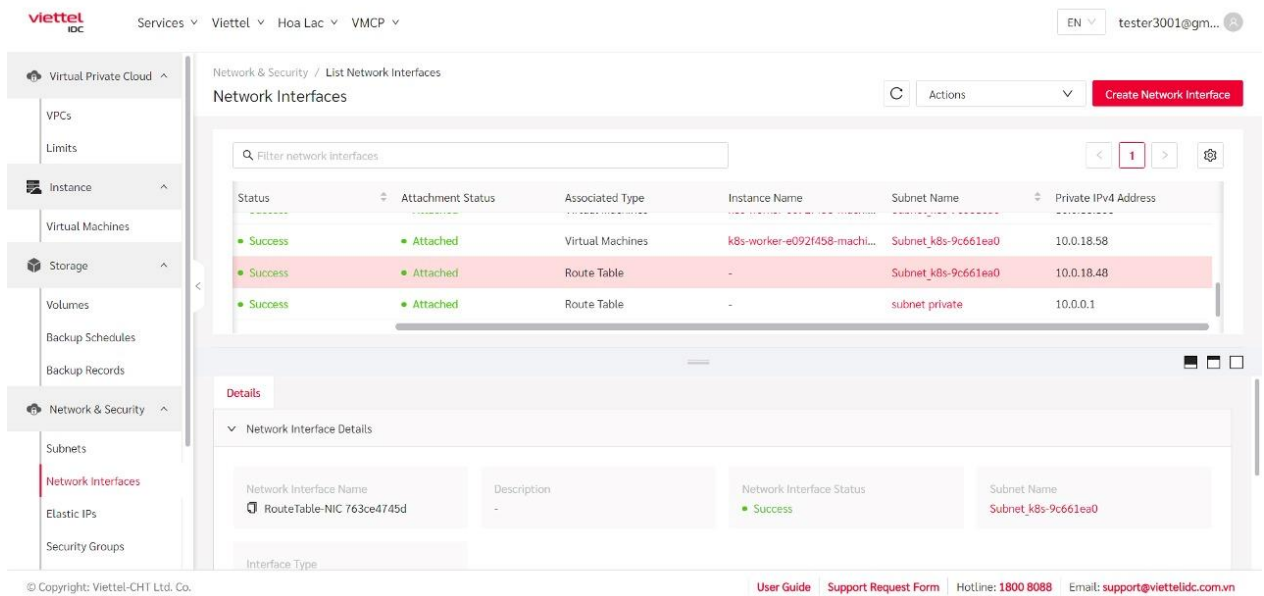
Bước 2: Attach 2 Subnet của VM (X.X.X.0/24) và của cluster vOKS (Y.Y.Y.0/24) vào router

Người dùng lựa chọn Router vừa tạo -> click **Actions** -> **Attach Subnet** -> lần lượt lựa chọn **2 subnet** tương ứng của VM & Cluster K8s



Bước 3: Lấy thông tin **2 IPv4 Address** của 2 Network Interface có type là Route Table tương ứng với 2 subnet

Người dùng vào trang quản trị dịch vụ vOVPC -> tab **Network & Security** -> **Network Interface** -> xem 2 Network Interface có type là **Route Table**, tương ứng với 2 subnet



VD: IPv4 Address của Subnet VM là 10.0.0.1

IPv4 Address của Cluster K8s là 10.0.18.48

Bước 4: Thêm route cho VM vOVPC

KH truy cập vào VM vOVPC muốn kết nối với cluster k8s, thêm route:

ip r add Y.Y.Y.0/24 via x.x.x.x

Bước 5: Thêm route cho cluster K8s, liên hệ Viettel IDC qua email support@viettelidc.com.vn để được hỗ trợ cấu hình network bên trong các node

6.3 Kết nối với dịch vụ Viettel Database Service (vDBS)

Khi khởi tạo dịch vụ vDBS để kết nối cùng dịch vụ vOKS, Người dùng có thể lựa chọn tạo cùng VPC Network để dễ dàng kết nối private giữa 2 dịch vụ. Dịch vụ vDBS khi đăng ký sẽ tạo ở 1 subnet private để đảm bảo mức độ an toàn cao nhất cho máy chủ dữ liệu. Người dùng cần thực hiện theo các bước sau

Bước 1: Xem thông tin Subnet của dịch vụ vOKS

- Với dịch vụ vOKS, Người dùng xem ở tab **Network & Security** -> **Subnet** -> Xem tên **Subnet** của Cluster tương ứng

The screenshot shows the Viettel IDC management console. On the left sidebar, under 'Network & Security', the 'Subnets' tab is selected. The main area displays a table of subnets. One subnet is listed: 'Subnet k8s-7679a733' with a status of 'Success', IPv4 CIDR '10.2.26.0/24', VPC Network Name 'vpc-network-d3c8776e', and Cluster Name '-'. Below the table, the 'Details' section shows the same information for the selected subnet, including 'Public Subnet: Yes'.

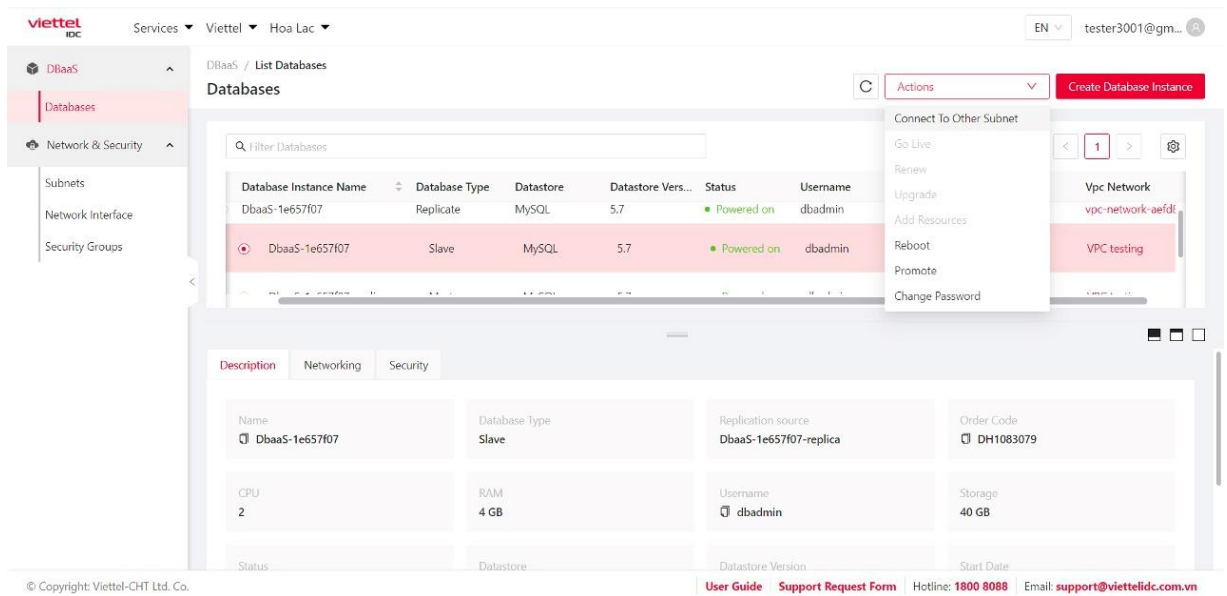
Subnet Name	Status	IPv4 CIDR	VPC Network Name	Description	Cluster Name
Subnet k8s-7679a733	Success	10.2.26.0/24	vpc-network-d3c8776e	-	-

Details:

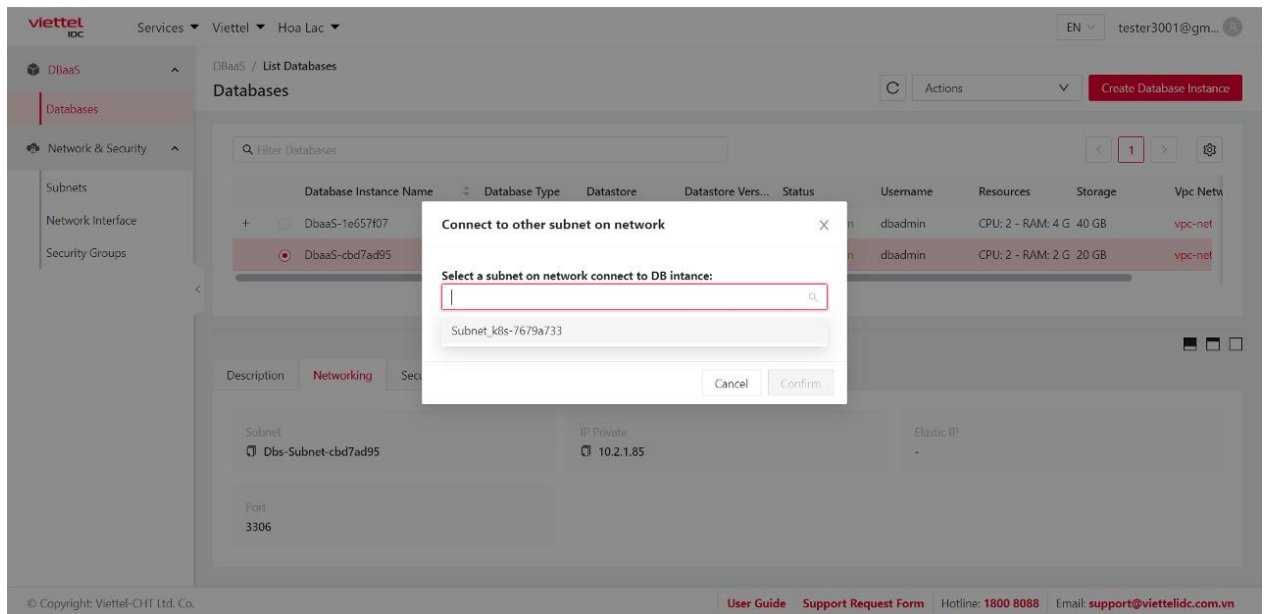
Cluster Name	Status	IPv4 CIDR	Description
-	Success	10.2.26.0/24	-
Public Subnet	Yes		

© Copyright: Viettel-CHT Ltd. Co. | User Guide | Support Request Form | Hotline: 1800 8088 | Email: support@viettelidc.com.vn

Bước 2: Trên màn hình quản trị của dịch vụ vDBS, Người dùng tìm tới ô **Actions**, click hiển thị menu -> Click lựa chọn **Connect To Other Subnet**



Bước 3: Ở cửa sổ lựa chọn subnet, Người dùng **lựa chọn Subnet** của dịch vụ vOKS cần kết nối tới dịch vụ vOKS, bấm **Update**



Khi đó, hệ thống sẽ tạo 1 Network Interface thuộc dải Subnet của dịch vụ vOKS cần connect đến máy chủ DB của dịch vụ vDBS, gắn vào DB Instance, tạo Security Groups rule cho phép dải subnet vOKS kết nối theo cổng của database.

6.4. CNI

6.4.1. Khái niệm

[CNI \(Container Network Interface\)](#) là một bộ công cụ tiêu chuẩn cung cấp khả năng kết nối mạng cho các container trong một cụm Kubernetes. Nói một cách đơn giản, CNI là một lớp trừu tượng, giúp Kubernetes quản lý và cấu hình mạng cho các pod (một tập hợp các container chia sẻ cùng một mạng) một cách linh hoạt và hiệu quả.

6.4.2. Các Plugin CNI phổ biến

Có nhiều plug-in phổ biến được sử dụng, nhưng phổ biến nhất hiện tại ở Viettel IDC có cung cấp là:

- **Calico:**
 - **Calico** là giải pháp mạng và bảo mật cho phép các workload trên Kubernetes và cả các workload không thuộc Kubernetes hoặc legacy giao tiếp một cách liền mạch và an toàn.
 - Sử dụng routing trực tiếp mà không cần overlay, dựa vào BGP để tối ưu hiệu năng.
 - Hỗ trợ Network Policies để kiểm soát lưu lượng giữa các Pod.
 - Nên sử dụng khi không yêu cầu hiệu suất quá cao.
- **Cilium:**
 - **Cilium** là phần mềm mã nguồn mở giúp bảo mật kết nối mạng giữa các dịch vụ ứng dụng được triển khai trên các nền tảng quản lý container Linux như **Docker** và **Kubernetes** một cách minh bạch.
 - Dựa trên eBPF để tối ưu việc truyền dữ liệu và bảo mật.
 - Đơn giản, không yêu cầu hiệu suất quá cao tuy nhiên có nhu cầu monitor chuyên sâu (Hubble).

6.5. Cấu hình DNS của Viettel IDC

6.5.1. Giới thiệu

Tài liệu này hướng dẫn cách cấu hình thêm DNS trên cụm Kubernetes mà dịch vụ vOKS của Viettel IDC đang cung cấp. Việc cấu hình thêm các DNS Viettel IDC giúp hệ thống có thể phân giải tên miền một cách nhanh chóng và chính xác, đảm bảo kết nối ổn định đến các

dịch vụ bên ngoài.

6.5.2. Lý do sử dụng DNS của Viettel IDC

Mặc định, CoreDNS có thể sử dụng các máy chủ DNS công cộng 8.8.8.8 và 8.8.4.4 của Google. Tuy nhiên, việc sử dụng các DNS quốc tế này có thể gặp một số vấn đề:

- Phụ thuộc vào đường truyền quốc tế, có thể ảnh hưởng đến tốc độ truy vấn khi xảy ra đứt cáp hoặc nghẽn mạng.
- Độ trễ cao hơn so với các DNS nội địa, do các truy vấn phải đi qua nhiều tuyến mạng hơn.

Để đảm bảo tốc độ và độ ổn định, Viettel IDC cung cấp thêm các DNS (115.84.181.20, 115.84.177.20) giúp:

- Cải thiện hiệu suất khi truy vấn DNS trong nước.
- Giảm độ trễ do sử dụng hệ thống máy chủ DNS gần hơn về mặt địa lý.
- Hạn chế sự cố mạng quốc tế, đảm bảo hệ thống hoạt động mượt mà hơn.

6.5.3. Các bước thực hiện

Bước 1: Tạo file yaml Config Map mới config-coredns.yaml với nội dung:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
data:
  Corefile: |
    .:53 {
      errors
      health {
        lameduck 5s
      }
      ready
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        fallthrough in-addr.arpa ip6.arpa
        ttl 30
      }
      prometheus :9153
      forward . 115.84.181.20 115.84.177.20 /etc/resolv.conf {
        max_concurrent 1000
      }
    }
```

```

}
cache 30
loop
reload
loadbalance
}

```

Giải thích: Bổ sung thông tin 2 DNS của Viettl IDC 115.84.181.20, 115.84.177.20 vào cấu hình forward của cấu hình coredns

Bước 2: Cập nhật config mới

Lệnh: `kubectl apply -f config-coredns.yaml`

Bước 3: Restart lại các pod coredns để nhận cấu hình mới

Lệnh: `kubectl rollout restart deployment coredns -n kube-system`

7. Triển khai ứng dụng trên cụm

7.1 Cài đặt môi trường điều khiển phía người dùng

Công cụ command-line trong Kubernetes - kubectl, cho phép người dùng thực thi các câu lệnh điều khiển Kubernetes clusters. Kubectl hỗ trợ thực hiện triển khai các ứng dụng, theo dõi và quản lý tài nguyên, xem log hệ thống.

Để sử dụng, người dùng có thể cài đặt trên các môi trường sử dụng khác nhau, chi tiết xem hướng dẫn cài đặt trên từng môi trường tương ứng bên dưới:

*Cài đặt kubectl trên môi trường Windows

- **Bước 1:** Download kubectl từ link cài đặt:

<https://storage.googleapis.com/kubernetes-release/release/v1.19.0/bin/windows/amd64/kubectl.exe>

Để kiểm tra version kubectl mới nhất, truy cập vào link:

<https://storage.googleapis.com/kubernetes-release/release/stable.txt>

- **Bước 2:** Thêm đường dẫn file kubectl vào PATH:

Tạo folder .kube trong thư mục User đang sử dụng.

Lưu ý: User đăng nhập phải có quyền administrator local.

Cách 1: đăng nhập tài khoản Windows với quyền administrator local.

Nhấn tổ hợp phím Windows + R, gõ %userprofile%

Tạo folder .kube tại đây.

Cách 2: đăng nhập tài khoản Windows với quyền administrator local

Mở Command Prompt, gõ câu lệnh: mkdir .kube

Đặt file kubectl.exe vào đường dẫn %userprofile%/.kube

- **Bước 3:** Kiểm tra phiên bản kubectl đã cài:

```
kubectl version --client
```

```
kubectl cluster-info
```

```
PS C:\Users\QuangDA\.kube> .\kubectl.exe cluster-info
Kubernetes master is running at https://171.244.49.28:6443
KubeDNS is running at https://171.244.49.28:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS C:\Users\QuangDA\.kube> .\kubectl.exe get nodes
NAME          STATUS    ROLES    AGE   VERSION
mstr-obcy     Ready    master   3h56m v1.17.2
node-gd6c     Ready    <none>   3h47m v1.17.2
node-i8vk     Ready    <none>   3h46m v1.17.2
node-ldd5     Ready    <none>   3h48m v1.17.2
PS C:\Users\QuangDA\.kube>
```

- **Bước 4:** Kết nối thành công

Thực hiện các câu lệnh điều khiển kubectl

*Cài đặt kubectl trên môi trường Centos Linux

- **Bước 1:** Chạy lệnh sau:

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
```

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
```

```
enabled=1 gpgcheck=1
```

```
repo_gpgcheck=1
```

```
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
```

```
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

```
EOF
```

- **Bước 2:** Cài đặt package kubectl:

```
yum install -y kubectl
```

- **Bước 3:** Tạo directory chứa file config:

```
mkdir ~/.kube
```

- **Bước 4:** Kiểm tra file cấu hình trên server

```
kubectl cluster-info
```

Kết quả:

Kubernetes master is running at https://master_ip:6443

KubeDNS is running at https://master_ip/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

- **Bước 5:** Kết nối thành công

Bắt đầu thực hiện các lệnh điều khiển bằng kubectl

*Cài đặt kubectl trên môi trường MacOS

- **Bước 1:** Tải phiên bản mới nhất bằng lệnh curl

```
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl"
```

Để tải phiên bản người dùng mong muốn, vui lòng thay thế biến `$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)` bằng phiên bản cụ thể. Ví dụ: tải về phiên bản v1.17.0 trên macOS

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.17.0/bin/darwin/amd64/kubectl
```

- **Bước 2:** Gán quyền thực thi kubectl binary

```
chmod +x ./kubectl
```

- **Bước 3:** Đưa bản binary vào biến môi trường PATH

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

- **Bước 4:** Kiểm tra chắc chắn rằng phiên bản người dùng cài là mới nhất

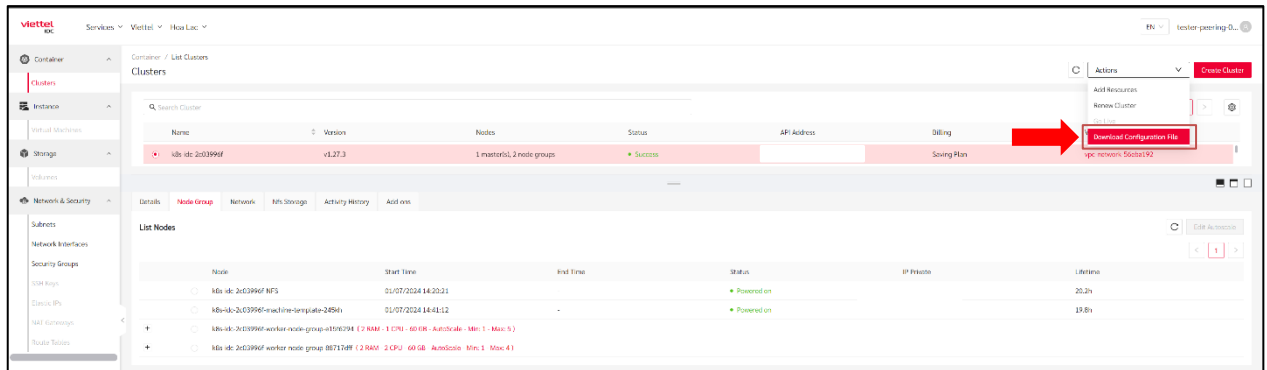
```
kubectl version
```

7.2 Hướng dẫn tải file Kube-config & sử dụng Kubernetes Dashboard

7.2.1 Tải file Kube-config

Trên màn hình thông tin chi tiết 01 gói dịch vụ vOKS, mục “Thao tác gói” có các nút chức năng như sau:

- **Tải file Kube-config:** tải về file để truy cập và cài đặt thông số cho từng cụm K8s;

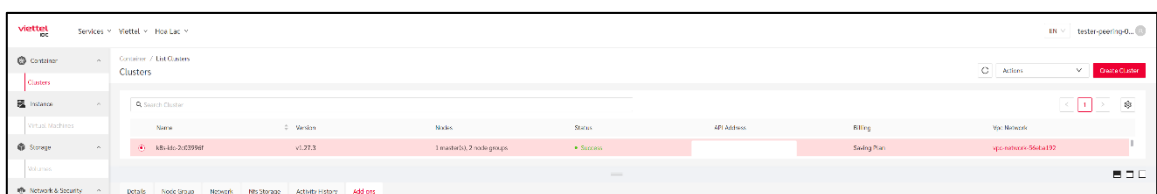


Lưu ý: sau khi download file về máy tính cá nhân. Phải thay đổi tên file đã download: từ *download.config* sang *config*.

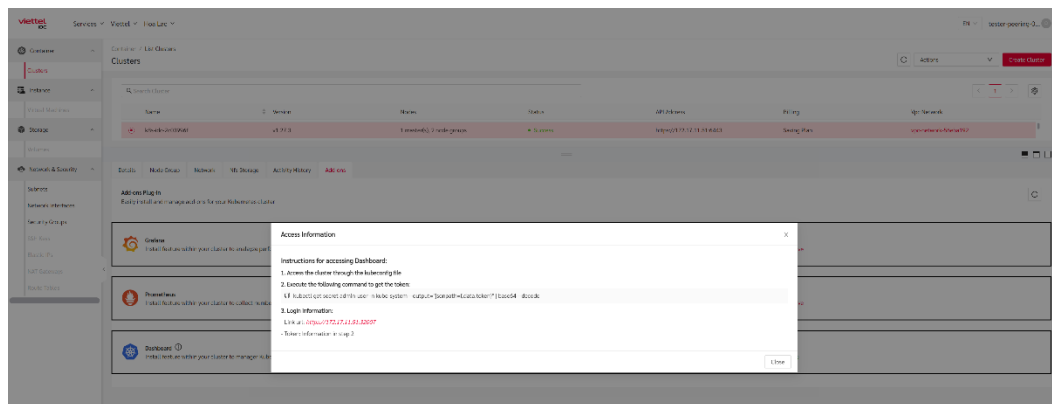
Đặt file config vào tương ứng môi trường cài đặt bên dưới theo đường dẫn:
%userprofile%\kube\config

7.2.2 Sử dụng Kubernetes Dashboard

- Truy cập Kubernetes Dashboard: người dùng có thể truy cập vào trang quản trị kỹ thuật cụm K8s native của Kubernetes bằng cách cài đặt version mong muốn;



Sau khi Kubernetes Dashboard được cài đặt, người dùng có thể xem thông tin truy cập & redirect sang trang quản trị Kubernetes với cluster đang đã cài đặt. Các version của Dashboard sẽ được cập nhật liên tục ở cột version



- Để truy cập trang Dashboard, người dùng cần lấy token theo hướng dẫn sau:

Bước 1: Sử dụng kubectl để lấy token

KH copy file kubeconfig đã tải về máy ở [phần hướng dẫn tải file kube config](#) và file kubectl.exe vào thư mục .kube đã tạo

Bước 2: Lấy token bằng lệnh

```
kubectl describe secrets admin-user -n kube-system
```

```
PS C:\Users\thanhpt> kubectl describe secrets admin-user -n kube-system
Name:         admin-user
Namespace:    kube-system
Labels:       app.kubernetes.io/managed-by=Helm
              helm.toolkit.fluxcd.io/name=dashboard-helm
              helm.toolkit.fluxcd.io/namespace=tp-test-v27-ns
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 7c3ee16-fb42-4dc3-8639-9fd6be7d937a
              meta.helm.sh/release-name: kube-system-dashboard-helm
              meta.helm.sh/release-namespace: kube-system
Type:         kubernetes.io/service-account-token
Data
====
ca.crt:       1070 bytes
namespace:    11 bytes
token:        [REDACTED]
```

Bước 3: Sử dụng token ở bước 2 để truy cập vào Kubernetes Dashboard

Kubernetes Dashboard

☐ Kubeconfig
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

☒ Token
Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

Enter token
.....

SIGN IN

Bước 4: Login vào trang Kubernetes Dashboard

Kubernetes Search + CREATE

Overview

Cluster

Namespaces
Nodes
Persistent Volumes
Roles
Storage Classes

Discovery and Load Balancing

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
✓ kubernetes	component: apiserver provider: kubernetes		kubernetes:443 TCP	-	an hour

Config and Storage

Secrets

Name	Type	Age
default-token-qz7j4	kubernetes.io/service-account-token	an hour

Namespace
default

Overview

Workloads
Cron Jobs

7.3 Tính năng Horizontal Pod Autoscaling

7.3.1 Tổng quan

[Horizontal Pod Autoscaling \(HPA\)](#) là quá trình thực hiện tăng & giảm số lượng pod trong 1 node lên một số lượng đã được định sẵn (mở rộng theo chiều ngang). Quá trình này được thực hiện khi xảy ra một hoặc nhiều sự kiện.

Ví dụ: Trên một Pod, khi CPU sử dụng đạt trên 70%, Memory sử dụng đạt 80% thì sẽ tiến hành tự động sinh thêm Pod mới.

7.3.2 Cơ chế hoạt động

Mặc định cứ mỗi 15s, HPA sẽ lấy thông tin từ metrics-server để so sánh điều kiện.

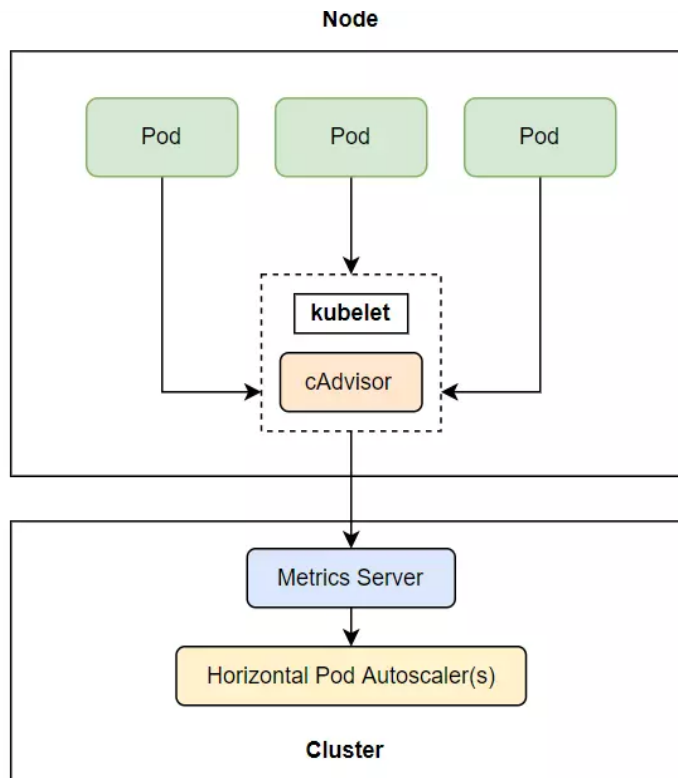
- + Nếu giá trị **currentMetricValue / desiredMetricValue** nhỏ hơn 1 hoặc gần bằng 1 (chênh lệch không đáng kể) thì bỏ qua tiếp tục lắng nghe từ metrics-server;

- + Nếu giá trị trên nhỏ hơn 1 thì sẽ mặc định tiếp tục lắng nghe trong 300s, nếu thông số vẫn nhỏ hơn 1 thì sẽ tiến hành scale down từ từ.

Người dùng có thể tùy chỉnh lại việc scale up/down HPA thông qua config behavior (chỉ áp dụng cho version Kubernetes 1.23 trở lên). Chi tiết cách thức thực hiện tham khảo [tại đây](#)

Ngoài ra, người dùng có thể thiết lập HPA scale up/down theo custom metrics hoặc theo nhiều loại metrics khác nhau (áp dụng cho version Kubernetes 1.23 trở lên). Chi tiết cách thức thực hiện tham khảo [tại đây](#):

Để có thể lấy được thông tin cho việc giám sát (CPU, Memory) làm cơ sở cho việc scale pod mới, đầu tiên khách hàng cần cài đặt **metrics-server**. Cách thức cài đặt metrics-server tham khảo [tại đây](#):



7.3.3 Hướng dẫn cấu hình các metrics HPA

Cách 1: Sử dụng file YAML

Bước 1. Chạy câu lệnh deploy metrics-server

```
Kubectl          apply          -f          https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

Bước 2. Kiểm tra status của pod metrics-server sử dụng câu lệnh sau:

```
kubectl get deployment metrics-server -n kube-system
```

Bước 3. Tạo config Horizontal Pod Autoscaler

(+) Tạo file HPA service-hpa.yml với điều kiện khi Ram > 80% và số pod có thể tối thiểu replicas từ 2 và tối đa replicas đến 5 pod.

```

apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: test-app-autoscale
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: test-app-autoscale
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: memory
      target:
        type: Utilization
        averageUtilization: 80

```

=> **Áp dụng file** service-hpa.yml bằng câu lệnh sau:

```
kubectl apply -f service-hpa.yml
```

(+) Tạo file HPA service-hpa.yml với điều kiện khi CPU > 70% và số pod có thể tối thiểu replicas từ 2 và tối đa replicas đến 5 pod.

```

apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:

```

```

name: test-app-autoscale
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: test-app-autoscale
  minReplicas: 2
  maxReplicas: 5
  metrics:
    targetCPUUtilizationPercentage: 70

```

Cách 2: Sử dụng Kubernetes Dashboard

Bước 1. Thực hiện deploy metrics-server

Người dùng tải file metrics-server về máy [tại đây](#)

Create from input **Create from file** Create from form

Select YAML or JSON file specifying the resources to deploy to the currently selected namespace. [Learn more](#)

Choose YAML or JSON file
components.yaml

Upload Cancel

Sau khi tải về, người dùng thực hiện cài đặt metrics-server:

Bước 2. Kiểm tra trạng thái Pod của metrics-server running

Pods								
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
metrics-server-8446d6695-cqbc5	registry.k8s.io/metrics-server/metrics-server:v0.6.4	k8s-app: metrics-server pod-template-hash: 8446d6695	worker-tp-test-v27-machine-template-ctb7q	Running	0	100.00m	10.42Mi	a minute ago

Bước 3. Thực hiện cấu hình HPA

(+) Với điều kiện RAM > 80%

Create from input Create from file Create from form

Enter YAML or JSON content specifying the resources to create to the currently selected namespace. [Learn more](#)

```
1  apiVersion: autoscaling/v1
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: php-apache
5  spec:
6    scaleTargetRef:
7      apiVersion: apps/v1
8      kind: Deployment
9      name: php-apache
10   minReplicas: 2
11   maxReplicas: 5
12   metrics:
13     - type: Resource
14       resource:
15         name: memory
16         target:
17           type: Utilization
18           averageUtilization: 80
```

Upload Cancel

(+) Với điều kiện CPU > 70%

Create from input
Create from file
Create from form

Enter YAML or JSON content specifying the resources to create to the currently selected namespace. [Learn more](#)

```

1 apiVersion: autoscaling/v1
2 kind: HorizontalPodAutoscaler
3 metadata:
4   name: php-apache-cpu
5 spec:
6   scaleTargetRef:
7     apiVersion: apps/v1
8     kind: Deployment
9     name: php-apache
10  minReplicas: 2
11  maxReplicas: 5
12  metrics:
13    targetCPUUtilizationPercentage: 70

```

Upload
Cancel

Bước 4. Theo dõi kết quả scale Pod khi đạt điều kiện

Pods								
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
php-apache-69f9bc5fd5-69d67	k8s.gcr.io/hpa-example	pod-template-hash: 69f9bc5fd5 run: php-apache	worker-tp-test-v27-machine-template-skip5r	Running	0	142.00m	16.51Mi	a minute ago
php-apache-69f9bc5fd5-b5d5x	k8s.gcr.io/hpa-example	pod-template-hash: 69f9bc5fd5 run: php-apache	worker-tp-test-v27-machine-template-ctb7q	Running	0	151.00m	12.57Mi	a minute ago
php-apache-69f9bc5fd5-n44jg	k8s.gcr.io/hpa-example	pod-template-hash: 69f9bc5fd5 run: php-apache	worker-tp-test-v27-machine-template-n2dtm	Running	0	113.00m	12.73Mi	a minute ago
load-generator	busybox:1.28	run: load-generator	worker-tp-test-v27-machine-template-ctb7q	Running	0	21.00m	1.00Mi	2 minutes ago
php-apache-69f9bc5fd5-bznqk	k8s.gcr.io/hpa-example	pod-template-hash: 69f9bc5fd5 run: php-apache	worker-tp-test-v27-machine-template-n2dtm	Running	0	128.00m	12.77Mi	31 minutes ago
php-apache-69f9bc5fd5-b7bzb	k8s.gcr.io/hpa-example	pod-template-hash: 69f9bc5fd5 run: php-apache	worker-tp-test-v27-machine-template-skip5r	Running	0	147.00m	22.64Mi	an hour ago

8. Quản lý chi phí sử dụng

8.1 Kiểm tra thời gian hoạt động của các Worker Node

viettel IDC Services Viettel Hoa Lac

EN lytt2603@gmail.com

Container / List Clusters

Clusters

Search Cluster

Name	Version	Nodes	Status	Autoscale	API Address	Dashboard Address
k8s-idx-24022bcc	v1.22.0	1 masters, 1 workers	Suspended	Enable	https://172.17.11.122:6443	https://172.17.11.12

Details Instance Network Nfs Storage Activity History

List nodes

Filter node

Worker	Start Time	End Time	Status	IP Private	Lifetime
k8s-idx-24022bcc-nfs	26/03/2024 08:48:33	-	Powered on	10.17.8.9	32.7h
k8s-idx-24022bcc-machine-tem...	26/03/2024 09:10:43	27/03/2024 17:28:20	TERMINATED	10.17.8.42	32.3h
k8s-worker-24022bcc-machine-t...	26/03/2024 09:10:43	-	Powered on	10.17.8.161	32.4h
k8s-worker-24022bcc-machine-t...	26/03/2024 09:59:48	-	Powered on	10.17.8.106	31.6h
k8s-worker-24022bcc-machine-t...	26/03/2024 14:12:54	-	Powered on	10.17.8.234	27.3h

Copyright: Viettel CHT Ltd. Co. User Guide Support Request Form Hotline: 1800 8088 Email: support@viettelidc.com

Người dùng có thể kiểm tra thời gian hoạt động và lịch sử khởi tạo, tạm dừng Worker Node bằng cách chọn vào cluster -> chọn tab **Instances** để xem thông tin thời gian hoạt động ở cột Lifetime của từng Worker Node

8.2 Xem lịch sử thông báo cước & lịch sử thanh toán cước

Quản trị Báo cáo

lytt2603 testing lytt2603@gmail.com 03645221364

Thống kê

3 Dịch vụ 3 Hoạt động 0 Sắp hết hạn

Hỗ trợ 24/7

Email: support@viettelidc.com.vn Hotline: 1800 8088 (miễn phí cước gọi)

Tư vấn viên

Họ và tên: Số điện thoại: Email:

Dịch vụ

1 Viettel Database Service Đang hoạt động 1 Hết hạn: 0

1 Viettel Virtual Private Cloud Đang hoạt động 1 Hết hạn: 0

Click để tải xuống ngay

Click để tải xuống ngay

Click để tải xuống ngay

Click để tải xuống ngay

Click để tải xuống ngay

Dịch vụ Pay as You go

1 Viettel Kubernetes Service Đang hoạt động 1 Hết hạn: 0

Click để tải xuống ngay

Dịch vụ có thể bạn quan tâm

Virtual Private Cloud Cloud Security

172.16.55.4:8002/ServiceNit/Dashboard

Xin chào lytt2603@gmail.com

Quản lý dịch vụ

Quản lý thanh toán

Đổi yêu cầu hỗ trợ

Thông tin tài khoản

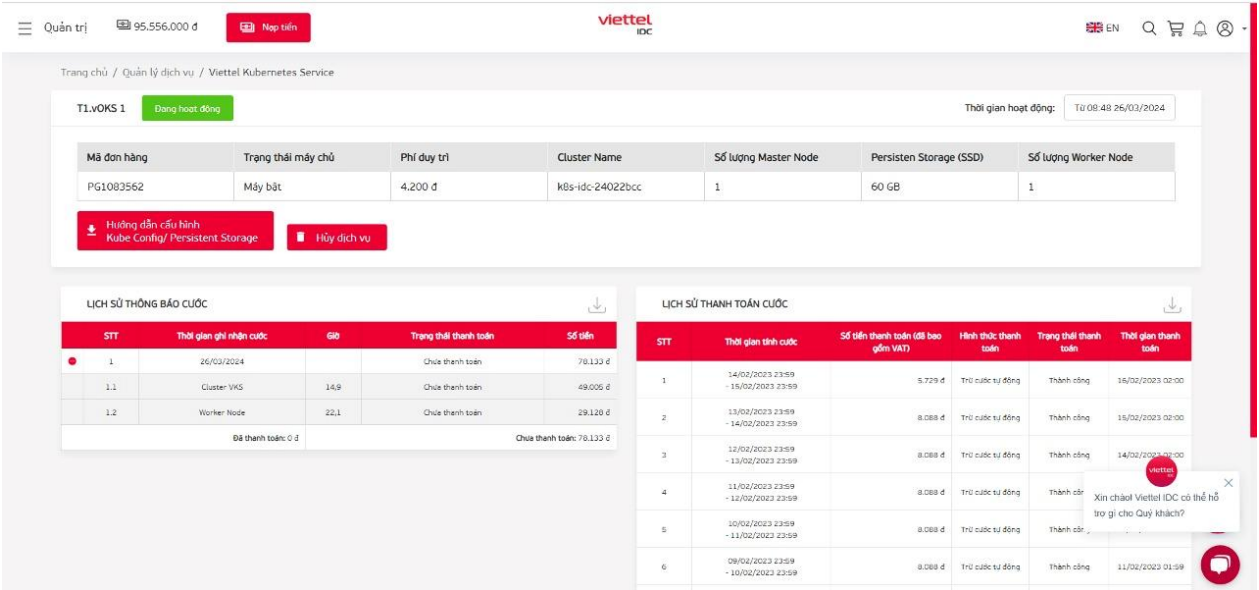
Đăng xuất

Xin chào Viettel IDC có thể hỗ trợ gì cho Quý khách?

Người dùng có thể xem lịch sử tính cước trên website bằng cách truy cập **Quản lý dịch**

vụ -> Chọn tên dịch vụ **Viettel Open Kubernetes Service** -> Xem lịch sử thông báo cước & lịch sử thanh toán cước, cụ thể:

- **Lịch sử thông báo cước** gồm các bản ghi ghi nhận cước phí hoạt động của Cluster & Worker Node Autoscale theo từng ngày, mỗi ngày được tính cước 1 lần cho tới khi Người dùng không còn nhu cầu sử dụng dịch vụ
 - o Click button  để mở rộng và xem chi tiết cước phí gồm 2 phần, cluster bao gồm các thành phần Master Node, Persistent Storage, IP và Bảng thông. Phần Worker Node là cước phí sử dụng của Worker theo thời gian đã hoạt động. Người dùng có thể đối chiếu và [kiểm tra phần thời gian hoạt động Giờ](#) với màn hình thời gian hoạt động của các Worker Node
 - o Số tiền được tính bằng cước phí của cluster và Worker node theo giờ, nhân với số giờ hoạt động. Tổng tiền của gói dịch vụ vOKS PAYG theo ngày sẽ bằng tổng tiền của 2 thành phần Cluster & Worker Node
- **Lịch sử thanh toán cước:** Người dùng cần thanh toán cước phí theo ngày và có đủ tiền trong để đảm bảo dịch vụ được hoạt động ổn định.
- Với trường hợp nợ cước dịch vụ từ 1 ngày, dịch vụ có thể bị gián đoạn. Vì vậy Viettel IDC khuyến nghị người dùng nên theo dõi phần cước dịch vụ vOKS hàng ngày để kiểm soát & dự báo được chi phí sử dụng dịch vụ, tránh gián đoạn hệ thống



The screenshot displays the Viettel IDC vOKS service interface. At the top, there's a navigation bar with 'Quản trị', a balance of '95,556,000 đ', a 'Nạp tiền' button, the 'viettel IDC' logo, and language/notifications icons. Below this, the main header shows 'Trang chủ / Quản lý dịch vụ / Viettel Kubernetes Service' and 'TL.vOKS 1' with a 'Đang hoạt động' status. A table lists service details: Mã đơn hàng (PG1083562), Trạng thái máy chủ (Máy bất), Phí duy trì (4,200 đ), Cluster Name (k8s-idc-24022bcc), Số lượng Master Node (1), Persisten Storage (SSD) (60 GB), and Số lượng Worker Node (1). Below the table are links for 'Hướng dẫn cấu hình Kube Config/ Persistent Storage' and 'Hỗ trợ dịch vụ'. The main content area is divided into two sections: 'LỊCH SỬ THÔNG BÁO CƯỚC' and 'LỊCH SỬ THANH TOÁN CƯỚC'. Both sections have a download icon. The 'LỊCH SỬ THÔNG BÁO CƯỚC' table shows billing events with columns: STT, Thời gian ghi nhận cước, Giờ, Trạng thái thanh toán, and Số tiền. It lists two events: one for 'Cluster vOKS' on 26/03/2024 and another for 'Worker Node' on 22/1. The 'LỊCH SỬ THANH TOÁN CƯỚC' table shows payment events with columns: STT, Thời gian tính cước, Số tiền thanh toán (đã bao gồm VAT), Hình thức thanh toán, Trạng thái thanh toán, and Thời gian thanh toán. It lists six payment events from 14/02/2023 to 11/02/2023. A chatbot window is visible on the right side of the dashboard.

8.3 Xem & tải lịch sử các hành động với cluster

Bước 1: Trên màn hình List Clusters, Chọn cluster cần thêm node group, chọn tab Event History

The screenshot shows the Viettel Cloud console interface. On the left is a sidebar with navigation options: Container, Clusters, Instance, Virtual Machines, Storage, Volumes, Network & Security, Subnets, Network Interfaces, Security Groups, SSH Keys, Elastic IPs, NAT Gateways, and Route Tables. The main area displays the 'List Clusters' page. A table lists two clusters:

Name	Version	Nodes	Status	API Address	Billing	VPC Network
k8s-ic-csdb04	v1.27.1	1 master01, 2 node groups	Success	https://172.17.11.74:443	Saving Plan	vpc-network-84c0d0f
k8s-ic-84c0d0f	v1.27.1	1 master01, 4 node groups	Success	https://172.17.11.113:443	PayAsYouGo	vpc-network-84c0d0f

Below the table, the 'Event History' tab is selected for the first cluster. It shows a list of events:

Event Name	Event Type	Update Time	Status	Process	Description
Install prometheus	Add-ons	30/07/2024 15:26:19	Success		Cluster k8s-ic-csdb04 install prometheus version 25.21.1
Install dashboard	Add-ons	30/07/2024 15:26:19	Success		Cluster k8s-ic-csdb04 install dashboard version 6.0.6
Change node group auto scale	Autoscale	30/07/2024 15:18:19	Success		K8S node group k8s-ic-csdb04-worker node group-92689470 change autoscale to min=2 and max=7.
Change node group auto scale	Autoscale	30/07/2024 15:17:19	Success		K8S node group k8s-ic-csdb04-worker node group-92689470 change autoscale to min=1 a...
Update node group name	Node group name	30/07/2024 15:16:23	Success		Rename node group k8s-ic-csdb04-worker node group-64475413 to node group backend
Create node group	Node group	29/07/2024 18:46:55	Success		Create node group k8s-ic-csdb04-worker node group-92689470.
Create node group	Node group	29/07/2024 18:46:55	Success		Create node group k8s-ic-csdb04-worker node group-8425415.

Bước 2: Trên danh sách các hành động mà người dùng có thể thực hiện với cluster sẽ hiển thị các hành động trong 7 ngày gần nhất, gồm có các loại hành động sau:

- Tạo thêm Node group, bao gồm mua bổ sung Node group của cluster gói Saving Plan & tạo thêm Node group gói PAYG
- Chỉnh sửa cấu hình autoscale của Node group
- Xem lịch sử cài đặt và gỡ cài đặt các add-ons như [Dashboard](#), [Grafana](#), [Prometheus](#) cho cluster

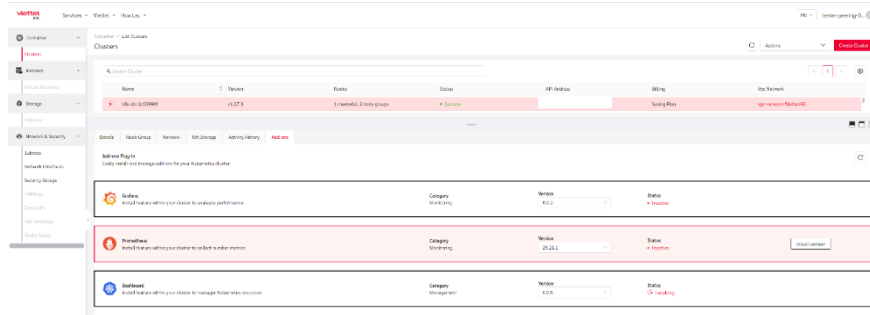
Bước 3: Người dùng có thể tải danh sách các event diễn ra với cluster để theo dõi, quản lý và kiểm soát tài nguyên dịch vụ.

9. Theo dõi cụm

9.1 Sử dụng Prometheus, Grafana

[Prometheus](#) và [Grafana](#) là các phần mềm mã nguồn mở (Open-Source), giúp giải quyết

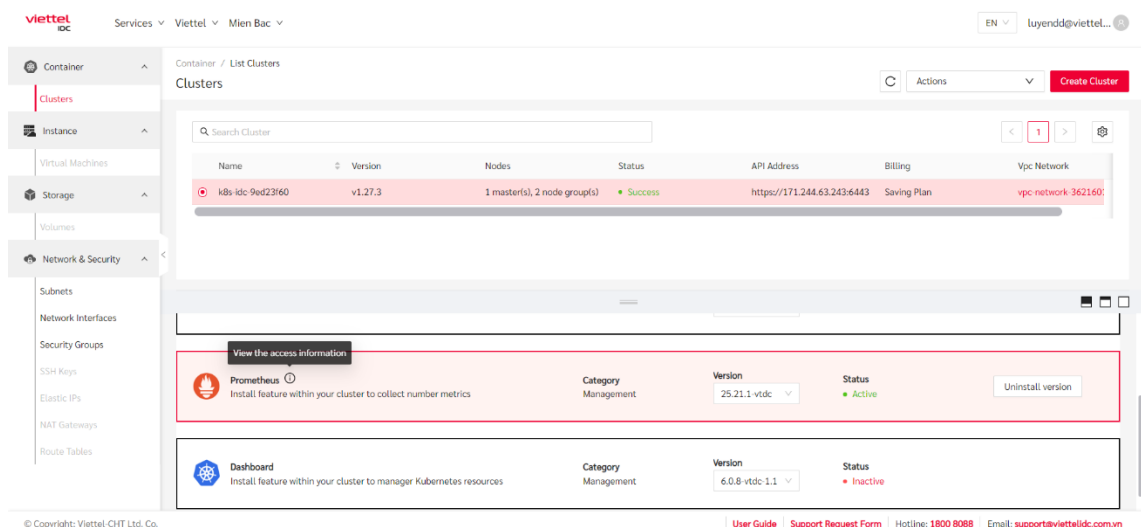
nhu cầu về thu thập và trực quan hóa dữ liệu của các dịch vụ để hỗ trợ theo dõi và giám sát trên cụm Kubernetes. Viettel IDC hỗ trợ người dùng cài đặt trực tiếp trên giao diện quản trị dịch vụ



9.1.1 Prometheus

[Prometheus](#) là một phần mềm giám sát mã nguồn mở, và đang trở nên ngày một phổ biến hơn. Prometheus rất mạnh và lại đặc biệt phù hợp để dùng cho việc giám sát các dịch vụ trên Kubernetes vì nó hỗ trợ sẵn rất nhiều các bộ template giám sát với các service opensource, giúp việc triển khai và cấu hình giám sát nhanh chóng và hiệu quả. Đi kèm với Prometheus (đóng vai trò giám sát) thì cần có thêm Alert Manager (đóng vai trò cảnh báo) để người quản trị có thể phát hiện được một vấn đề xảy ra sớm nhất có thể.

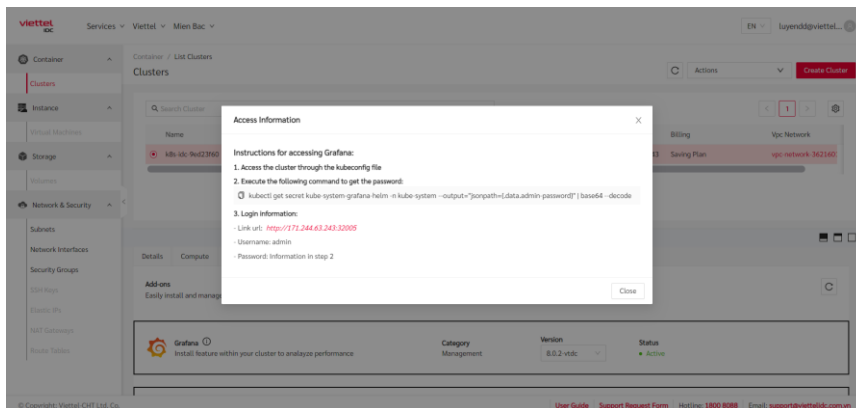
KH có thể cài đặt Prometheus và truy cập trang quản trị của Prometheus trực tiếp trên giao diện quản trị bằng cách click icon ⓘ



9.1.2 Grafana

[Grafana](#) là phần mềm mã nguồn mở dùng để phân tích và hiển thị trực quan dữ liệu. Nó giúp việc xử lý hiển thị dữ liệu trên các dashboard với khả năng tùy biến hoàn hảo, hỗ trợ rất lớn cho việc theo dõi phân tích dữ liệu theo thời gian. Nó lấy nguồn dữ liệu từ nguồn như Prometheus, Graphite hay ElasticSearch.

KH cũng có thể cài đặt và xem thông tin Grafana trực tiếp trên giao diện quản trị bằng cách click icon ⓘ tương tự với Prometheus



9.2 Sử dụng công cụ Viettel Cloud Watch (coming soon)

10. Các câu hỏi thường gặp (FAQ)

10.1 Sự khác nhau cơ bản giữa Container, Docker và Kubernetes là gì?

Container là một đơn vị phần mềm được tiêu chuẩn hóa dùng để chứa mã nguồn và các thành phần phụ thuộc cần thiết để chạy 01 ứng dụng hoặc service.

Docker là nền tảng và công cụ để đóng gói 01 ứng dụng vào container, tự động hóa việc triển khai ứng dụng dưới dạng container từ môi trường này sang môi trường khác đảm bảo ứng dụng hoạt động ổn định và đồng nhất.

Kubernetes không tự tạo ra các container mà nhiệm vụ quan trọng nhất của nó là điều phối, lập lịch chạy, quản lý trạng thái các container khi mở rộng quy mô triển khai với yêu cầu cần phối hợp nhiều container hoạt động cùng nhau để phân phối các dịch vụ riêng lẻ trên 01 hạ tầng vật lý hoặc ảo hoá.

10.2 Dịch vụ vOKS cung cấp gì cho khách hàng?

vOKS cung cấp cho khách hàng các Kubernetes cluster với lựa chọn 01 Master Node hoặc 03 Master Nodes kèm Loadbalancer để nâng cao mức độ sẵn sàng của dịch vụ, Worker Nodes mở rộng không giới hạn số lượng:

- Master Node là 1 VM, chứa tất cả thành phần logic để điều phối hoạt động của các Worker Node và định nghĩa, triển khai, quản trị vòng đời các container chạy trên Worker Node
- Worker Node: Các ứng dụng được container hóa sẽ triển khai trên các Worker Node. Với mỗi gói dịch vụ vOKS, Viettel IDC cung cấp tối thiểu 03 Worker Node trong nhằm nâng cao khả năng chịu lỗi và tính sẵn sàng khi triển khai ứng dụng.
- NFS Node - Persistent Storage là 1 NFS Server (Network File System) được sử dụng như một giao thức kết nối các container đang hoạt động trong Worker Node tới một vùng lưu trữ dữ liệu độc lập (Persistent Storage).

10.3 Khác biệt giữa Node Storage và Persistent Storage trong 01 cụm K8s?

Node Storage có dung lượng cố định là 50 GB (SSD), được sử dụng để lưu trữ dữ liệu (dạng cache) của container trong quá trình hoạt động. Dữ liệu này chỉ được lưu tạm thời và sẽ bị xóa khi người dùng restart/reschedule các container.

Mặt khác, dữ liệu của container trong quá trình hoạt động được ghi lại trên vùng lưu trữ riêng (Persistent Storage) và sẽ được giữ lại khi các container bị restart/reschedule; dữ liệu này có thể được chuyển giao giữa các container.

10.4 vOKS có tương thích với các ứng dụng dạng container đang chạy trên Cloud của các nhà cung cấp khác?

Người dùng hoàn toàn có thể sử dụng tất cả plugin hoặc công cụ hiện có được phát triển từ cộng đồng Kubernetes. Các ứng dụng đang chạy trên bất kỳ môi trường Kubernetes tiêu chuẩn ở hạ tầng Cloud khác (AWS, Microsoft Azure..) hay trung tâm dữ liệu On-Premise đều tương thích và dễ dàng di chuyển sang dịch vụ vOKS của Viettel IDC.