BÁO CÁO THỰC HÀNH

**Bài thực hành số 5: Kubernetes K3S**

**Môn học:** Hệ tính toán phân bố

**Lớp:** NT533.P21.1

**GVHD**: **Lê Anh Tuấn**

**THÀNH VIÊN THỰC HIỆN (Nhóm 14):**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Phạm Đức Anh | 22520067 |
| 2 | Tăng Dũng Cẩm | 22520141 |

|  |
| --- |
| **Điểm tự đánh giá** |
| **9** |

**ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| Tổng thời gian thực hiện | 3 ngày |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất, kiến nghị |  |

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

***Mục Lục***

[**1. Triển khai K3s cluster trên 2 máy ảo ubuntu sử dụng K3sup (7đ) 3**](#_Toc199024000)

[**2. Triển khai Dashboard (1đ) 6**](#_Toc199024001)

[**3. Tìm hiểu ứng dụng OpenFaaS để triển khai một microservices tự xây dựng (bằng Python, Go, JavaScript...) (2đ) 10**](#_Toc199024002)

**BÁO CÁO CHI TIẾT**

## Triển khai K3s cluster trên 2 máy ảo ubuntu sử dụng K3sup (7đ)

* Để bắt đầu quá trình triển khai, thực hiện tải script cài đặt K3sup từ địa chỉ chính thức của dự án cho cả máy master và máy node.
* Tiếp theo, cài đặt file thực thi **k3sup** vào thư mục /usr/local/bin, giúp người dùng có thể gọi đến k3sup từ bất kỳ đâu trên hệ thống.

A screenshot of a computer

AI-generated content may be incorrect.

* Tiến hành tạo một **cặp khóa SSH** (được lưu vào ~/.ssh/) cho từng máy gồm có:
* id\_rsa: private key (giữ lại máy)
* id\_rsa.pub: public key (chia sẽ cho các máy khác)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

* Dùng lệnh ssh-copy-id để sao chép khóa công khai (id\_rsa.pub) từ master tới node để thiết lập xác thực SSH mà không cần mật khẩu:

A black screen with white text

AI-generated content may be incorrect.

* Tương tự, sao chép ssh public key từ node tới master:

A computer screen with white text

AI-generated content may be incorrect.

* SSH thành công từ master tới node mà không cần mật khẩu đăng nhập:

A computer screen shot of a black screen

AI-generated content may be incorrect.

* Trên 2 máy, dùng lệnh sudo visudo để chỉnh sửa tệp /etc/sudoers và thêm vào *ducanh ALL=(ALL) NOPASSWD: ALL* giúp người dùng có thể chạy bất kỳ lệnh nào với quyền root mà không cần nhập mật khẩu:

A screenshot of a computer

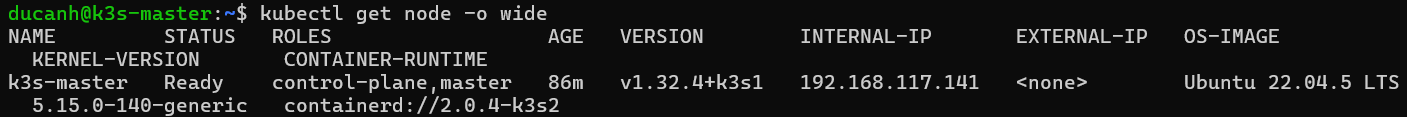
AI-generated content may be incorrect.

* Thực hiện tạo thư mục ~/.kube trong home của người dùng hiện tại. Đó là nơi lưu file cấu hình kết nối đến cluster Kubernetes, thường sẽ là ~/.kube/config
* Thực hiện cài đặt K3s lên server thông qua SSH bằng công cụ k3sup và đồng thời lấy file kubeconfig về máy hiện tại để dùng kubectl quản lý cluster.

A screen shot of a computer

AI-generated content may be incorrect.

* Dùng lệnh get node để hiển danh sách các node trong Kubernetes cluster kèm theo thông tin chi tiết của chúng.

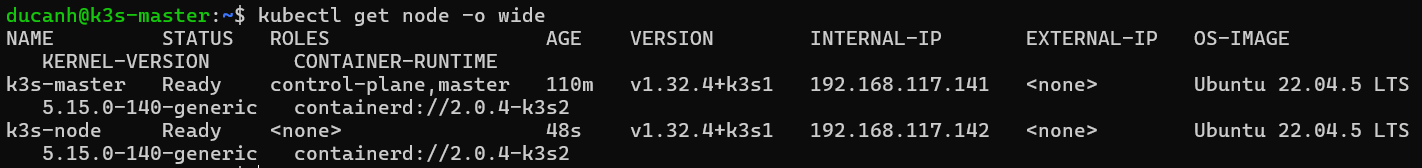


* Thực hiện thêm node vào cluster đang hoạt động với câu lệnh k3sup join và những thành phần sau:
* --ip: Địa chỉ IP của máy node (agent)
* --server-ip: Địa chỉ IP nội bộ của node master
* --user: Tài khoản có quyền SSH vào máy agent
* --k3s-channel stable: Yêu cầu cài đặt phiên bản K3s ổn định

A black screen with white text

AI-generated content may be incorrect.

* Gọi lệnh get node để hiển thị lại danh sách các node có trong Kubernetes cluster, lúc này đã thấy một node mới tên là k3s-node được thêm thành công vào cluster.



## Triển khai Dashboard (1đ)

* Đầu tiên cần tải và cài đặt Helm v3, là một trình quản lý package dành cho ứng dụng chạy trong cluster Kubernetes.
* Tiếp theo tạo thêm một **Helm chart repository** mới vào hệ thống local. Đối với bài thực hành này sẽ là repository của Kubernetes Dashboard, cũng chính là giao diện web dùng để quản lý Kubernetes cluster.

A screen shot of a computer

AI-generated content may be incorrect.

* Tiếp theo cần cập nhật Kubernetes Dashboard lên phiên bản mới với câu lệnh **helm upgrade –install**.
* Thành phần:
* **--create-namespace** sẽ tự động tạo namespace nếu chưa tồn tại.
* **--namespace kubernetes-dashboard** sẽ cài đặt Dashboard vào namespace kubernetes-dashboard.

A screenshot of a computer

AI-generated content may be incorrect.

* Sau khi cài đặt Kubernetes Dashboard, file **dashboard-adminuser.yaml** thường được dùng để tạo tài khoản admin cho phép đăng nhập vào giao diện web Dashboard với đầy đủ quyền.
* Dựa vào nội dung trong file yaml, hệ thống sẽ thực hiện tạo một ServiceAccount và gán cho tài khoản đó quyền cluster-admin. Điều này cho phép đăng nhập vào Kubernetes Dashboard với toàn quyền quản trị

A screenshot of a computer

AI-generated content may be incorrect.

* Thực hiện triển khaicác tài nguyên được định nghĩa trong file **dashboard-adminuser.yaml** với công cụ dòng lệnh kubectl để tương tác với Kubernetes.
* Tiếp theo cần tạo một token truy cập cho ServiceAccount **admin-user.**
* Thành phần **-n kubernetes-dashboard** sẽ chỉ định namespace nơi tài khoản **admin-user** được tạo
* Sau khi token được xuất ra, để truy cập giao diện Kubernetes Dashboard qua trình duyệt, sử dụng lệnh port-forward để ánh xạ cổng từ máy local đến Service nội bộ của Dashboard

A screen shot of a computer

AI-generated content may be incorrect.

* Tại đây khi kiểm tra xem cổng 8443 trên máy có đang được một tiến trình nào sử dụng không thì thấy rằng tiến trình kubectl đang sử dụng để lắng nghe trên cả IPv4 và IPv6.



* Gọi lại lệnh kubectl get pods để liệt kê tất cả các pod đang chạy trong Kubernetes cùng với chỉ định namespace là kubernetes-dashboard, cũng là nơi chứa các Pod của giao diện Dashboard.
* Từ kết quả xuất ra, có thể thấy rằng các Pod xử lý các công việc quan trọng như: API backend của Dashboard, xác thực người dùng, thu thập và hiển thị metric hệ thống, giao diện frontend đều trong trạng thái hoạt động ổn định với không lần khởi động lại nào (RESTARTS: 0).

A screen shot of a computer

AI-generated content may be incorrect.

* Tại đây khi muốn truy cập vào Kubernetes Dashboard, sử dụng đường dẫn <https://localhost:8443> trình duyệt sẽ trả về trang login vào hệ thống.

A screenshot of a computer

AI-generated content may be incorrect.

* Sử dụng token đã được tạo ra trước đó để có thể đang nhập vào ServiceAccount **admin-user**:

A screenshot of a computer

AI-generated content may be incorrect.

## Tìm hiểu ứng dụng OpenFaaS để triển khai một microservices tự xây dựng (bằng Python, Go, JavaScript...) (2đ)

* OpenFaaS là một nền tảng mã nguồn mở cho phép triển khai các functions hoặc ứng dụng container dưới dạng FaaS.
* Theo đó nó giúp ta triển khai mã nguồn dưới dạng các hàm nhỏ có thể gọi qua HTTP mà không cần quản lý hạ tầng phức tạp.
* Tính năng nổi bật của OpenFaaS:
* Có giao diện web để quản lý hàm
* Viết hàm bằng nhiều ngôn ngữ (Python, Node.js, Go, Java, v.v.)
* Hỗ trợ autoscaling
* Kết nối với Prometheus để dễ dàng theo dõi hiệu suất và thống kê
* CLI tiện dụng (faas-cli)
* Để triển khai OpenFaaS đầu tiên cần cài đặt công cụ dòng lệnh (CLI) Arkade để giúp triển khai trên cụm k3s:

A computer screen with white text

AI-generated content may be incorrect.

* Tạo namespace openfaas để nhóm các tài nguyên và tách biệt các tài nguyên của OpenFaaS với các phần còn lại của cụm Kubernetes:



* Tiếp tục, tạo namspace openfaas-fn (nơi chứa các function triển khai bằng OpenFaaS) và tiến hành cài đặt OpenFaaS trên cụm K3s sử dụng Helm:

A screenshot of a computer screen

AI-generated content may be incorrect.

* Khi liệt kê các deployments trong namespace openfaas, có thể thấy 5 deployments quan trọng của OpenFaaS: alertmanager, gateway, nats, prometheus và queue-worker đều đang ở trạng thái ready:

A screen shot of a black and white screen

AI-generated content may be incorrect.

* Cài đặt OpenFaaS CLI:

A screen shot of a computer

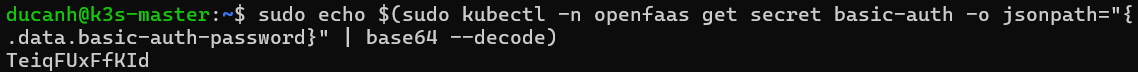
AI-generated content may be incorrect.

* Trước khi kết nối tới OpenFaaS gateway, ta kiểm tra trạng thái triển khai của deployment gateway trong namespace openfaas và thiết lập port forwarding cho OpenFaaS gateway. Thông qua đó, lưu lượng từ cổng 8080 trên máy cục bộ sẽ được chuyển tiếp đến cổng 8080 của gateway trên k3s (“&” ở cuối lệnh giúp chạy trên background):

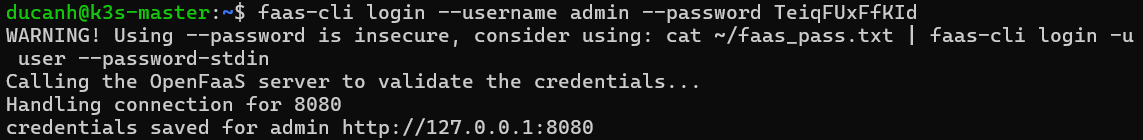
A screen shot of a computer

AI-generated content may be incorrect.

* Truy xuất Secret basic-auth trong namespace openfaas, rồi trích xuất trường basic-auth-password bằng jsonpath, sau đó giải mã nó từ định dạng Base64 để hiển thị mật khẩu của admin:



* Tiến hành đăng nhập vào OpenFaaS Gateway:



* Liệt kê các function có sẵn trong OpenFaaS store. Ở đây, có thể thấy nhiều hàm như nodeinfo, env, figlet, printer, curl, external-ip,.., cùng với thông tin tên nhà phát triển (AUTHOR) cùng mô tả (DESCRIPTION) của từng function:

A screenshot of a computer program

AI-generated content may be incorrect.

* Triển khai function figlet thông qua lệnh deploy và faas-cli sẽ gửi yêu cầu triển khai đến Gateway thông qua kết nối port-forwarding đã thiết lập trước đó (<http://127.0.0.1:8080>).
* OpenFaaS gateway sau đấy, sẽ kéo Docker image của function (openfaas/figlet:latest), tạo các tài nguyên Kubernetes cần thiết như deployment, service và khởi chạy function:

A black screen with white text

AI-generated content may be incorrect.

* Truy cập giao diện OpenFaaS (<http://localhost:8080/ui/>) để kiểm tra function figlet đã được triển khai, các thông tin hiển thị gồm:
  + Status: Trạng thái khởi tạo của hàm (Not ready, Ready)
  + Replicas: Bản sao của hàm
  + Invocation count: Số lần hạm được gọi
  + Image: Docker image đang được sử dụng
  + URL: URL để gọi (invoke) hàm figlet từ bên ngoài

A screenshot of a computer

AI-generated content may be incorrect.

* Tải về template python3-http do OpenFaaS cung cấp và tạo một hàm mới tên là python3-func dựa trên template đó. Trong đó, một thư mục mới (python3-func) sẽ được tạo và chứa file code handler mặc định (handler.py) và một file cấu hình stack.yaml:

A screen shot of a computer

AI-generated content may be incorrect.

* Ở handler.py, ta định nghĩa một danh sách sinh viên với mã số sinh viên và họ tên tương ứng. Hàm sẽ trả về mã trạng thái (statusCode - 200) và danh sách có dạng JSON:

A screen shot of a computer

AI-generated content may be incorrect.

* Ở file cấu hình cho function (stack.yaml), tiến hành xác định gateway đã thiết lập port-forwading trước đó để OpenFaaS CLI kết nối tới. Sau đó, định nghĩa tên, template sử dụng, tên Docker image sẽ được tạo cho function (dùng username Docker Hub để đảm bảo image sẽ được đẩy tới registry và có thể pull về bởi Kubernetes):

A screen shot of a computer

AI-generated content may be incorrect.

* Sau khi đăng nhập vào Docker (docker login), tiến hành build Docker image cho python3-func dựa trên cấu hình trong file stack.yaml:

A screenshot of a computer program

AI-generated content may be incorrect.

* Sau khi build xong, push Docker image từ máy cục bộ lên Docker Hub:

A computer screen with white text and black background

AI-generated content may be incorrect.

* Cuối cùng, triển khai function lên cụm k3s, theo đó faas-cli sẽ đọc file stack.yaml để lấy thông tin cấu hình như tên function, image và template, sau đấy Kubernetessẽ kéo image, tạo các tài nguyên cần thiết (Deployment, Service,...) và khởi chạy function.
* Sau khi triển khai thành công, ta có thể kiểm tra danh sách các fucntion đang được quản lý cùng thông tin về số lần gọi (Invocations) và số bản sao (Replicas):

A computer screen shot of a black screen

AI-generated content may be incorrect.

* Mở giao diện, function đã được triển khai và đang ở trạng thái Ready.
* Khi tiến hành gọi (Invoke) hàm thì ở phần Response status trả ra kết quả 200 (OK) cho thấy hàm đã xử lý yêu cầu thành công cùng Response body với nội dung JSON được định nghĩa trong handler.py và thời gian thực hiện (Round-trip (s)):

A screenshot of a computer

AI-generated content may be incorrect.

**KẾT THÚC BÁO CÁO**