BÁO CÁO BÀI TẬP 3

**Môn học: An toàn mạng máy tính nâng cao**

**BT3:**

Tiếp cận với môi trường Kubernetes và sử dụng kube bench để kiểm tra tính an toàn của môi trường Kubernetes

*GVHD: Nguyễn Duy*

1. **THÔNG TIN CHUNG:**

Lớp: NT534.N21.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hoàng Văn Anh Đức | 20520890 | 20520890@gm.uit.edu.vn |
| 2 | Nguyễn Tú Ngọc | 20521665 | 20521665@gm.uit.edu.vn |
| 3 | Trần Đặng Hồng Loan | 20521543 | 20521543@gm.uit.edu.vn |
| 4 | Lê Nhật Hào | 20520489 | 20520489@gm.uit.edu.vn |

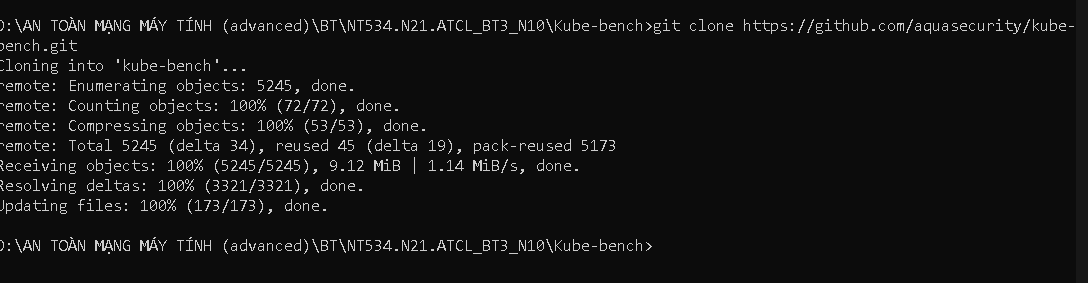
1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Tải kube bench | 100% |
| 2 | Sử dụng kube bench | 100% |
| 3 | Nhận xét kết quả đánh giá | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

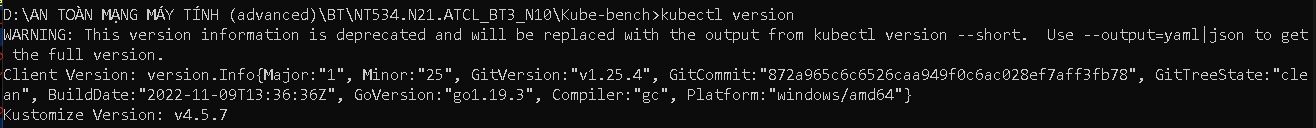
1.**Tải kube bench**: <https://github.com/aquasecurity/kube-bench.git>



**Cài đặt kubectl bằng WindowPowerShell**:

curl.exe -Lo kind-windows-amd64.exe https://kind.sigs.k8s.io/dl/v0.18.0/kind-windows-amd64

Move-Item .\kind-windows-amd64.exe E:\kubectl \kind.exe



Kubectl 1.25.4

Graphical user interface, application

Description automatically generated

2. **Sử dụng kube bench**

Tạo tài khoản trên AWS EKS

Sử dụng EKSCTL tool để :

* Set up
* Tạo Cluster Control Plane (Master node)
* Worker nodes và kết nối đến Cluster
  + Cluster name , k8s version
  + Region và VPC for cluster
  + Set security for cluster
  + Create node group
  + Choose cluster it ‘ll attach to
  + Define security group , select instance type , resources

Chỉ = 1 control command (use default values) nhưng ta sẽ làm bằng web interface

1. **Tạo role để dùng cho việc tạo Control plane**

Ta tạo role vì :

* Nó cho phép các pod trong cluster của ta truy cập vào các tài nguyên AWS mà không cần phải cung cấp các thông tin đăng nhập hoặc chứng chỉ riêng của mỗi pod. Nó cũng cho phép ta quản lý quyền truy cập của các pod dựa trên các vai trò khác nhau mà chúng có thể đóng vai trò trong ứng dụng của mình.

Graphical user interface, text, application

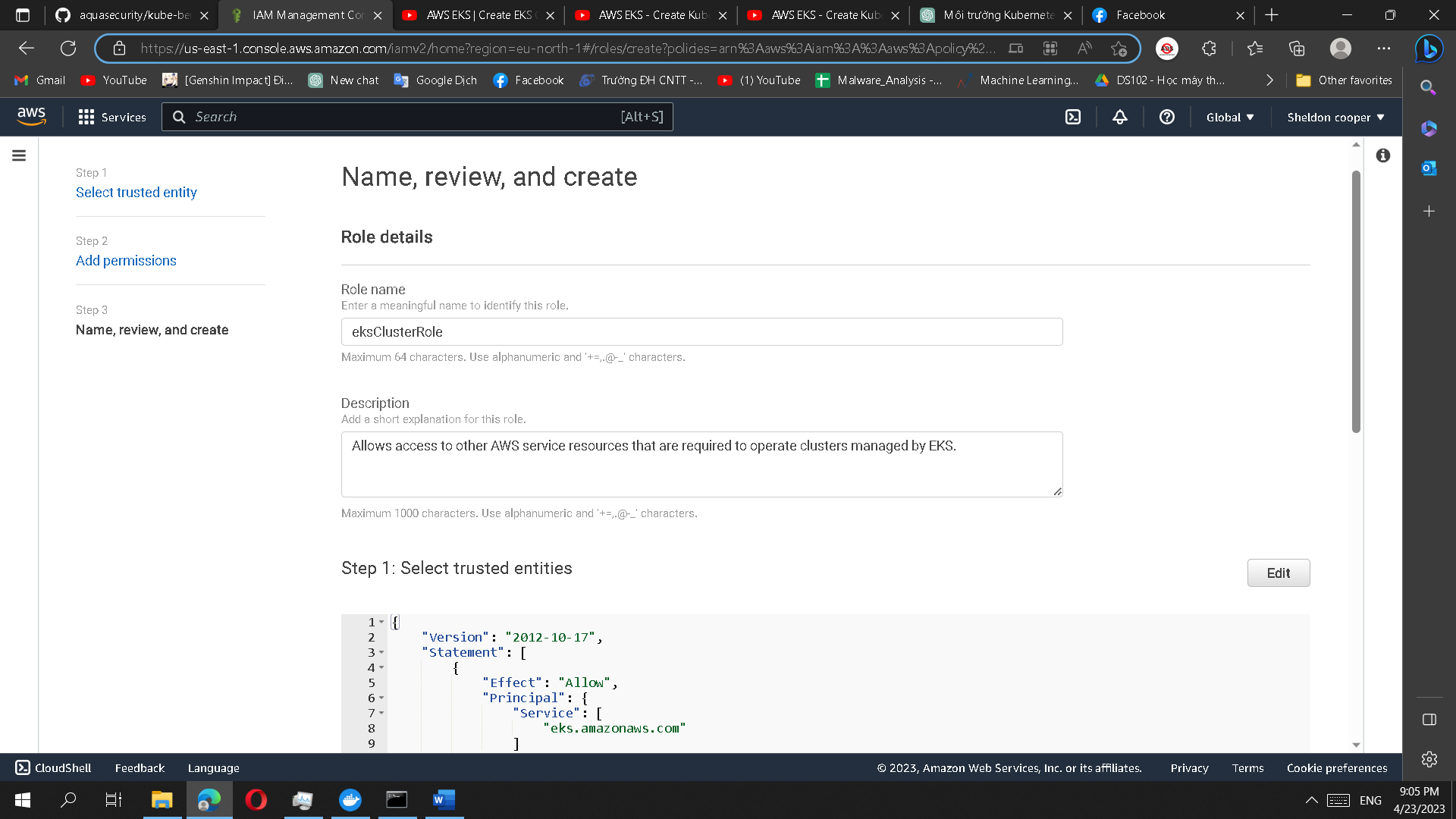
Description automatically generated



Icon

Description automatically generatedGraphical user interface, text

Description automatically generated



A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

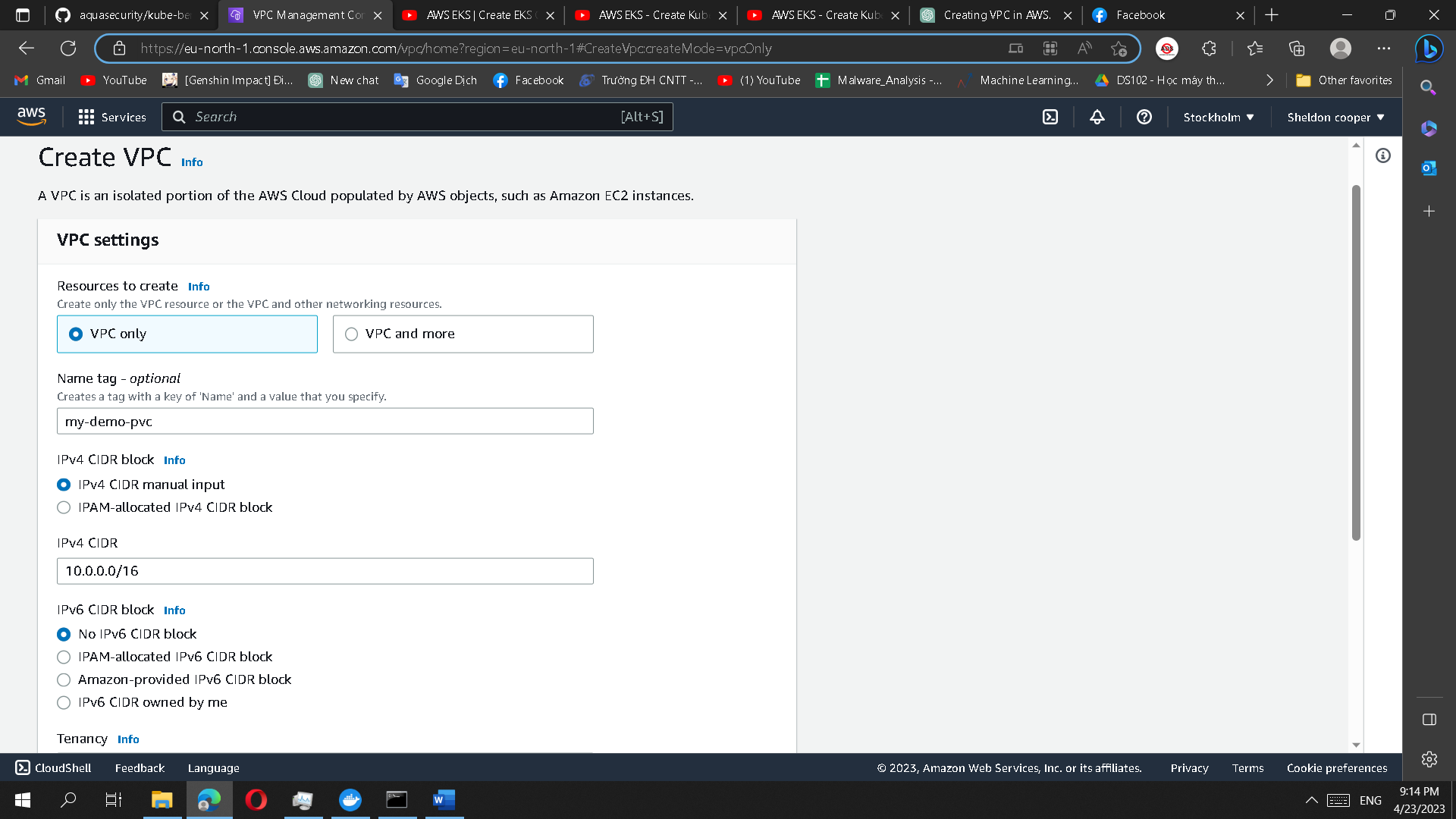
Tạo role thành công

1. **Tạo VPC**

* Cho phép chúng ta tạo một môi trường đám mây riêng, nơi có thể quản lý và kiểm soát các tài nguyên đám mây theo cách mà ta muốn, mà không cần phải chia sẻ tài nguyên với người dùng khác

Graphical user interface, text

Description automatically generated

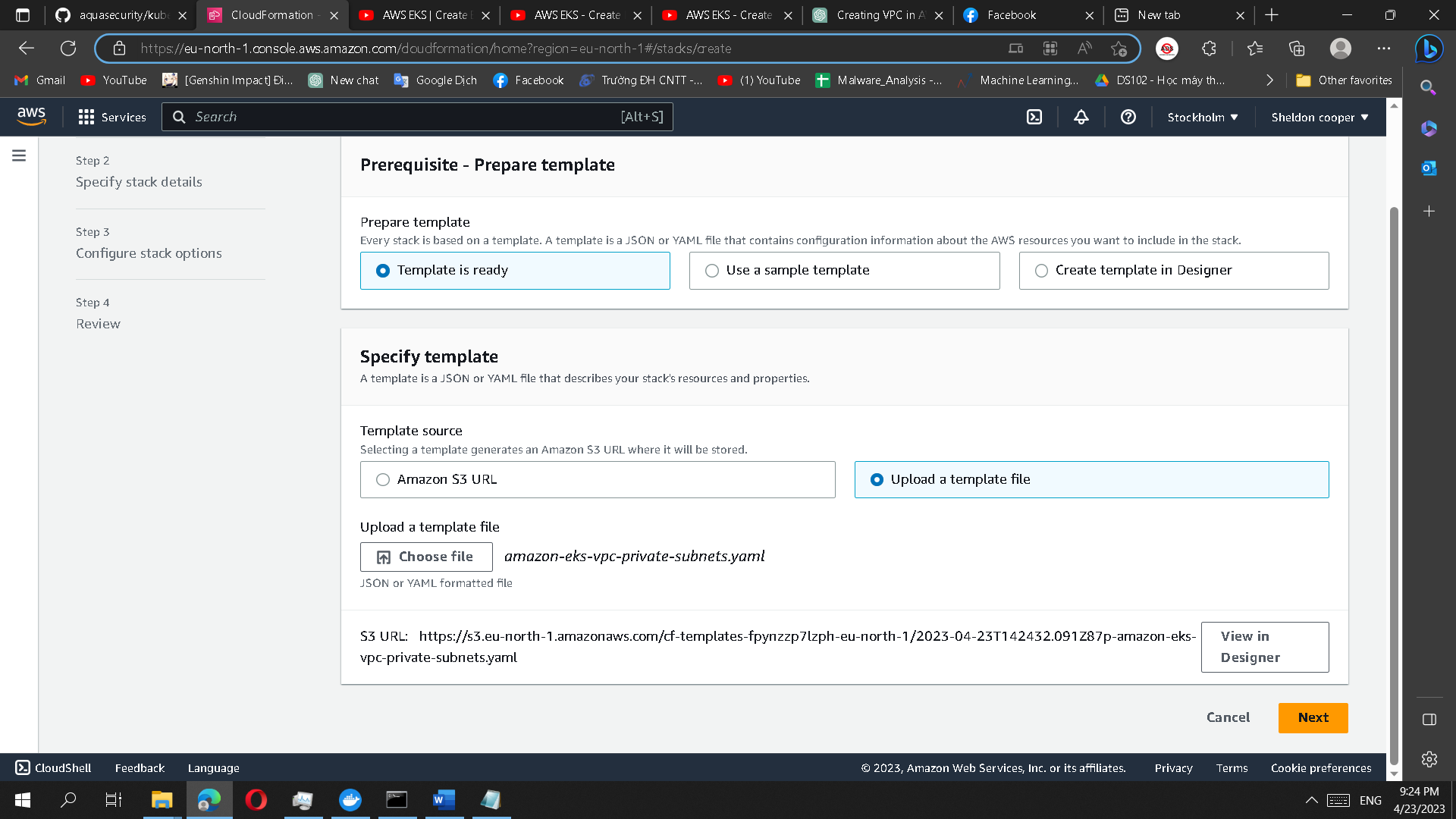


1. **Tạo CloudFormation Stack**

* CloudFormation có thể giúp quản lý việc triển khai cụm Kubernetes của ta và các tài nguyên liên quan đến đó. Và nó còn có thể được sử dụng để triển khai các tài nguyên như:
  + Các mẫu tùy chỉnh để triển khai cụm Kubernetes trên EKS
  + Các tài nguyên liên quan đến EKS như node group, IAM roles và VPC endpoints
  + Các ứng dụng và dịch vụ đang chạy trên cụm Kubernetes
* CloudFormation cũng có thể giúp đơn giản hóa việc quản lý cụm Kubernetes bằng cách tự động hóa quá trình triển khai và cập nhật, cung cấp khả năng phục hồi khi có sự cố xảy ra và đảm bảo tính nhất quán của các tài nguyên đám mây .

Graphical user interface, text

Description automatically generated



Đặt tên : MyEksStack và giữ mọi thứ default

Graphical user interface, text, application, email

Description automatically generated

Submit stack

Đang tạo

A screenshot of a computer

Description automatically generated

VPC->subnet

Graphical user interface, table

Description automatically generated with medium confidence

VPC->Router table

Graphical user interface, text, application, email

Description automatically generated

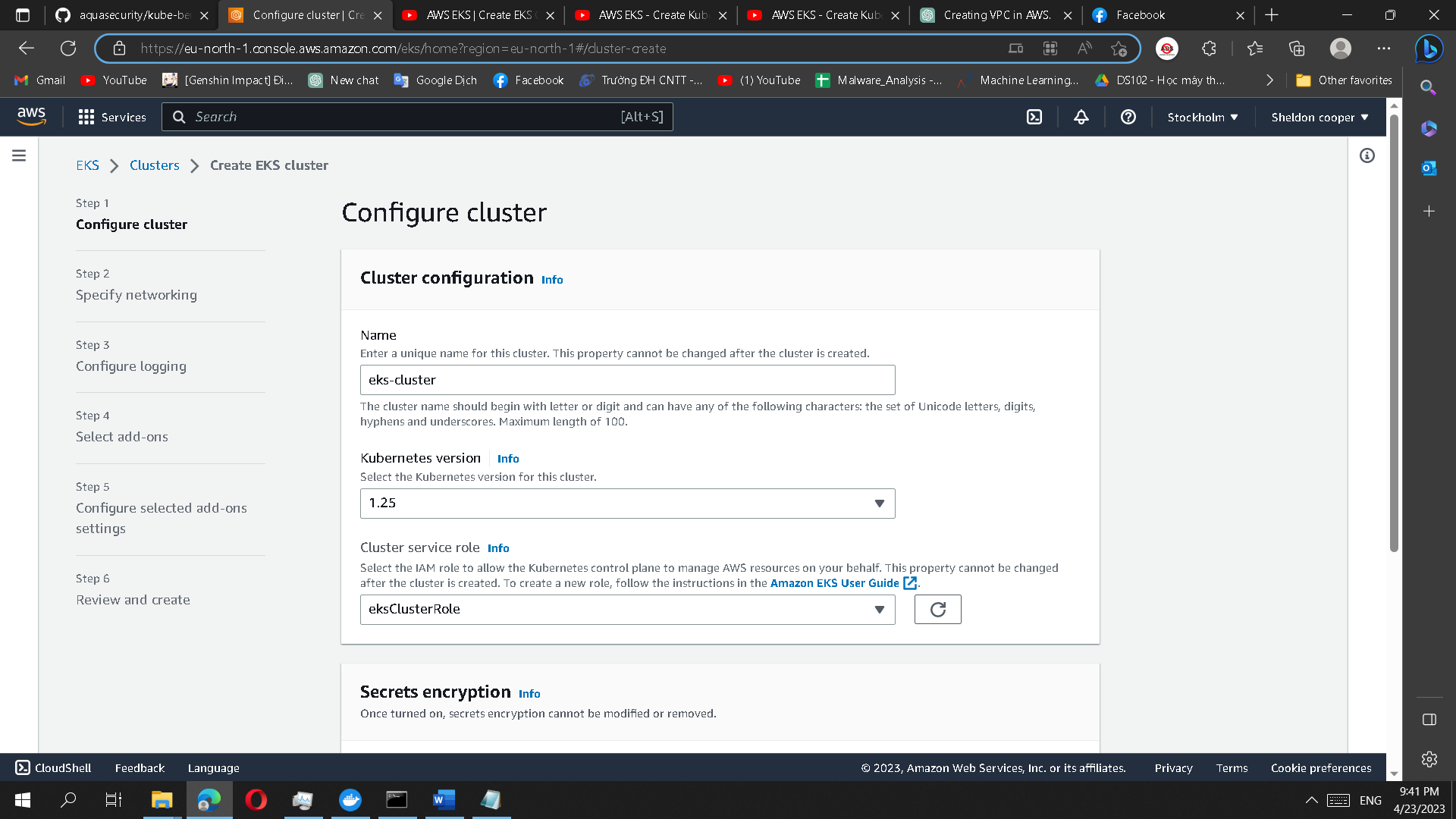
Thấy có data là đc

4 .Tạo cluster

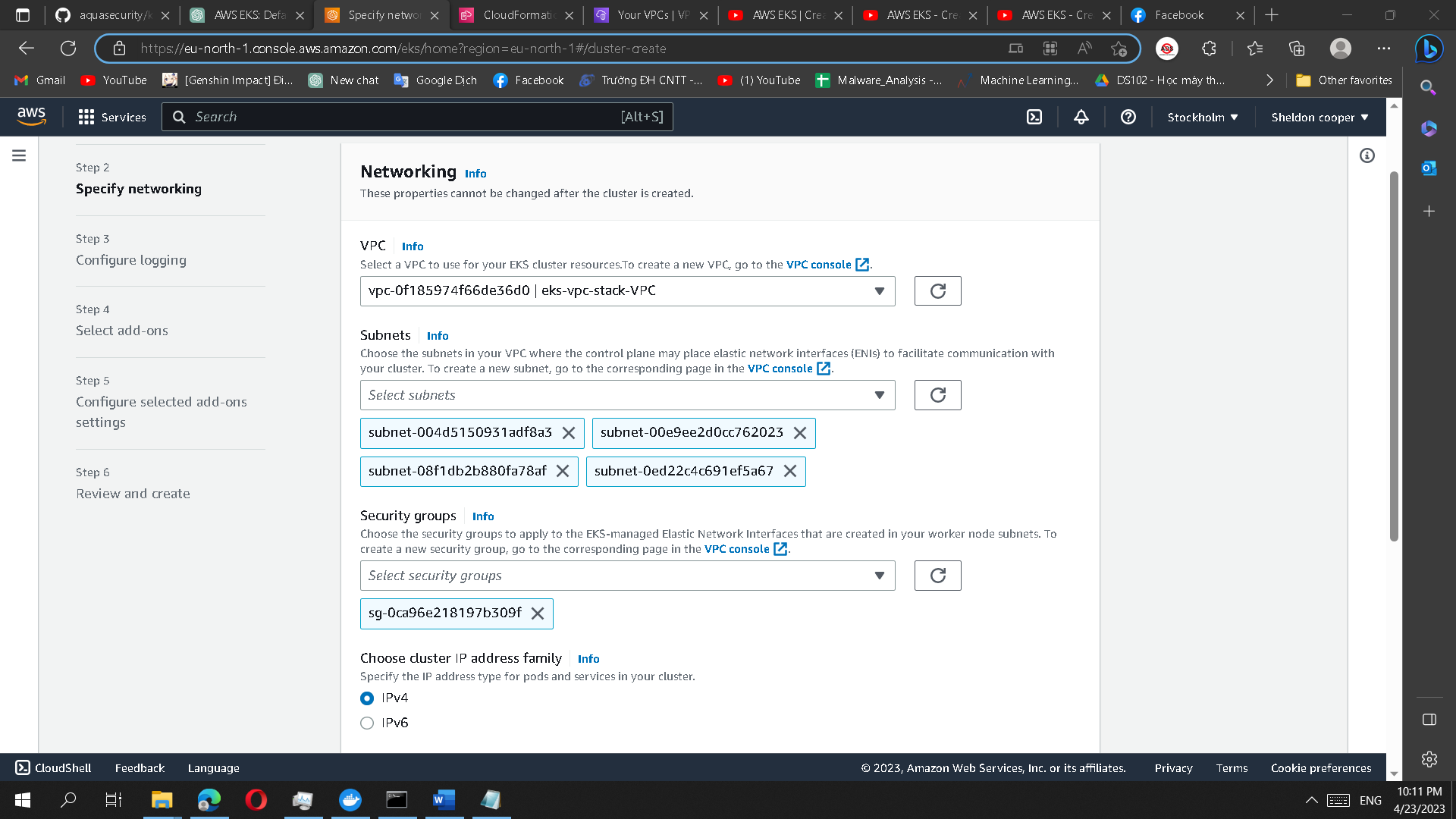
Graphical user interface, text, application

Description automatically generated

Đặt tên và chọn Role chúng ta vừa mới tạo



Chọn VPC và ở Group security chọn stack mà chúng ta mới tạo



* API endpoint của cluster EKS sẽ được cung cấp thông qua Internet và qua kết nối mạng riêng tư. Điều này cho phép các ứng dụng hoặc dịch vụ có thể truy cập vào các API endpoint thông qua Internet hoặc một mạng riêng tư.

Graphical user interface, text, application, email

Description automatically generated

default

A screenshot of a computer

Description automatically generated

default

A screenshot of a computer

Description automatically generated

default

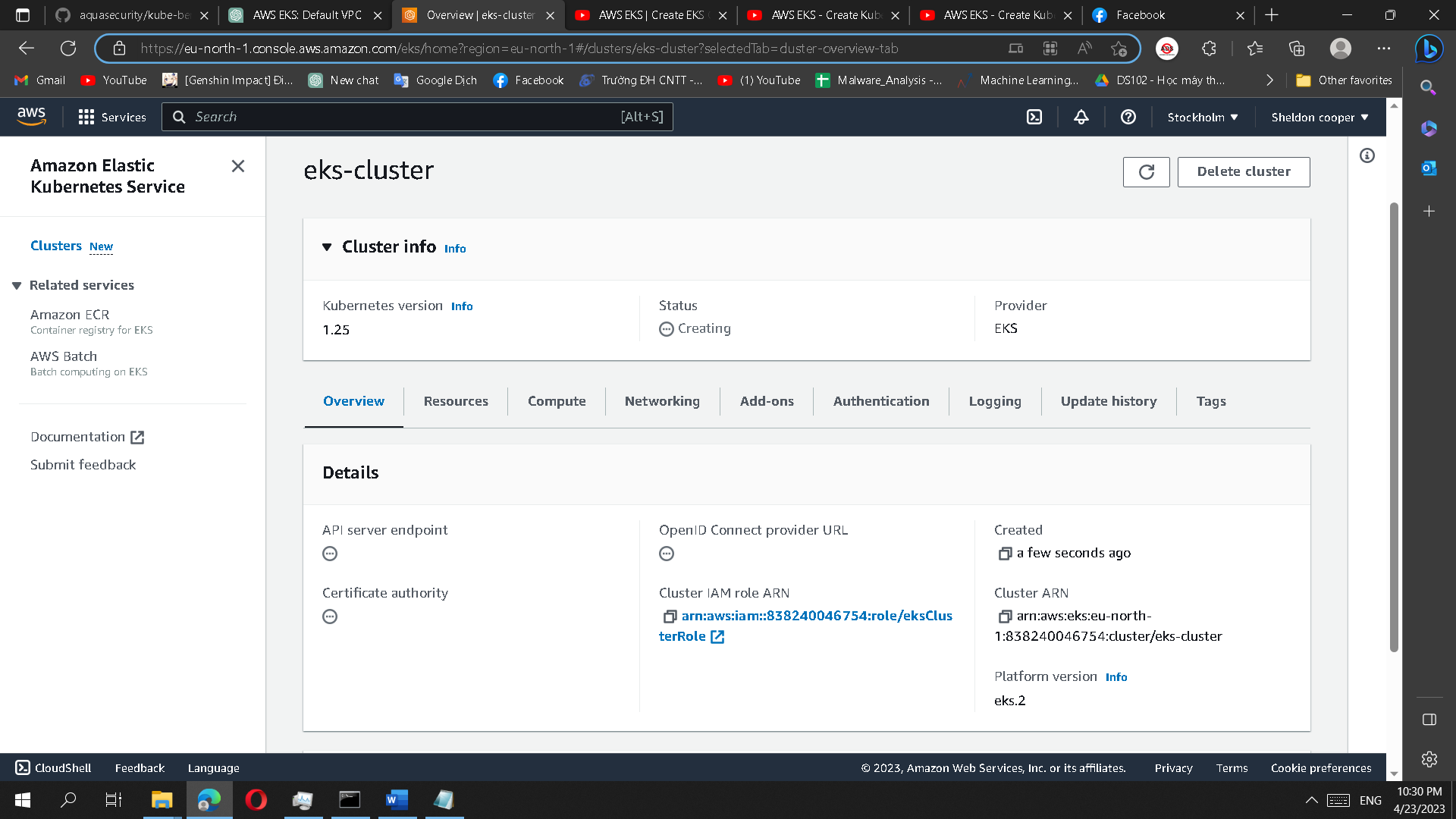
A screenshot of a computer

Description automatically generated

default

A screenshot of a computer

Description automatically generated



Nhưng mà nhiều lỗi quá

* Dùng tool eksctl

Dùng third-party để tải :

chocolatey install eksctl

Text

Description automatically generated

Sử dụng command để tạo cluster:

**eksctl create cluster --name MyEksCluster --region ap-southeast-1 --nodegroup-name linux-nodes --node-type t2.micro --nodes 2**

* Command này sẽ tạo ra một cluster trong Amazon Elastic Kubernetes Service (EKS) với các thông số như sau:
  + Tên cluster: " MyEksCluster "
  + Vùng khu vực: "ap-southeast-1" (vùng châu Á Thái Bình Dương - Singapore)
  + Tên nodegroup: "linux-nodes"
  + Loại node: "t2.micro" (một loại máy ảo EC2 có cấu hình nhỏ)
  + Số lượng node đc truy cập vào cluster: 2

Trước khi làm điều đó thì ta phải tạo user

Graphical user interface, text, application

Description automatically generated

Thêm quyền cho user

A screenshot of a computer

Description automatically generated Graphical user interface, text, application

Description automatically generated

Tạo accesskey cho user

Graphical user interface, text, application

Description automatically generated

Link tải AWS CLI :

<https://awscli.amazonaws.com/AWSCLIV2.msi>

Tạo AWS credentials để eksclt có thể authenticate với AWS

User “ADMIN” tương tác với các dịch vụ AWS thông qua các lệnh dòng lệnh

Text

Description automatically generated

Text

Description automatically generated

Tạo Cluster thành công

A picture containing background pattern

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Sử dụng command để cấu hình kubectl để kết nối đến cluster Amazon EKS:

**aws eks --region ap-southeast-1 update-kubeconfig --name MyEksCluster**

Text

Description automatically generated

Xem các nodes sẽ được scan bằng kube bench

**Kubectl get nodes**

Scan nodes bằng 1 job đã định nghĩa sẵn bởi kube-bench

**Kubectl apply -f job-eks,yaml**

File job-eks.yaml

Text

Description automatically generated

Sau khi chạy lệnh kubectl apply -f job-eks.yaml, Kubernetes sẽ tạo một Job mới và bắt đầu quá trình quét bảo mật trên cluster. Khi Job hoàn thành nhiệm vụ của mình, Kubernetes sẽ tạo một hoặc nhiều Pod để thực hiện công việc quét bảo mật

Sử dụng lệnh dưới đây cho phép xem trạng thái của các Pod được tạo ra để thực hiện công việc quét bảo mật này :

**kubectl get pods**

Sử dụng lệnh để xem logs của Pod và biết được kết quả của quá trình quét bảo mật.

**kubectl logs <pod-name>**

Text

Description automatically generated

**3.Nhận xét kết quả đánh giá**

**Phân tích logs :**

* **[INFO] 3 Worker Node Security Configuration**: Đây là tiêu đề cho phần phân tích các cấu hình bảo mật cho các node worker trong cluster.
* **[INFO] 3.1 Worker Node Configuration Files**: Đây là tiêu đề cho phần kiểm tra các file cấu hình của các node worker.
  + **[PASS] 3.1.1 Ensure that the kubeconfig file permissions are set to 644 or more restrictive (Manual):** Đây là thông báo kết quả kiểm tra rằng quyền truy cập của file kubeconfig được thiết lập là 644 hoặc nghiêm ngặt hơn.
    - Kubeconfig file là một tệp cấu hình cho phép người dùng hoặc một ứng dụng kết nối và sử dụng Kubernetes API server.
    - Tệp này chứa thông tin xác thực và cấu hình được sử dụng bởi CLI (command line interface) như kubectl hoặc bởi các ứng dụng như Dashboard để truy cập vào Kubernetes cluster.
    - Trong hệ thống tệp Unix, quyền truy cập được xác định bằng cách gán cho tệp một loại quyền (owner, group, other) và một bộ quyền truy cập (read, write, execute).
    - Nó yêu cầu đảm bảo quyền truy cập của kubeconfig file là 644 hoặc giới hạn truy cập nghiêm ngặt hơn.
      * Quyền truy cập 644 có nghĩa là tệp kubeconfig sẽ có quyền đọc và ghi cho người sở hữu và chỉ đọc cho nhóm và người dùng khác.
      * Tuy nhiên, nếu có nhiều người sử dụng trong cùng một nhóm và cần truy cập đến kubeconfig file thì việc giới hạn truy cập hơn sẽ được khuyến khích hơn.
  + **[PASS] 3.1.2 Ensure that the kubelet kubeconfig file ownership is set to root:root (Manual):** Đây là thông báo kết quả kiểm tra rằng quyền sở hữu của file kubeconfig của kubelet được thiết lập là root:root.
  + **[PASS] 3.1.3 Ensure that the kubelet configuration file has permissions set to 644 or more restrictive (Manual):** Đây là thông báo kết quả kiểm tra rằng quyền truy cập của file cấu hình của kubelet được thiết lập là 644 hoặc nghiêm ngặt hơn.
  + **[PASS] 3.1.4 Ensure that the kubelet configuration file ownership is set to root:root (Manual):** Đây là thông báo kết quả kiểm tra rằng quyền sở hữu của file cấu hình của kubelet được thiết lập là root:root.
* **[INFO] 3.2 Kubelet**: Đây là tiêu đề cho phần kiểm tra các cấu hình của kubelet.
  + **[PASS] 3.2.1 Ensure that the --anonymous-auth argument is set to false (Automated):** Đây là thông báo kết quả kiểm tra rằng argument --anonymous-auth của kubelet được thiết lập là false. (Automated có nghĩa là kiểm tra tự động).
    - --anonymous-auth là một cờ trong Kubernetes được sử dụng để cho phép hoặc không cho phép kubelet xác thực các yêu cầu API mà không cần xác thực danh tính của người dùng.
    - Khi được đặt thành true, nó sẽ cho phép kubelet chấp nhận các yêu cầu API từ các yêu cầu được xác thực với danh tính người dùng rỗng hoặc danh tính người dùng hệ thống.
    - Nếu nó được đặt thành false, kubelet sẽ yêu cầu xác thực danh tính trước khi chấp nhận các yêu cầu API. Nên để giảm thiểu các lỗ hổng bảo mật có thể xảy ra, nên đặt --anonymous-auth thành false.
  + **[PASS] 3.2.2 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated):** Đây là thông báo kết quả kiểm tra rằng argument --authorization-mode của kubelet không được thiết lập là AlwaysAllow. (Automated có nghĩa là kiểm tra tự động).
    - Trong Kubernetes, --authorization-mode là một tùy chọn được sử dụng để xác định cách xác thực và ủy quyền được thực hiện trên Kubelet.
    - Kubelet là một thành phần quan trọng của Kubernetes và chịu trách nhiệm quản lý các container trên các nút worker.
    - --authorization-mode xác định phương thức được sử dụng để xác định người dùng hoặc các đối tượng của hệ thống có quyền truy cập vào các API Kubelet.
    - Có nhiều chế độ ủy quyền được hỗ trợ trong Kubernetes, bao gồm :
      * AlwaysDeny,
      * AlwaysAllow,
      * ABAC (Attribute-Based Access Control),
      * RBAC (Role-Based Access Control).
        + RBAC là phương thức ủy quyền được khuyến khích để sử dụng trong Kubernetes vì nó cung cấp khả năng ủy quyền cụ thể cho các đối tượng trong hệ thống.
        + Nó cho phép quản trị viên chỉ định các quyền cụ thể cho người dùng hoặc nhóm người dùng để thực hiện các hành động cụ thể trên các đối tượng trong hệ thống Kubernetes.
  + **[PASS] 3.2.3 Ensure that the --client-ca-file argument is set as appropriate (Manual):** Đây là thông báo kết quả kiểm tra rằng argument --client-ca-file của kubelet được thiết lập đúng cách. (Manual có nghĩa là yêu cầu thủ công).
    - --client-ca-file là một tùy chọn cho phép xác thực các yêu cầu API trên một mạng TLS.
    - Khi --client-ca-file được thiết lập, Kubelet sẽ yêu cầu các yêu cầu API kết nối đến nó phải được xác thực bằng cách sử dụng một chứng chỉ TLS đã được ký bởi CA được chỉ định trong tệp --client-ca-file. Nếu yêu cầu không được xác thực, Kubelet sẽ từ chối chúng.
    - Điều này giúp đảm bảo rằng các yêu cầu API chỉ được thực hiện bởi các thành phần hợp lệ và được ủy quyền cho việc thực hiện các tác vụ cụ thể.
  + **[PASS] 3.2.4 Ensure that the --read-only-port argument is set to 0 (Manual):** kiểm tra xem đối số --read-only-port có được đặt thành 0 không.
    - Đối số --read-only-port của kubelet được sử dụng để thiết lập cổng mà Kubelet sẽ lắng nghe các yêu cầu chỉ đọc trên đó. Nó cho phép quản trị viên yêu cầu thông tin và trạng thái của nút từ xa mà không có quyền ghi.
    - Các yêu cầu chỉ đọc bao gồm các thông tin như:
      * số lượng pod đang chạy,
      * các tài nguyên được sử dụng và còn lại trên nút,
      * các sự kiện hệ thống và nhiều hơn nữa.
    - Quản trị viên có thể thiết lập đối số này để giới hạn quyền truy cập chỉ đọc của các yêu cầu từ xa.
    - Mặc định, giá trị của đối số này là 0, điều này có nghĩa là không có cổng chỉ đọc nào được mở.
  + **[PASS] 3.2.5 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Manual):** kiểm tra xem đối số --streaming-connection-idle-timeout có được đặt thành 0 không.
    - Đối số --streaming-connection-idle-timeout là một cờ trong cấu hình Kubelet để thiết lập thời gian tối đa mà một kết nối liên tục có thể chờ đợi trước khi bị đóng.
    - Nếu không có tín hiệu hoạt động nào được gửi qua kết nối trong khoảng thời gian này, kết nối sẽ bị đóng để giải phóng tài nguyên.
    - Giá trị mặc định của tham số này là 0, tức là không có thời gian chờ đợi, tuy nhiên, trong môi trường thực tế, nên thiết lập thời gian chờ đợi để giảm thiểu tình trạng kết nối bị giữ lại lãng phí tài nguyên.
  + **3.2.6**: Kiểm tra xem đối số "--protect-kernel-defaults" có được đặt thành "true" hay không. Kiểm tra này được thực hiện tự động và kết quả là "PASS", có nghĩa là đối số này được đặt thành "true".
    - Đối số "--protect-kernel-defaults" là một tùy chọn của kubelet trong Kubernetes, nó được sử dụng để bảo vệ cài đặt mặc định của kernel trên máy chủ worker node.
    - Nếu tùy chọn này được đặt thành "true", các tham số kernel mặc định sẽ được giữ nguyên và không bị thay đổi. Điều này giúp đảm bảo tính ổn định và an toàn của hệ thống và tránh xảy ra sự cố khi các tham số kernel mặc định bị thay đổi.
  + **3.2.7**: Kiểm tra xem đối số "--make-iptables-util-chains" có được đặt thành "true" hay không. Kiểm tra này được thực hiện tự động và kết quả là "PASS", có nghĩa là đối số này được đặt thành "true".
    - Đối số --make-iptables-util-chains là một tùy chọn được sử dụng trong kubelet, đảm bảo rằng kubelet sử dụng các chains cụ thể cho việc liên kết tới iptables.
    - Khi chạy kubelet, nếu tùy chọn này được đặt thành true, kubelet sẽ sử dụng các chains riêng để tạo các rule iptables của mình thay vì sử dụng chains chung của hệ thống.
    - Việc sử dụng các chains cụ thể này giúp tránh việc xung đột với các rule iptables do các ứng dụng khác tạo ra.
  + **3.2.8**: Kiểm tra xem đối số "--hostname-override" có được đặt hay không. Kiểm tra này được thực hiện thủ công và kết quả là "PASS", có nghĩa là đối số này không được đặt.
    - Đối số "--hostname-override" trong kubelet là một tùy chọn dòng lệnh dùng để ghi đè lên tên máy chủ (hostname) của worker node trong các yêu cầu đến API server của Kubernetes. Thường được sử dụng để cấu hình địa chỉ IP của worker node nếu nó không được xác định đúng bởi hệ thống hoặc để thay đổi hostname hiện tại của worker node.
    - Tuy nhiên, nó có thể gây ra những vấn đề liên quan đến bảo mật và ổn định nếu sử dụng không đúng cách. Do đó, việc sử dụng đối số này cần được cân nhắc kỹ lưỡng và chỉ nên được sử dụng khi thực sự cần thiết.
  + **3.2.9**: Kiểm tra xem đối số "--eventRecordQPS" có được đặt thành "0" hoặc mức độ nào đó để đảm bảo việc ghi lại sự kiện được thực hiện đúng cách hay không. Kiểm tra này được thực hiện tự động và kết quả là "WARN", có nghĩa là đối số này không được đặt thành "0" hoặc mức độ đảm bảo việc ghi lại sự kiện không được thiết lập đúng.
  + **3.2.10**: Kiểm tra xem đối số "--rotate-certificates" có được đặt thành "false" hay không. Kiểm tra này được thực hiện thủ công và kết quả là "PASS", có nghĩa là đối số này không được đặt thành "false".
    - Đối số "--rotate-certificates" trong Kubernetes là một tùy chọn để cho phép Kubernetes tự động thay thế chứng chỉ TLS cho các kết nối mạng bảo mật giữa các thành phần của hệ thống.
    - Khi tùy chọn này được bật, kubelet sẽ đăng ký cho chuyển đổi chứng chỉ một cách định kỳ, để đảm bảo tính toàn vẹn và bảo mật của hệ thống Kubernetes.
  + **3.2.11**: Kiểm tra xem đối số "RotateKubeletServerCertificate" có được đặt thành "true" hay không. Kiểm tra này được thực hiện thủ công và kết quả là "PASS", có nghĩa là đối số này được đặt thành "true".
    - RotateKubeletServerCertificate là một cấu hình trong Kubernetes được sử dụng để quản lý các chứng chỉ TLS cho Kubelet trên các node worker.
    - Khi cấu hình này được thiết lập thành true, Kubelet sẽ tạo ra một cặp khóa công khai / riêng tư mới và yêu cầu chứng chỉ ký cho khóa mới này từ API server.
    - Sau đó, Kubelet sẽ sử dụng chứng chỉ mới này để xác thực với các thành phần khác trong cụm Kubernetes. Cấu hình này giúp đảm bảo rằng các chứng chỉ TLS được sử dụng bởi Kubelet được thay đổi định kỳ, tăng cường tính bảo mật của cụm Kubernetes.
  + **3.3**: Kiểm tra về Container Optimized OS (hệ điều hành được tối ưu hóa cho các môi trường container).
  + **3.3.1**: Kiểm tra về việc ưu tiên sử dụng Container Optimized OS khi có thể. Kiểm tra này được thực hiện thủ công và kết quả là "WARN", có nghĩa là nên ưu tiên sử dụng Container Optimized OS nếu có thể.
    - Container-Optimized OS là một hệ điều hành tối giản, nhẹ và được tối ưu hóa để chạy các container, được tạo ra bởi Google và được sử dụng rộng rãi trong các môi trường Kubernetes.
    - Hệ điều hành này được tối ưu hóa để chạy trên các máy chủ vật lý hoặc máy ảo và cung cấp một môi trường chạy Docker hoàn chỉnh để chạy các ứng dụng được đóng gói trong container.
    - Container-Optimized OS có kích thước nhỏ, ít cập nhật, tối ưu hóa bảo mật và có thể được triển khai dễ dàng trong các môi trường đám mây công cộng hoặc riêng tư

Text

Description automatically generated

* **Remediations node**: Các hướng dẫn để khắc phục các vấn đề đã được đưa ra.
  + Hướng dẫn này nhằm giải quyết cảnh báo "WARN 3.2.9 Ensure that the --eventRecordQPS argument is set to 0 or a level which ensures appropriate event capture (Automated)" trong phân tích.
  + Để giải quyết cảnh báo này, cần chỉnh sửa cấu hình của Kubelet để đảm bảo rằng đối số --eventRecordQPS được thiết lập thành 0 hoặc một mức độ phù hợp.
    - Có thể làm điều này bằng cách sửa đổi tệp cấu hình Kubelet hoặc tệp dịch vụ kubelet trên mỗi nút làm việc (worker node) theo hướng dẫn của đoạn remediation node.
    - Nếu đang sử dụng tệp cấu hình Kubelet, ta có thể chỉnh sửa tệp để đặt eventRecordQPS: thành một mức độ phù hợp.
    - Nếu đang sử dụng các tham số dòng lệnh, ta có thể chỉnh sửa tệp dịch vụ kubelet và thiết lập tham số dưới biến môi trường KUBELET\_SYSTEM\_PODS\_ARGS. Sau đó, cần khởi động lại dịch vụ kubelet để áp dụng các thay đổi bằng cách sử dụng các lệnh systemctl daemon-reload và systemctl restart kubelet.service

Text

Description automatically generated

* **Summary node**: Tóm tắt kết quả đánh giá cho các nút (node) trong cụm Kubernetes.
  + có tổng cộng 16 checks (3.1.1 đến 3.3.1):
    - 14 checks được xác nhận là đạt (PASS),
    - không có check nào thất bại (FAIL),
    - 2 checks báo cảnh báo (WARN)
    - không có check nào cung cấp thông tin (INFO).
* **Summary total**: Tóm tắt kết quả đánh giá cho toàn bộ cụm Kubernetes. (giống trên)

==========HẾT==========

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)