.

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN ĐIỆN TOÁN ĐÁM MÂY**

-----🙞🙜🕮🙞🙜-----

**Logo

Description automatically generated**

**NGUYỄN KIỀU CHÂU ANH - 20110234**

**TRẦN THỊ TRÀ MY - 20110324**

**PHAN TRỊ DIỄM TRINH - 20110193**

**Đề Tài: AWS CLOUD FORMATION VÀ ỨNG DỤNG**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. HUỲNH XUÂN PHỤNG**

**TP HCM, tháng 06 năm 2022**

**KHÓA 2020 - 2024**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ tên sv 1: Nguyễn Kiều Châu Anh Mã số sinh viên 1: 20110234

Họ tên sv 2: Trần Thị Trà My Mã số sinh viên 2: 20110324

Họ tên sv 3: Phan Thị Diễm Trinh Mã số sinh viên 3: 20110193

Ngành: Công nghệ thông tin

Đề tài: **Aws Cloud Formation Và Ứng Dụng**

Giáo viên hướng dẫn: **TS. Huỳnh Xuân Phụng**

**LỜI NHẬN XÉT CỦA GIẢNG VIÊN**

1. Về nội dung đề tài & khối lượng thực hiện được trong đề tài:

2. Ưu điểm của đề tài:

3. Khuyết điểm cần sửa đổi trong đề tài:

5. Đánh giá xếp loại:

6. Điểm:

|  |  |  |
| --- | --- | --- |
| *Tp. HCM, ngày 10 tháng 12 năm 2022* | | |
|  | Giáo viên hướng dẫn  *(Ký & ghi rõ họ tên)* |

# LỜI CẢM ƠN

*Đầu tiên nhóm chúng em xin được phép gửi lời cảm ơn chân thành đến với Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện tốt nhất cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc từ đó có thể được đề tài kết thúc môn học này.*

*Bên cạnh đó nhóm chúng em xin gửi đến thầy Huỳnh Xuân Phụng lời cảm ơn sâu sắc nhất trong suốt thời gian dài học tập cũng như thực hiện đề tài trong thời gian qua.*

*Với sự chỉ bảo nhiệt tình, giảng dạy tận tình và đầy đủ kiến thức của thầy Huỳnh Xuân Phụng, nhóm chúng em đã học tập và hiểu được những kiến thức cơ bản về AWS Cloud formation – một dịch vụ hỗ trợ lập mô hình và thiết lập các tài nguyên AWS để bạn có thể dành ít thời gian hơn cho việc quản lý các tài nguyên đó và có nhiều thời gian hơn để tập trung vào các ứng dụng chạy trong AWS. Một lần nữa, nhóm xin cảm ơn thầy!*

*Tuy nhiên lượng kiến thức là vô tận và với khả năng còn nhiều hạn chế, chúng em đã cố gắng hết sức để hoàn thành đề tài một cách tốt nhất có thể. Chính vì vậy, việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Nhóm chúng em hi vọng nhận được sự góp ý, nhận xét và phê bình tận tình của quý thầy (cô)để có thể rút ra được các bài học kinh nghiệm từ đó hoàn thiện và cải thiện nâng cấp lại đề tài của mình một cách tốt nhất có thể.*

*Nhóm chúng em xin chân thành cảm ơn quý thầy(cô)!*

***Sinh viên thực hiện đồ án***

**MỤC LỤC**

[**LỜI CẢM ƠN 3**](#_heading=h.1fob9te)

[**PHẦN MỞ ĐẦU 5**](#_heading=h.tyjcwt)

[**1.**](#_heading=h.3dy6vkm) **Tính cấp thiết của đề tài** 5

[**2.**](#_heading=h.1t3h5sf) **Đối tượng nghiên cứu** 5

[**3.**](#_heading=h.4d34og8) **Phạm vi nghiên cứu** 5

[**4.**](#_heading=h.2s8eyo1) **Kết quả dự kiến đạt được** 5

[**PHẦN NỘI DUNG 6**](#_heading=h.17dp8vu)

[**Chương 1 : CƠ SỞ LÝ THUYẾT VỀ APACHE HIVE 6**](#_heading=h.3rdcrjn)

[**1.**](#_heading=h.26in1rg) **Hive là gì ?** 6

[**2.**](#_heading=h.lnxbz9) **Ra đời giải quyết vấn đề gì ?** 6

[**3.**](#_heading=h.35nkun2) **Các tính năng của Hive** 6

[**4.**](#_heading=h.1ksv4uv) **Kiến trúc của Hive** 6

[**5.**](#_heading=h.44sinio) **Quy trình làm việc của Hive** 7

[**Chương 2 : CÀI ĐẶT, THIẾT KẾ VÀ XÂY DỰNG DATA 9**](#_heading=h.2jxsxqh)

**[1.](#_heading=h.z337ya)****Cài đặt môi trường Hadoop** 9

[**2.**](#_heading=h.3j2qqm3) **Cài đặt Hive** 23

[**3.**](#_heading=h.1y810tw) **Cài đặt AWS EMR** 27

[**4.**](#_heading=h.4i7ojhp) **Sử dụng HUE để tạo cơ sở dữ liệu và thực hiện các truy vấn** 31

[**PHẦN KẾT LUẬN 42**](#_heading=h.2xcytpi)

[**1.**](#_heading=h.1ci93xb) **Kết quả đạt được** 42

[**2.**](#_heading=h.3whwml4) **Ưu điểm** 42

[**3.**](#_heading=h.2bn6wsx) **Nhược điểm** 42

[**4.**](#_heading=h.qsh70q) **Hướng phát triển** 42

[**TÀI LIỆU THAM KHẢO 43**](#_heading=h.3as4poj)

# PHẦN MỞ ĐẦU

1. **Tính cấp thiết của đồ án**

Cùng với sự phát triển và phổ biến ngày càng rộng rãi của Điện toán đám mây, các nền tàng ứng dụng xây dựng trên nền tàng điện toán đám mây đòi hỏi một cấu hình mẫu đáng tin cậy, có thể sửa đổi linh hoạt để phù hợp với nhu cầu và mục tiêu thực tế, có thể tái sử dụng nhằm tiết kiệm thời gian cấu hình cũng như hạn chế những sai sót có thể xảy ra trong quá trình cấu hình thủ công. Xuất phát từ yêu cầu thực tế trên, Dịch vụ Cloud Formattion được AWS cho ra đời nhằm cung cấp cho các nhà phát triển dự án một công cụ mạnh mẽ nhưng không kém phần linh hoạt để tạo ra các template có thể dễ chỉnh sửa, quản lý tài nguyên và tái sử dụng cho các dự án tương tự. Nói đơn giản thì, template là một bản thiết kế chi tiết về cấu hình cần thiết cho dự án của bạn. Chỉ cần xác định tất cả tài nguyên cần tạo vào một bản thiết kế chi tiết này, và nhấn nút tạo stack, AWS sẽ giúp bạn phần còn lại.

1. **Đối tượng nghiên cứu**

Trong phạm vi đề tài, nhóm sẽ tập trung tìm hiểu về các lý thuyết nền tảng liên quan tới Cloud formation, áp dụng kiến thức học được để thiết kế hai template cơ bản cho website tĩnh và động.

1. **Phạm vi nghiên cứu**

* Lý thuyết về cách hoạt động của AWS Cloud formation
* Ứng dụng thực tế của AWS Cloud formation

1. **Kết quả dự kiến đạt được**

* Hiểu được cách thức hoạt động cũng như lý thuyết liên quan đến AWS Cloud Formation
* Đề xuất template cho website tĩnh và website động

# PHẦN NỘI DUNG

# Chương 1 : LÝ THUYẾT VỀ AWS CLOUD FORMATION

1. **AWS Cloud Formation là gì ? (Trinh)**

AWS CloudFormation là dịch vụ hỗ trợ việc thiết lập các tài nguyên AWS để có thể dành ít thời gian hơn cho việc quản lý các tài nguyên đó và có nhiều thời gian hơn để tập trung vào các ứng dụng chạy trong AWS. Chúng ta tạo một mẫu mô tả tất cả các tài nguyên AWS mà mình muốn (chẳng hạn như phiên bản Amazon EC2 hoặc phiên bản Amazon RDS DB) và CloudFormation sẽ đảm nhận việc cung cấp và định cấu hình các tài nguyên đó . Chúng ta không cần phải tạo và định cấu hình tài nguyên AWS riêng lẻ cũng như tìm ra cái gì phụ thuộc vào cái gì; CloudFormation xử lý việc đó.

1. **AWS Cloud Formation hoạt động như thế nào? (Trinh)**

Mỗi khi chúng ta tạo một stack, AWS CloudFormation truyền lệnh tới các dịch vụ chạy ngầm của AWS để cung cấp và thiết lập các tài nguyên (thực hiện các lệnh nằm trong quyền hạn của bạn

Tạo instance Amazon EC2 : Cần có quyền tạo instance. Xóa stack (cần có quyền tắt instance)

Sử dụng [**AWS Identity and Access Management**](http://docs.aws.amazon.com/IAM/latest/UserGuide/) để quản lý việc phân quyền.

1. Tạo một template (document định dạng JSON) trong một text editor hoặc lấy một template có sẵn. Ví dụ, nếu bạn muốn tạo một Amazon EC2 instance. Bạn sử dụng template có dạng như sau:



1. Nếu chúng ta tạo template, chúng ta có thể đặt tên file với đuôi mở rộng bất kì như .json, .txt...Có thể lưu file ở máy tính cá nhân hoặc lưu trên Amazon S3 bucket.
2. Chúng ta tạo AWS CloudFormation stack và chỉ ra địa chỉ của file template. Địa chỉ đó có thể là một file trên máy tính local hoặc một URL của Amazon S3. Bạn có thể tạo stack bằng cách sử dụng AWS CloudFormation [console](http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-create-stack.html), [API](http://docs.aws.amazon.com/AWSCloudFormation/latest/APIReference/API_CreateStack.html), [AWS CLI](http://docs.aws.amazon.com/cli/latest/reference/cloudformation/create-stack.html).

**Chú ý**: Nếu sử dụng một file template nằm ở local, AWS CloudFormation cũng vẫn upload nó lên một Amazoon S3 bucket trong AWS account của bạn. AWS CloudFormation tạo ra một bucket duy nhất cho mỗi region nhằm mục đích phục vụ cho việc upload file template. Bucket này có thể được truy nhập bởi mọi user với quyền truy nhập S3 trong account của bạn. Nếu bucket được tạo ra bởi AWS CloudFormation đã có từ trước, file template sẽ được đưa vào bucket đó.

Chúng ta có thể sử dụng bucket của riêng bạn và điều chỉnh quyền của nó để upload file template một cách thủ công vào Amazon S3. Sau đó, mỗi khi tạo ra hoặc cập nhật một stack, bạn chỉ ra cả Amazon S3 URL của file template.

AWS CloudFormation cung cấp và thiết lập các tài nguyên bằng cách gọi tới các AWS service được mô tả trong template. Sau khi tất cả các resource đã được tạo, AWS CloudFormation truyền tín hiệu thông báo stack đã được tạo thành công. Sau đó bạn có thể sử dụng tất cả các resource trong stack. Nếu gặp lỗi trong việc tạo ra stack, AWS CloudFormation sẽ roll-back tất cả các thay đổi và xóa mọi tài nguyên vừa được tạo ra.

1. **Templates, Stacks, Stack sets**

**3.1. Templates (My)**

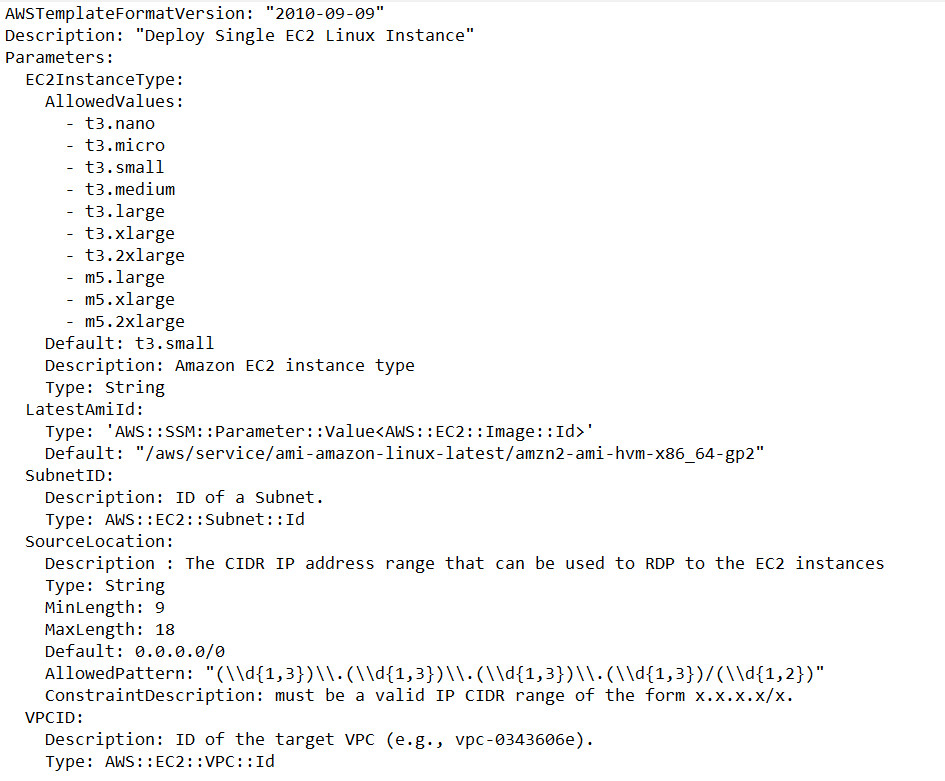
**3.1.1. Khái niêm:**

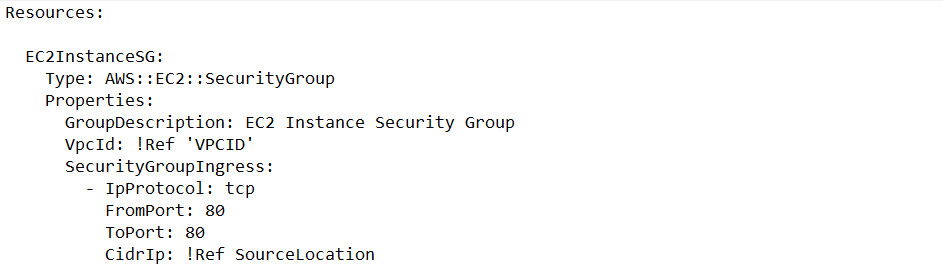
Một template là một khai báo về các tài nguyên AWS tạo nên một ngăn xếp. Template được lưu trữ dưới dạng tệp văn bản có định dạng JSON hoặc tiêu chuẩn YAML. Vì template một tệp văn bản nên chúng ta có thể tạo và chỉnh sửa trong bất kỳ trình soạn thảo văn bản nào và quản lý chúng trong hệ thống hệ thống điều khiển nguồn của bạn với phần còn lại trong source code của bạn.

**3.1.2. Các thành phần của một template**

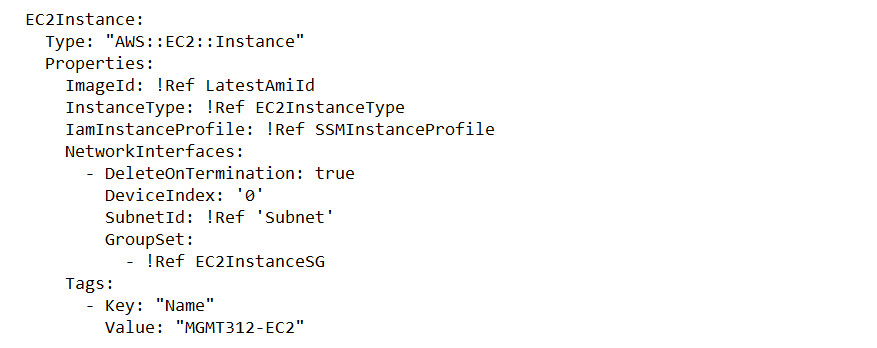
* **AWSTemplateFormatVersion:** Là phiên bản template tuân theo. Phiên bản định dạng template không giống với phiên bản API hoặc WSDL. Phiên bản định dạng template có thể thay đổi độc lập với phiên bản API và WSDL.
* **Description:** Một chuỗi văn bản mô tả temlate. Phần này phải luôn tuân theo phần phiên bản định dạng template.
* **Paramters:** Các giá trị để chuyển đến template trong thời gian chạy (khi bạn tạo hoặc cập nhật stack). Bạn có thể tham khải các thông số từ phần Resource và Outputs của template
* **Metadatas:** Các đối tượng cung cấp thông tin bổ sung về template
* **Rules:** Xác thực một parameter hoặc tổ hợp các parameter được truyền cho template trong quá trình tạo stack hoặc cập nhật stack
* **Mappings:** ánh xạ các khóa và các giá trị liên quan mà bạn có thể sử dụng để chỉ định các giá trị tham số có điểu kiện, tương tự như bảng tra cứu. Chúng ta có thể khớp một khóa với giá trị tương ứng bằng cách sử dụng hàm nội tại FN : : FindInmap trong các Resource và Outputs
* **Conditions:** Các điều kiện kiểm soát xem một số tài nguyên nhất định có được tại hay liệu các thuộc tính tài nguyên nhất định có được gán giá trị trong quá trình tại hoặc cập nhật stack hay không. Ví dụ: chúng ta có thể tạo điều kiện tạo mọt tài nguyên phụ thuộc vào việc stack cho môi trường sản xuất hoặc thử nghiệm.
* **Transform:** Đối với các ứng dụng không có máy chủ (còn được gọi là các ứng dụng dựa trên LAMBDA), chỉ định phiên bản của mô hình ứng dụng AWS Serverless (AWS SAM) để sử dụng. Khi bạn chỉ định một biến đổi, bạn có thể sử dụng cú pháp AWS SAM để khai báo tài nguyên trong mẫu của mình. Mô hình xác định cú pháp mà bạn có thể sử dụng và cách xử lý. Bạn cũng có thể sử dụng AWS :: bao gồm các biến đổi để hoạt động với các đoạn mã được lưu trữ riêng biệt với mẫu AWS CloudFormation chính. Bạn có thể lưu trữ các tệp đoạn trích của mình trong thùng Amazon S3 và sau đó sử dụng lại các chức năng trên nhiều mẫu.
* **Resources:** Chỉ định tài nguyên stack và thuộc tính của chúng, chẳng hạn như phiên bản Amazon Elastic Compute Cloud hoặc bộ chứa Amazon Simple Storage Service. Chúng ta có thể tham khảo các tài nguyên trong phần Resource và Outputs
* **Outputs:** Mô tả các giá trị được trả về bất cứ khi nào chúng ta xem các thuộc tính của stack. Ví dụ: chúng ta có thể khai báo Outputs cho tên bộ chứa S3, sau đó gọi lệch AWS CLI description-stacks aws cloudformation để xem tên

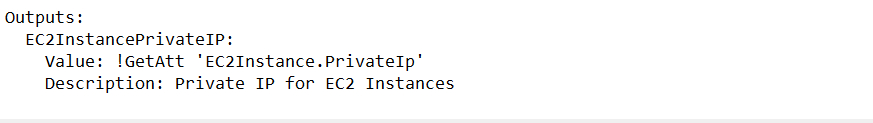
**3.1.3. Ví dụ một template**

****

****

****

****

****

* Khi khởi tạo stack, một stack sẽ được tạo ra dựa trên mô tả của template trên dùng để quản lý các tài nguyên được khai báo trong phần ‘Resources’ của template. Các tài nguyên này gồm có, một EC2 Instance, Security group (cho phép lưu lượng truy cập trên cổng 80), một Instance role (tạo ra một role để gán cho EC2 instance cho phép gửi thông tin giám sát và giao tiếp qua SSM), và một instance profile (được tạo ra để tham chiếu tới instance role). Chi tiết cấu trúc của một template sẽ được trình bày chi tiết hơn phía dưới.

Link tham khảo[1] <https://000037.awsstudygroup.com/vi/3-cloudformationbasic/3.2-createcloudformationtemplate/?fbclid=IwAR2Eb_C-wgMp_7G0H-_SGY58DfOQUXXHVUaleMI7Wl3k_wgVbwPHPZzGDNc>

[2] <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-anatomy.html>

**Link tham khảo:** [**https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html**](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html)

**3.2. Stacks (Trinh)**

Stacks (Ngăn xếp) là tập hợp các tài nguyên AWS mà bạn có thể quản lý dưới

dạng một đơn vị. Nói cách khác, bạn có thể tạo, cập nhật hoặc xóa bộ sưu tập tài nguyên bằng cách tạo, cập nhật hoặc xóa ngăn xếp. Tất cả tài nguyên trong ngăn xếp được xác định bởi mẫu AWS CloudFormation của ngăn xếp. Ví dụ, ngăn xếp có thể bao gồm tất cả các tài nguyên cần thiết để chạy ứng dụng web, chẳng hạn như máy chủ web, cơ sở dữ liệu và quy tắc mạng. Nếu bạn không còn yêu cầu ứng dụng web đó nữa, bạn chỉ cần xóa ngăn xếp và tất cả các tài nguyên liên quan của nó sẽ bị xóa.

**Link tham khảo:** [**https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html**](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html)

**3.3.**  **Stack sets (Trinh)**

AWS CloudFormation StackSets mở rộng khả năng của ngăn xếp bằng cách cho phép chúng ta tạo, cập nhật hoặc xóa ngăn xếp trên nhiều tài khoản và Khu vực AWS chỉ bằng một thao tác. Khi sử dụng tài khoản quản trị viên, chúng ta xác định và quản lý mẫu AWS CloudFormation, đồng thời sử dụng mẫu đó làm cơ sở để cung cấp ngăn xếp cho các tài khoản mục tiêu đã chọn trên các khu vực AWS được chỉ định.

**Graphical user interface

Description automatically generated with medium confidence**

**Link tham khảo:** [**https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cfnstacksets.html**](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cfnstacksets.html)

1. **Lợi ích khi sử dụng AWS Cloud Formation (Trinh)**

**Lợi ích của AWS :**

* **Simplify infrastructure management (đơn giản hóa trong việc quản lý cơ sở hạ tầng )**

Việc sử dụng ứng dụng web bao gồm các Auto Scaling group, an Elastic Load Balancing load balancer, và an Amazon Relational Database Service database instance. Chúng là các dịch vụ riêng lẻ và khi sử dụng bạn phải cấu hình để chúng có thể hoạt động cùng nhau à phức tạp tốn thời gian hơn trong quá trình làm việc của bạn. Thay vào đó bạn có thể tạo mẫu **CloudFormation** hoặc sửa đổi mẫu trước đó. Mẫu mô tả tất cả các tài nguyên của bạn để tạo các CloudFormation stack, CloudFormation provisions the Auto Scaling group, load balancer và cơ sở dữ liệu. Sau khi được tạo chúng ta có thể dễ dàng thao tác và quản lý các tài nguyên trên một tập duy nhất

* **Quickly replicate your infrastructure( Nhanh chóng tái tạo cơ sở hạ tầng )**

Các mẫu AWS Formation mà bạn đã tạo bạn tạo có thể sử dụng được nhiều lần thay vì mỗi lần sử dụng bạn phải thiết lập thủ công lại ban đầu. Việc sử dụng mẫu giúp cho việc tạo tài nguyên một cách nhất quán, thuận tiện cho cho phép sao chép nó ở nhiều khi vực mà không cần cung cấp lại những tài nguyên giống nhau cũng như tái tao lại ứng dụng ban đầu .

* **Easily control and track changes to your infrastructure (dễ dàng kiểm soát và theo dõi các thay đổi đối với cơ sở hạ tầng của bạn )**

Có thể mở rộng các cơ sở hạ tầng của mình một cách nhanh chóng. Qua việc sử dụng AWS Formation, bạn cso thể thêm các phiên bản máy ảo hoặc tăng không gian lưu trữ. Nếu trong quá trình sử dụng khi lượng truy cập tăng bạn chỉ cần mở rộng môi trường. Song song đó vẫn có thể thu nhỏ lại quy mô để tiết kiệm chi phí .

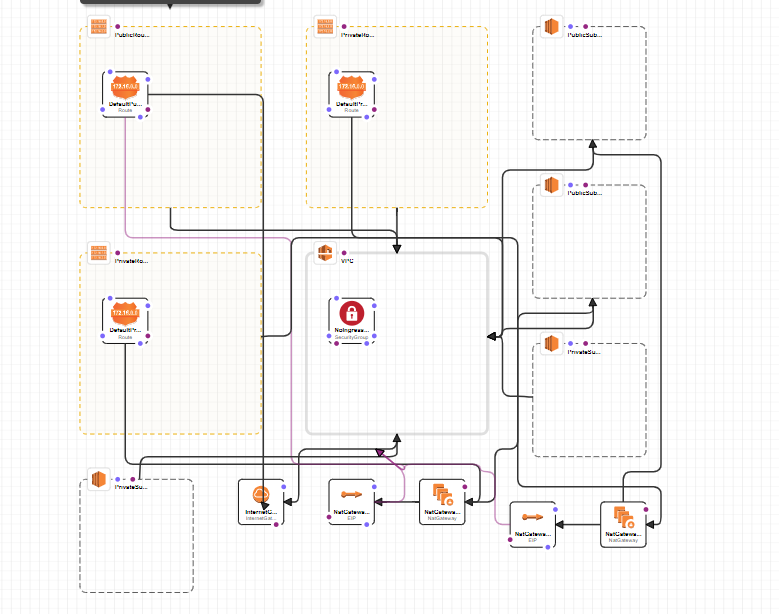
Trong các trường hợp tài cơ sở mà bạn đang triển khai có xu hướng phát triển và muốn nâng cấp lên dần. Việc sử dụng CloudFormation sẽ giúp chúng ta đơn giản hơn trong việc mô tả chính xác những tài nguyên nào được cung cấp và cài đặt bởi chúng. Các mẫu là các tệp văn bản,  nên trong quá trình mở rộng quy mô có sự cố chúng ta chỉ cần theo dõi sự thay dổi sự khác biệt trong các tệp này để kiểm soát khôi phục lại lỗi. Ví dụ : bạn có thể kiểm soát các mẫu của mình nếu có sự thay đổi, biết được chính xác thời gian và tác nhân gây ra lỗi để quay lại sử dụng bản mẫu trước khi sảy ra sự cố.

# Chương 2 : THIẾT KẾ VÀ CÀI ĐẶT MỘT SỐ TEMPLATES

Các template dưới đây dưới dạng file .json hoặc .yaml. Các files này có thể viết trên bất kì text editor nào và sẽ được deploy khi tạo stack trong dịch vụ Cloud formation được cung cấp bởi AWS.

1. **VPC** 
   1. **Mô tả:**

Template cho một VPC (virtual private clouod) với mạng con public và private trên hai vùng khả dụng (Availability zones/ AZ). Đồng thời, template sẽ cấu hình cổng mạng (internet gateway) với một route mặc định trên mạng con public đã tạo, một NAT gateway cho mỗi vùng khả dụng với route mặc định nằm trên các mạng con private.



* 1. **Các parameters:**

Khai báo các thông tin về tham số trong phần ‘Parameters’ của template. Một số tham số được gắn giá trọ mặc định sẵn như sau:

**Text

Description automatically generated**

* EnvironmentName (môi trường hoạt động do người dùng nhập vào)
* VpcCIDR (IP range) VPC: 10.192.0.0/16 (mặc định)
* PublicSubnet1CIDR (IP range cho mạng con public của vùng khả dụng đầu tiên): 10.192.10.0/24
* PublicSubnet1CIDR (IP range cho mạng con public của vùng khả dụng thứ hai): 10.192.11.0/24
* PrivateSubnet1CIDR (IP Range cho mạng con private của vùng khả dụng thứ nhất): 10.192.20.0/24
* PrivateSubnet2CIDR (IP Range cho mạng con private của vùng khả dụng thứ hai): 10.192.21.0/24
  1. **Resources**

Các tài nguyên cần thiết để tạo và cấu hình VPC được khai báo trong phần ‘Resources’ như dưới đây. Các tài nguyên được khai báo sẽ được cấu hình trong phần Properties. **Các tài nguyên và cấu hình của các tài nguyên này như sau:**

* **VPC:**
* CidrBlock: IP range của VPC (được tham chiếu tới tham số VpcCIDR được tham báo ở phần ‘Parameters’)
* Cho phép hỗ trợ DNS và DNS hostname
* **InternetGateway**: có môi trường hoạt động là tham số EnvironmentName do người dùng nhập vào
* InternetGgatewayAttachment:
* InternetGatewayId: được tham chiếu tới id của InternetGateway
* VpcId: được tham chiếu tới id của VPC
* **PublicSubnet1:**
* VpcId: được tham chiếu tới id của VPC
* Availability zone: được chọn mặc định ở đây là region đầu tiên trong danh sách các AZ lấy được từ **!GetAZs**
* CidrBlock (IP Range): được tham chiếu tới tham số PublicSubnet1CIDR được khai báo ở phía trên
* Cho phép MapPublicIpOnLaunch để gán giá trị địa chỉ puclic IP, khi instance chạy trên subnet đang xét.

**Text

Description automatically generated**

* **PublicSubnet2**: Cấu hình các properties tương tự PublicSubnetId1
* **PrivateSubnet1**: Cấu hình các propeties tương tự PublicSubnetId1. Tuy nhiên không cho phép (**false**) MapPublicIpOnLaunch
* **PrivateSubnet2:** Cấu hình các propeties tương tự PublicSubnetId1. Tuy nhiên không cho phép (**false**) MapPublicIpOnLaunch

**Text

Description automatically generated**

* **NatGateway1EIP:**
* Phụ thuộc vào InternetGateway Attachment (khi tạo stack, thì sẽ tạo InternetGateway Attachment trước, sau đó mới tạo NateGateway1EIP)
* Hoạt động trên domain vpc
* **NatGateway2EIP:** Cấu hình các properties tương tự như NatGateway1EIP
* **NatGateway1:**
* AllocationId: Tham chiếu tới thuộc tính AllocationId của NatGateway1EIP
* SubnetId: Tham chiếu tới PublicSubnet1
* **NatGateway2:** Cấu hình các properties tương tự như NateGateway1EIP
* **PublicRouteTable:**
* VpcId: tham chiếu tới id của VPC

**Text

Description automatically generated**

* **DefaultPublicRoute:**
* Phụ thuộc vào InternetGatewayAttachment
* RouteTableId: tham thiếu tới id của PublicRouteTable
* DestinationCidrBlock: 0.0.0.0/0
* GatewayId: tham chiếu tới id của InternetGateway
* **PublicSubnet1RouteTableAssociation:**
* RouteTableId: tham chiếu tới id của PublicRouteTable
* SubnetId: tham chiếu tới id của PublicSubnet1
* **PublicSubnet1RouteTableAssociation:**

Cấu hình các properties tương tự như PublicSubnet1RouteTableAssociation

* **PrivateRouteTable:**
* VpcId: được tham chiếu tới id của VPC

**Text

Description automatically generated**

* **DefaultPrivateRoute1:**
* RouteTableId: tham chiếu tới id của PrivateRouteTable1
* DestinationCidrBlock: 0.0.0.0/0
* NatGatewayId: tham chiếu tới id của NatGateway1
* **PrivateSubnet1RouteTableAssociation:**
* RouteTableId: tham chiếu tới id của PrivateRouteTable1
* SubnetId: tham chiếu tới id của PrivateSubnet1
* **PrivateRouteTable2:**
* VpcId: tham chiếu tới id của VPC
* **DefaultPrivateRoute2:**
* RouteTableId: tham chiếu tới id của PrivateRouteTable2
* DestinationCidrBlock: 0.0.0.0/0
* NatGatewayId: tham chiếu tới id của NatGateway1
* **PrivateSubnet2RouteTableAssociation:**
* RouteTableId: tham chiếu tới id của PrivateRouteTable2
* SubnetId: tham chiếu tới id của PrivateSubnet2

**Text

Description automatically generated**

* **noIngressSecurityGroup:**
* GroupName: “no-ingress-sg”
* VpcId**:** tham chiếu tới id của VPC
* Mô tả: Tạo ra một security group không chứa bất kì rule nào

**Text

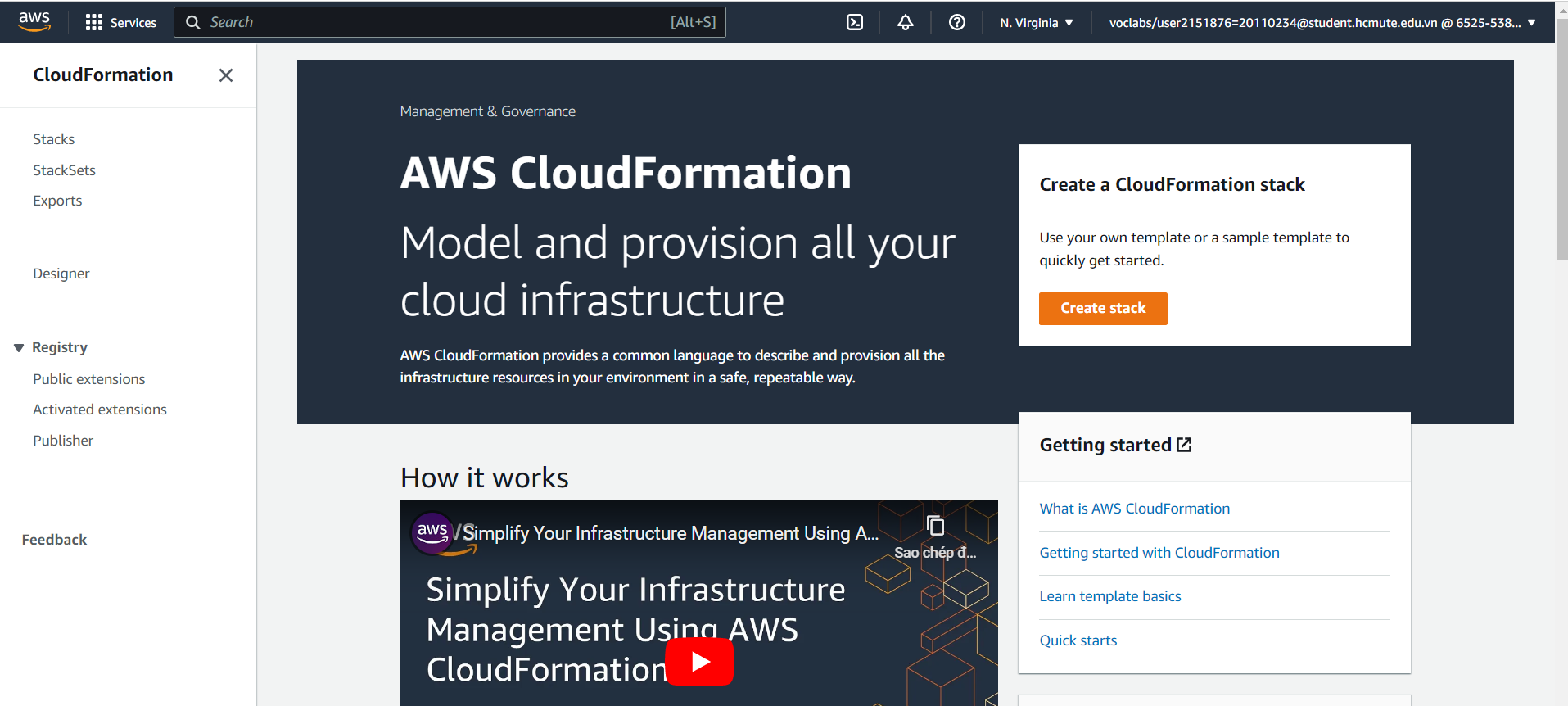
Description automatically generated**

* 1. **Outputs**

**Text

Description automatically generated**

* 1. **Tạo Stack**
     1. Truy cập vào tài khoản AWS
     2. Mở dịch vụ Cloud Formation lên
     3. Chọn **Create Stack**



* + 1. Trong giao diện hiện ra, chọn **Upload a template file,** sau đó Next

**Graphical user interface, text, application, email

Description automatically generated**

* + 1. Thiết lập thông số cho các Parameter như hình

Graphical user interface, text, application, email

Description automatically generated

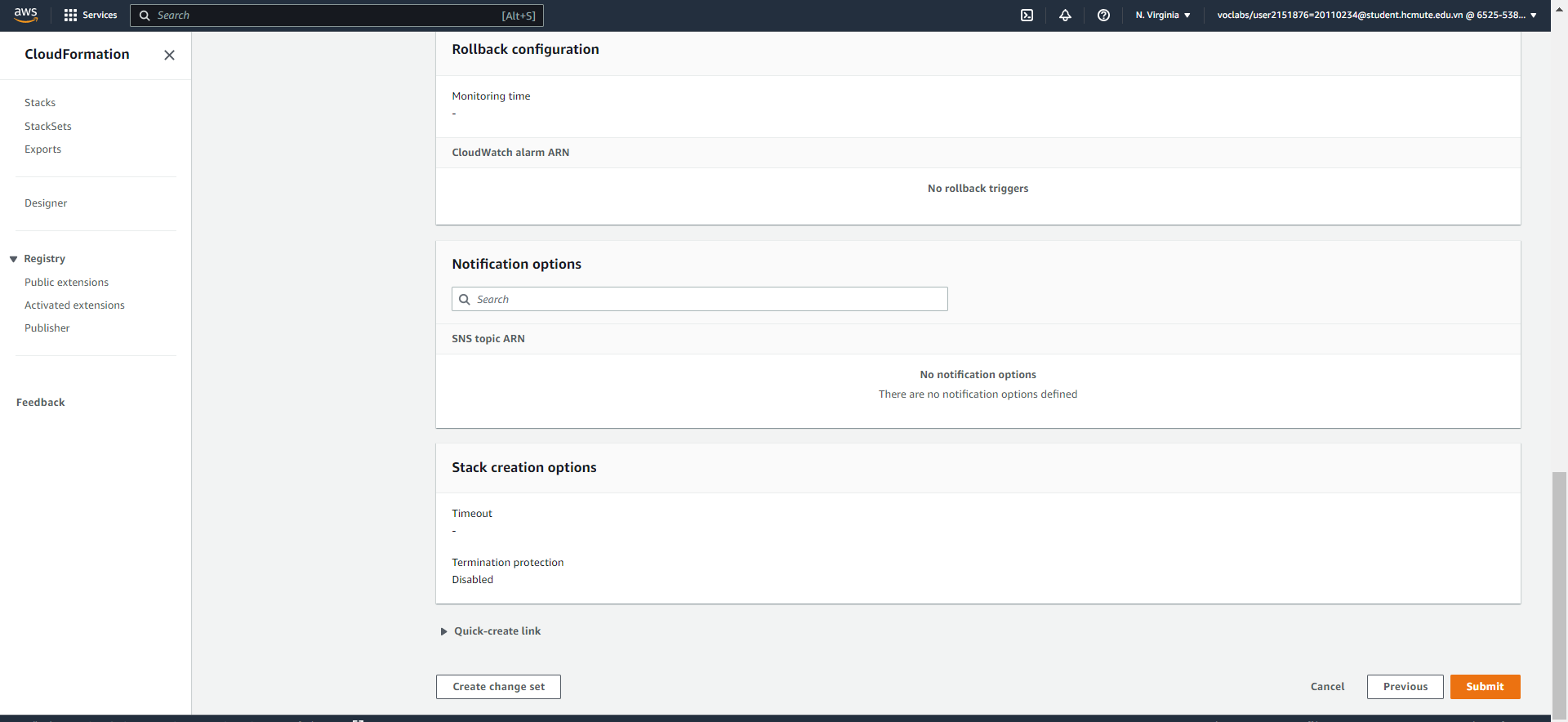
Graphical user interface, text, application, email

Description automatically generated

* + 1. Kiểm tra template lại một lần nữa rồi nhấn nút **Submit** để tạo Stack

Graphical user interface, text, application, email

Description automatically generated

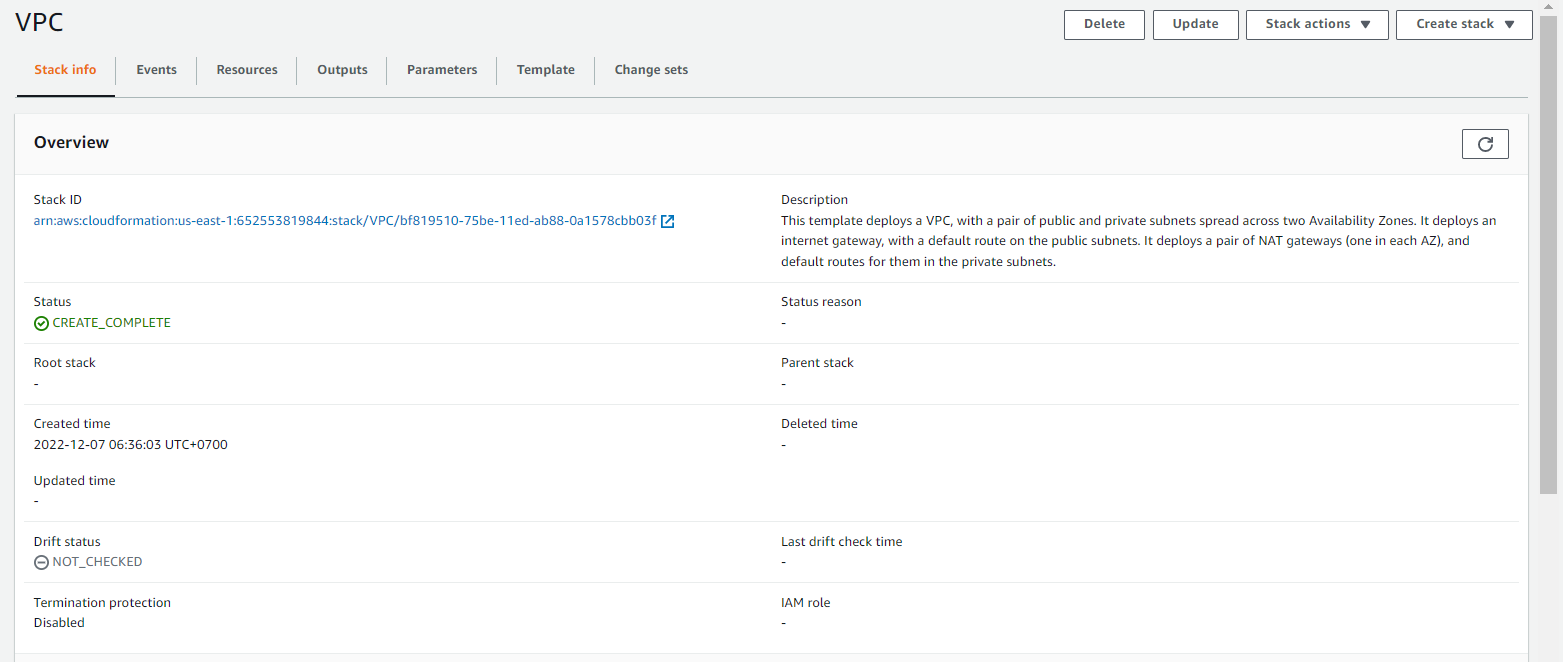


* + 1. Stack đã được tạo xong

Graphical user interface, text, application

Description automatically generated

* + 1. Kiểm tra lại Stack



Stack Info

Graphical user interface, application, Teams

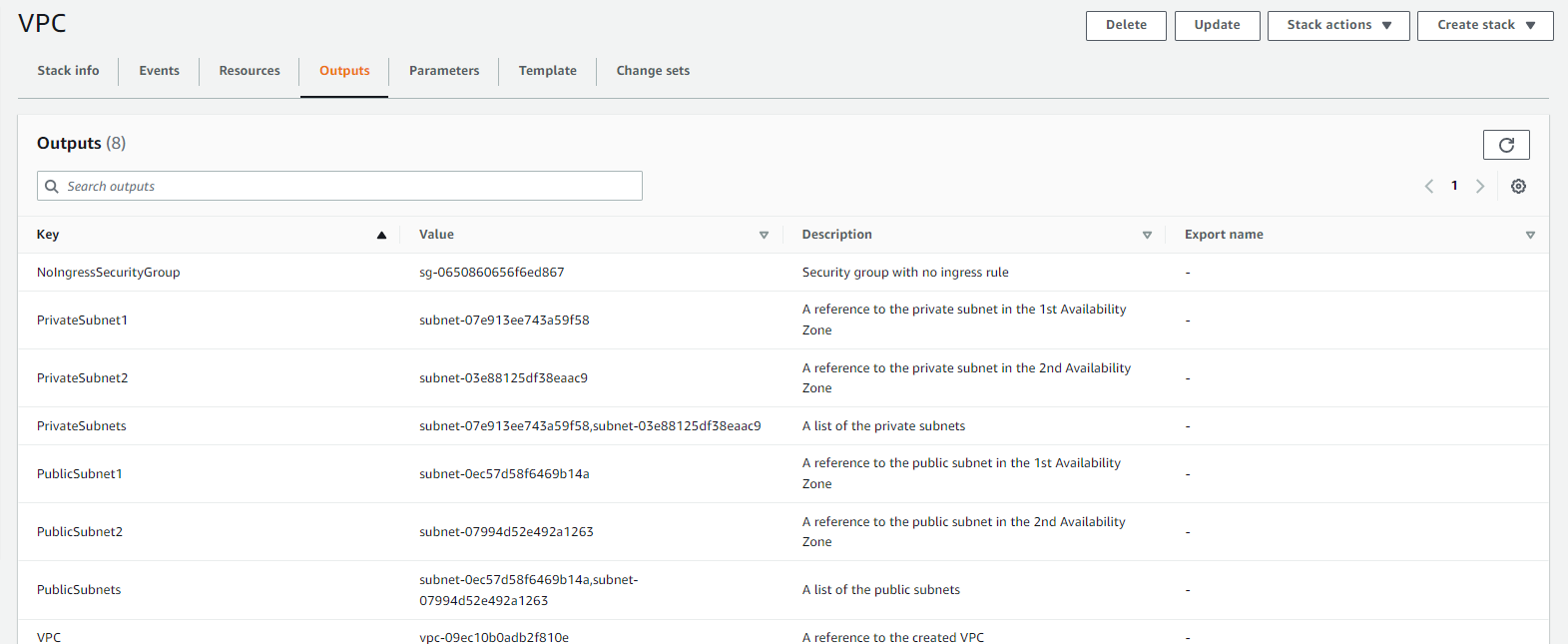
Description automatically generated

Parameters

Graphical user interface, application

Description automatically generated

Resources



Outputs

1. **DynamoDB**
2. **Simple Wordpress**

# PHẦN KẾT LUẬN

1. **Kết quả thực hiện và thực nghiệm được**

Sau thời gian tìm hiểu, nghiên cứu và cũng như thực hiện đề tài “AWS Cloud Formation và ứng dụng”, nhóm chúng em đã đạt được những kết quả như sau:  
- Nắm bắt được các kiến thức nền tảng về AWS Cloud Formation

* Hiểu được cách hoạt động của AWS Cloud Formation và có thể áp dụng kiến thức được học để tra cứu tài liệu để thiết kế và xây dựng template cho ba loại ứng dụng chính: Websites, Mobile apps, và Streaming application.

1. **Hướng phát triển**

* Hoàn thiện các templates
* Phát triển thêm các templates cho ứng dụng AI

# TÀI LIỆU THAM KHẢO

[1]