

- CloudFormation
  - 步驟紀錄
  - Configure stack options
  - 查詢
  - 使用 CLI 部署 CloudFormation Stack
  - 徹底刪除
  - 手動刪除 VPC
  - 手動刪除安全群組
  - 刪除子網
  - 刪除網際網路閘道
  - 檢查所有資源是否被刪除
  - 補充

# CloudFormation

---

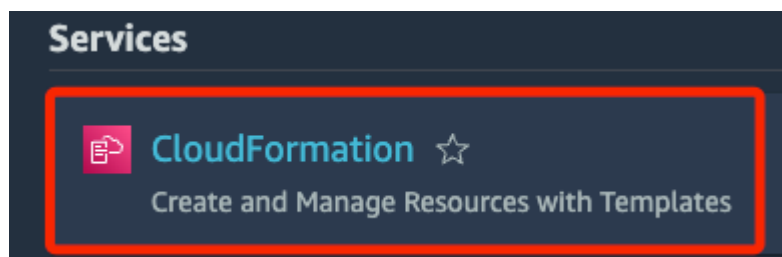
以下嘗試完成老師提到的使用 `.yaml` 腳本自動化建立雲計算 `EC2` 環境任務；特別說明，尚未找到合適可操作的 Lab，所以以下筆記使用 Root 帳號進行操作；另外，老師也提到應使用 IAM 使用者進行操作，並為特定的任務授權所需的最小權限，但這是另一個主題，這裏暫時無視。

## 步驟紀錄

---

使用 `CloudFormation` 完成自動化建立雲計算環境任務，可透過 `.yaml` 或 `.json` 格式的模板進行部署和管理 CloudFormation Stack，以下分別使用主控台 與 CLI 來進行，既然使用了模板，這裡就暫不使用 SDK 部署。

1. 搜尋並進入 `CloudFormation`。



## 2. 點擊 **Create stack**。

### Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

Create stack

3. 在本地建立模板文件 `*.yaml`，命名為 `cloudformation-template.yaml`，完整內容如下；這個模板會自動建立一個 VPC、子網路、網際網路閘道、路由表、安全群組，並在此網路架構中啟動一個 EC2 實例，後續進行操作時會上傳到 AWS 服務中。

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '使用 CloudFormation 一鍵式部署 EC2 實例的雲端環境'

Resources:
  MyVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: '10.0.0.0/16'
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: MyVPC

  MySubnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: '10.0.1.0/24'
      MapPublicIpOnLaunch: true
      AvailabilityZone: 'us-east-1a'
      Tags:
        - Key: Name
          Value: MySubnet

  MyInternetGateway:
    Type: 'AWS::EC2::InternetGateway'
    Properties:
      Tags:
        - Key: Name
          Value: MyInternetGateway
```

```

AttachGateway:
  Type: 'AWS::EC2::VPCGatewayAttachment'
  Properties:
    VpcId: !Ref MyVPC
    InternetGatewayId: !Ref MyInternetGateway

MyRouteTable:
  Type: 'AWS::EC2::RouteTable'
  Properties:
    VpcId: !Ref MyVPC
  Tags:
    - Key: Name
      Value: MyRouteTable

MyRoute:
  Type: 'AWS::EC2::Route'
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref MyRouteTable
    DestinationCidrBlock: '0.0.0.0/0'
    GatewayId: !Ref MyInternetGateway

MySubnetRouteTableAssociation:
  Type: 'AWS::EC2::SubnetRouteTableAssociation'
  Properties:
    SubnetId: !Ref MySubnet
    RouteTableId: !Ref MyRouteTable

MySecurityGroup:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    # 使用英文描述
    GroupDescription: 'Allow HTTP and SSH traffic'
    VpcId: !Ref MyVPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        CidrIp: '0.0.0.0/0'
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: '0.0.0.0/0'
  Tags:
    - Key: Name
      Value: MySecurityGroup

MyEC2Instance:
  Type: 'AWS::EC2::Instance'
  Properties:
    InstanceType: 't2.micro'
    # 替换 Key Pair 名稱
    KeyName: 'my-key-pair'
    # 使用最新的 Amazon Linux AMI ID
    ImageId: 'ami-098143f68772b34f5'
    NetworkInterfaces:

```

```
- AssociatePublicIpAddress: true
DeviceIndex: '0'
SubnetId: !Ref MySubnet
GroupSet:
  - !Ref MySecurityGroup
Tags:
  - Key: Name
    Value: MyEC2Instance
```

Outputs:

InstanceId:

Description: "EC2 Instance ID"

Value: !Ref MyEC2Instance

PublicIP:

Description: "Public IP Address of EC2 Instance"

Value: !GetAtt MyEC2Instance.PublicIp

4. 補充說明，EC2 安全群組的 **GroupDescription** 描述只支持 ASCII 字符，若使用中文描述將導致建立失敗；以下是測試過程中記錄下的錯誤的訊息。

⊗ CREATE\_FAILED

Resource handler returned message: "Value (?? HTTP ? SSH ??????) for parameter GroupDescription is invalid. Character sets beyond ASCII are not supported. (Service: Ec2, Status Code: 400, Request ID: 873622db-8789-48ea-ade9-a0ce87f2cf04)" (RequestToken: da00c17a-7f79-233f-01f6-31163e4ac181, HandlerErrorCode: InvalidRequest)

5. 點擊 **Choose an existing template**。

### Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ **Choose an existing template**  
Upload or choose an existing template.

☐ **Use a sample template**  
Choose from our sample template library.

☐ **Build from Application Composer**  
Create a template using a visual builder.

6. 選取 **Upload a template file**，然後點擊 **Choose file**。

## Specify template [Info](#)

A template is a JSON or YAML file that describes your stack's resources and properties.

### Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL

Provide an Amazon S3 URL to your template.

☒ Upload a template file

Upload your template directly to the console.

☐ Sync from Git - *new*

Sync a template from your Git repository.

### Upload a template file

 Choose file

JSON or YAML formatted file

7. 選擇前面編輯的腳本。

### Upload a template file

 Choose file

cloudformation-template.yaml

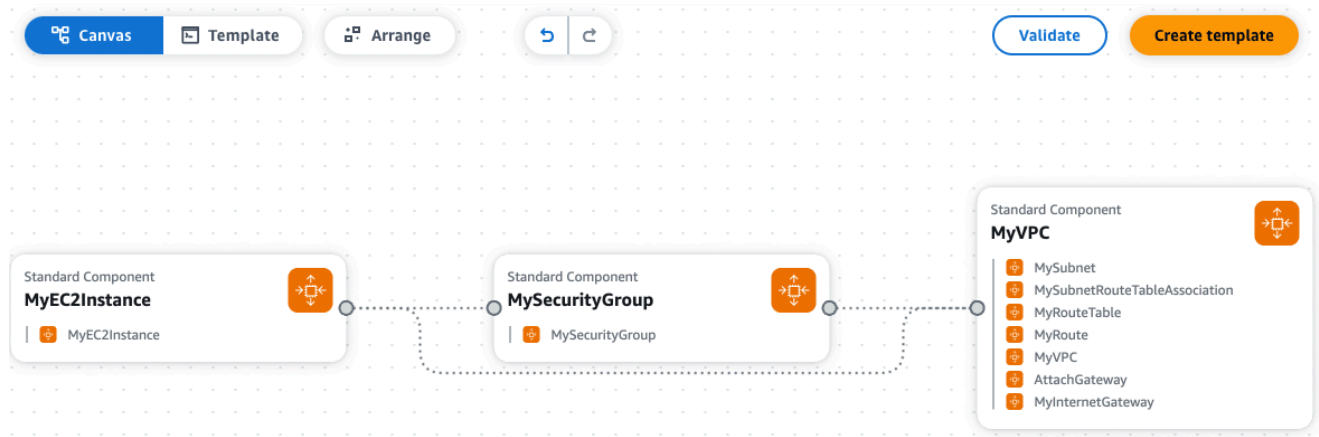
JSON or YAML formatted file

8. 可點擊 **View In Application Composer** 進行查看。

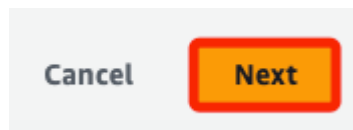
S3 URL: <https://s3.us-east-1.amazonaws.com/cf-templates-12brfq1cridbg-us-east-1/2024-09-12T162434.318Z9br-cloudformation-template.yaml>

**View in Application Composer**

9. 顯示如下圖；**MyEC2Instance** 與 **MySecurityGroup** 相關聯，表示 EC2 實例放置在定義的安全群組中；**MySecurityGroup** 與 **MyVPC** 相關聯，代表安全群組被應用在建立的 VPC 中，而 **MyVPC** 中包含了 **子網路**、**路由表**、**網際網路閘道** 等定義。



10. 點擊 **Next** 。



11. 命名為 **my-ec2-stack**，然後點擊 **Next** 。

## Configure stack options

1. 可點擊添加標籤 **Add new tag**；將 Key 設置為 **Project**、Value 設置為 **EC2-Setup**；編輯標籤有利於更好地組織與識別資源，特別在有多個 stack 的狀況。


**Tags - optional**  
Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organizing resources. You can add up to 50 unique tags for each stack.

Key	Value - Tags - optional
<input type="text" value="Project"/>	<input type="text" value="EC2-Setup"/>

2. 其他使用預設，然後點擊最下方 **Next**。

3. 檢查後點擊 **Submit** 建立。

4. 接下來在運行過程中會先顯示 **CREATE\_IN\_PROGRESS**。

Stacks	
my-ec2-stack	
2024-09-13 00:49:53 UTC+0800	
 <b>CREATE_IN_PROGRESS</b>	

5. 右側會顯示過程中事件 **Events** 日誌。

Stacks (1) Filter status Active View nested

Stacks

my-ec2-stack  
2024-09-13 01:33:09 UTC+0800  
CREATE\_IN\_PROGRESS

Events (22) Detect root cause

Search events

Timestamp	Logical ID	Status	Detailed description
2024-09-13 01:33:29 UTC+0800	MyEC2Instance	CREATE_IN_PROGRESS	-
2024-09-13 01:33:29 UTC+0800	MySubnetRouteTableAssociation	CREATE_IN_PROGRESS	-
2024-09-13 01:33:29 UTC+0800	MySubnet	CREATE_COMPLETE	-
2024-09-13 01:33:29 UTC+0800	MySecurityGroup	CREATE_COMPLETE	-

6. 完成時顯示 **CREATE\_COMPLETE**。

Stacks

my-ec2-stack  
2024-09-13 01:33:09 UTC+0800  
CREATE\_COMPLETE

## 查詢

1. 進入 EC2 可查看建立的實例。

Instances (1) Info Last updated 2 minutes ago Connect

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state
MyEC2Instance	i-09babc99bcefd873b	Running

2. 查看安全群組。



Security Groups (3) <a href="#">Info</a>				<a href="#">Refresh</a>	<a href="#">Actions</a> ▼	<a href="#">Export security groups to CSV</a> ▼	<a href="#">Create</a>
<input type="text" value="Find resources by attribute or tag"/>							
<input type="checkbox"/>	Name ▼	Security group ID ▼	Security group name ▼				
<input type="checkbox"/>	-	<a href="#">sg-010b64111bfd79227</a>	default				
<input type="checkbox"/>	MySecurityGroup	<a href="#">sg-02a270b26906f2338</a>	my-ec2-stack-MySecurityGroup-cO18...				
<input type="checkbox"/>	-	<a href="#">sg-019166c8904053811</a>	default				

3. 在 CloudFormation 的 Resources 頁籤中可查看更詳盡的資訊。

Stack info   Events   <b>Resources</b>   Outputs   Parameters   Template   Change sets   Git sync - new						
Resources (9) <a href="#">Refresh</a>						
<input type="text" value="Search resources"/>						
Logical ID ▲	Physical ID ▼	Type ▼	Status ▼	Module		
AttachGateway	IGW vpc-0f4e0a5c8a0d9457c	AWS::EC2::VPCGatewayAttachment	✔ CREATE_COMPLETE	-		
MyEC2Instance	<a href="#">i-09bab99bcefd873b</a>	AWS::EC2::Instance	✔ CREATE_COMPLETE	-		
MyInternetGateway	<a href="#">igw-0d7f5f9fe6e07cfdc</a>	AWS::EC2::InternetGateway	✔ CREATE_COMPLETE	-		
MyRoute	rtb-0cdf184d68b1330b1 0.0.0.0/0	AWS::EC2::Route	✔ CREATE_COMPLETE	-		
MyRouteTable	rtb-0cdf184d68b1330b1	AWS::EC2::RouteTable	✔ CREATE_COMPLETE	-		
MySecurityGroup	<a href="#">sg-02a270b26906f2338</a>	AWS::EC2::SecurityGroup	✔ CREATE_COMPLETE	-		
MySubnet	<a href="#">subnet-04f293d8528046c0a</a>	AWS::EC2::Subnet	✔ CREATE_COMPLETE	-		
MySubnetRouteTableAssociation	rtbassoc-0bd56b9b7182ed9b9	AWS::EC2::SubnetRouteTableAssociation	✔ CREATE_COMPLETE	-		
MyVPC	<a href="#">vpc-0f4e0a5c8a0d9457c</a>	AWS::EC2::VPC	✔ CREATE_COMPLETE	-		

## 使用 CLI 部署 CloudFormation Stack

確保已安裝並配置好 AWS CLI

1. 使用 `pwd` 指令確認腳本所在路徑。

```
(envAWS) samhsiao@SamdeMac-mini 00_template % pwd
/Volumes/SSD_01/00_課程講義專用/AWS_2024/01_課程筆記/20240905/02_0912/00_template
```

2. 開啟終端機運行以下指令，這會建立一個 CloudFormation Stack 並命名為 `my-ec3-stack`，並根據模板建立 EC2 實例及其相關的網路資源；使用 `file://` 路徑來指向模板。

```
aws cloudformation create-stack \
--stack-name my-ec3-stack \
--template-body file:///Volumes/SSD_01/00_課程講義專用/AWS_2024/01_課程筆記/20240905/02_0912/00_template/cloudformation-template.yaml \
--capabilities CAPABILITY_NAMED_IAM
```

3. 部署完成後，使用 AWS CLI 查詢 EC2 實例信息，驗證資源是否正確建立；這將返回 Stack 的詳細信息。

```
aws cloudformation describe-stacks --stack-name my-ec3-stack
```

4. 僅查詢公共 IP，記錄下來備用。

```
aws cloudformation describe-stacks \
--stack-name my-ec3-stack \
--query "Stacks[0].Outputs[?OutputKey=='PublicIP'].OutputValue" \
--output text
```

```
(envAWS) samhsiao@SamdeMac-mini 00_template
--stack-name my-ec3-stack \
--query "Stacks[0].Outputs[?OutputKey=='PublicIP'].OutputValue" \
--output text
52.91.88.126
```

5. 進入 \*.pem 憑證文件所在路徑，依據在之前筆記說明過的，官網指示 \*.pem 文件需透過 chmod 指令降低授權到 400，也就是 唯讀。

```
chmod 400 my-key-pair.pem
```

查詢、授權、確認

```
(envAWS) samhsiao@SamdeMac-mini Downloads % ls my-key-pair.pem
my-key-pair.pem
(envAWS) samhsiao@SamdeMac-mini Downloads % chmod 400 my-key-pair.pem
(envAWS) samhsiao@SamdeMac-mini Downloads % ls -l my-key-pair.pem
-r-----@ 1 samhsiao  staff  1674  9 13 00:12 my-key-pair.pem
```

6. 使用 SSH 讀取 Key Pair 文件連接到 EC2 實例的公共 IP，在這裡是 52.91.88.126。

```
ssh -i my-key-pair.pem ec2-user@<公共 IP>
```

```
(envAWS) samhsiao@SamdeMac-mini Downloads % ssh -i my-key-pair.pem ec2-user@52.91.88.126
_ _ | _ _ | _ )
_| ( / Amazon Linux 2 AMI
_ _ | \ _ _ | _ _ |

https://aws.amazon.com/amazon-linux-2/
35 package(s) needed for security, out of 58 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-60 ~]$
```

## 徹底刪除

包括 stack、VPC、安全群組；練習過後儘可能刪除一切使用的資源，避免殘留項目產生不必要的費用；在 Lab 中可透過 End Lab 進行。

1. 刪除 CloudFormation stack；另外，CloudFormation 會自動處理資源的刪除，所以當刪除 **stack** 時，會自動刪除該 **stack** 中建立的所有資源，包括 EC2 實例、VPC、子網、路由表、安全群組等。

```
aws cloudformation delete-stack --stack-name my-ec2-stack
```

2. 確認stack的刪除進度。

```
aws cloudformation describe-stacks --stack-name my-ec2-stack
```

## 手動刪除 VPC

如果 VPC 沒有被自動刪除，可手動刪除 VPC

1. 列出所有 VPC。

```
aws ec2 describe-vpcs --query "Vpcs[*].[VpcId,Tags]" --output table
```



DescribeVpcs	
vpc-0bc382c104053eb24	None

2. 確認是否為預設的 VPC；如果返回 **true** 則代表這是預設的 VPC，不建議刪除。

```
aws ec2 describe-vpcs --vpc-ids vpc-0bc382c104053eb24 --query "Vpcs[0].IsDefault"
```

3. 若確認是其他方式建立且無用的 VPC，可根據目標 ID 進行刪除。

```
aws ec2 delete-vpc --vpc-id <輸入目標識別 ID>
```

## 手動刪除安全群組

與前面相同，如果安全群組沒有被自動刪除，可手動進行刪除

1. 列出所有安全群組。

```
aws ec2 describe-security-groups --query "SecurityGroups[*].  
[GroupId,GroupName,Tags]" --output table
```



DescribeSecurityGroups		
sg-019166c8904053811	default	None

2. 可透過指令進行手動刪除安全群組。

```
aws ec2 delete-security-group --group-id <安全群組 ID>
```

## 刪除子網

手動刪除子網

1. 列出所有子網。

```
aws ec2 describe-subnets --query "Subnets[*].[SubnetId,Tags]" --output table
```

2. 刪除指定子網。

```
aws ec2 delete-subnet --subnet-id <指定子網 ID>
```

## 刪除網際網路閘道

1. 舉例來說，若要刪除以下的閘道 `igw-0b609a15f8ee56719`。

Internet gateways (1) <a href="#">Info</a>				
<input type="text" value="Search"/>				
<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID
<input type="checkbox"/>	-	<a href="#">igw-0b609a15f8ee56719</a>	✔ Attached	<a href="#">vpc-0bc382c104053eb24</a>

2. 先分離閘道與 VPC `vpc-0bc382c104053eb24` 的關聯。

```
aws ec2 detach-internet-gateway --internet-gateway-id `<如上閘道 ID>` --vpc-id `<如上 VPC ID>`
```

3. 刪除指定閘道。

```
aws ec2 delete-internet-gateway --internet-gateway-id `<指定閘道 ID>`
```

# 檢查所有資源是否被刪除

---

## 1. VPC。

```
aws ec2 describe-vpcs --query "Vpcs[*].[VpcId,Tags]" --output table
```

## 2. 安全群組。

```
aws ec2 describe-security-groups --query "SecurityGroups[*].  
[GroupId,GroupName,Tags]" --output table
```

## 3. 子網。

```
aws ec2 describe-subnets --query "Subnets[*].[SubnetId,Tags]" --output  
table
```

## 4. 閘道。

```
aws ec2 describe-internet-gateways --output table
```

## 5. 確認 CloudFormation stack 已完全刪除。

```
aws cloudformation describe-stacks --stack-name my-ec2-stack
```

```
(envAWS) samhsiao@SamdeMac-mini ~ % aws cloudformation describe-stacks --stack-name my-ec2-stack
```

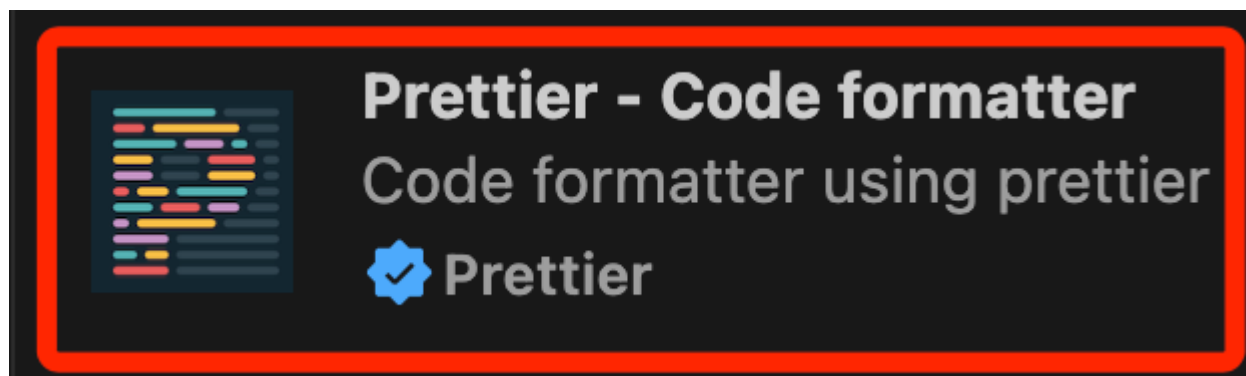
An error occurred (ValidationError) when calling the DescribeStacks operation: Stack with id my-ec2-stack does not exist

## 補充

---

關於 VSCode 中編輯 YAML 格式化的設定

1. 安裝插件 **Prettier**。



2. 在 **setting.json** 中進行設置即可；特別注意，不要使用 **redhat** 進行排版，實測不好用、不贅述。

```
"[yaml]": {  
  // 不要使用  
  //"editor.defaultFormatter": "redhat.vscode-yaml"  
  "editor.defaultFormatter": "esbenp.prettier-vscode"  
}
```

---

END