*Kevin Chen*

2021-06-15

Amazon EKS部署CICD使用Aws CodeCommit, CodePipeline, CodeBuild以及Flux

操作手冊

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 |  |  |  |
|  |  |  |  |

# 文件目的

採用AWS列管服務方案（Managed services），本文件目的說明各項相關操作步驟，如何藉由使用持續性整合（Continuous Integration, CI）傳輸管道（pipeline）快速發展Kubernetes（K8s）基礎建設，以採用FluxCD與GitOps原理做為持續性發佈（Continous Delivery, CD）元件建立Docker image與自動化部署Amazon Elastic Kubernetes（Amazon EKS）服務叢集（cluster）

相關解說，詳閱：<https://aws.amazon.com/blogs/devops/ci-cd-on-amazon-eks-using-aws-codecommit-aws-codepipeline-aws-codebuild-and-fluxcd/>

## 文件對象

假設閱讀者已經：

* 持有Amazon AWS使用者帳號且被授與administrator群組權限
* 瞭解Amazon ECS原理與架構
* 瞭解Amazon EKS原理與架構
* 瞭解Amazon ECR原理與架構
* 採用Ubuntu系統作為工作站
* 瞭解Amazon CodeCommit或類似GitHub原理與操作
* 瞭解unix-like命令列操作，如ubuntu

## 採用AWS CDK＋FluxCD的利弊（Pros and Cons）

使用AWS CDK好處在於設定完成後，下cdk deploy就能自動安裝所需的AWS EKS雲端執行環境，只剩下執行本範例或客戶的程式碼專案透過CodeCommit或GitHub機制觸發CD自動部署到AWS EKS。

(pros&cons)

### AWS CDK

### 1.2.2 FluxCD

# 環境需求

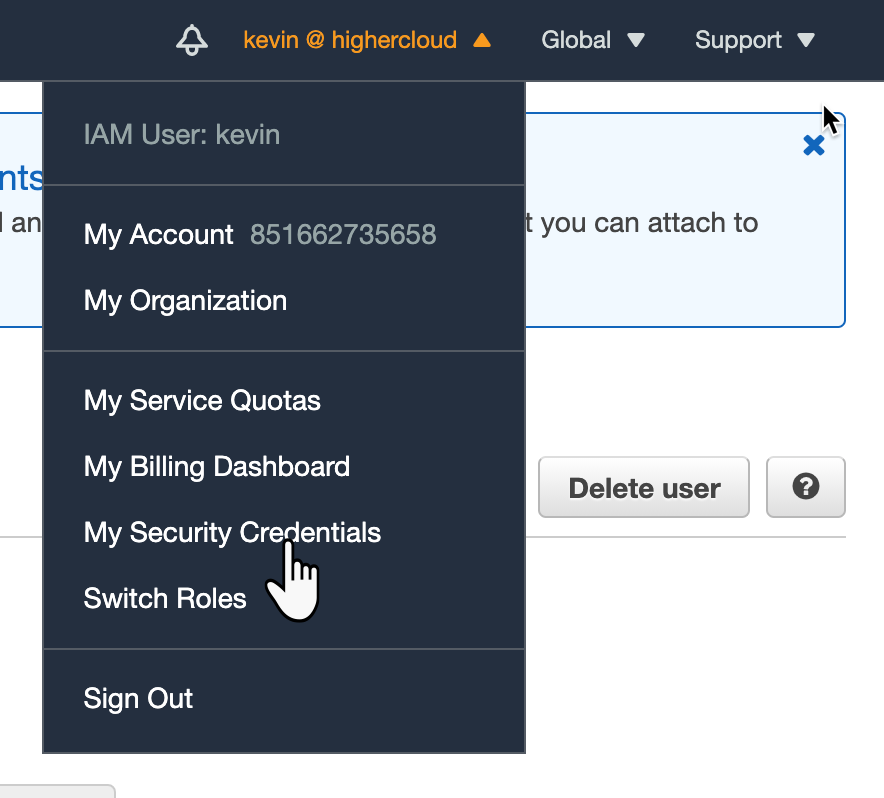
## 2.1 AWS帳號

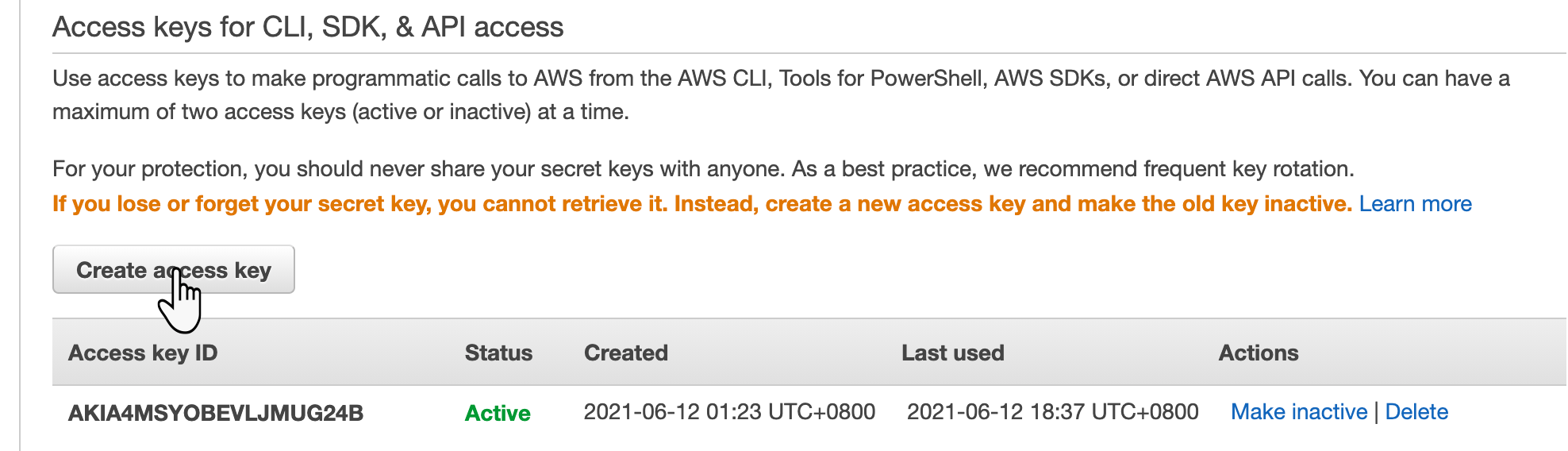
(TODO)

## 2.2 安裝AWS CLI

### 2.2.1 安裝最新版AWS CLI第2版。安裝說明，請查閱AWS使用者指南《[安裝、更新和解除安裝AWS CLI](https://docs.aws.amazon.com/zh_tw/cli/latest/userguide/cli-chap-install.html)》

### 設定AWS 帳號資訊於本地AWS CLI使用

* 步驟一：[到登入帳號的下拉選單點選下圖位置](https://console.aws.amazon.com/iam/home" \l "/security_credentials)。
* 步驟二：建立Access Keys for CLI, SDK, & API access，點選”Create access key”，隨後會有檔案可以下載：使用者\_accessKeys.csv，內有Access key ID、Secret access key。



* 步驟三：在AWS CLI設定使用者帳號安全憑證

$aws config

#下列依命令回應問題填入設定

#資料來自於2.2.1設定的使用者\_accessKeys.csv檔案內

AWS Access Key ID [None]: *Type your target AWS access key ID here, and then press Enter*

AWS Secret Access Key [None]: *Type your target AWS secret access key here, and then press Enter*

Default region name [None]: *Type a supported region for CodeCommit here, and then press Enter*

Default output format [None]: *Type* json *here, and then press Enter*

## 2.2 安裝AWS CDK

AWS 雲開發工具包（Cloud Development Kit, CDK）以五種程式語言（TypeScript、Javascript、Python、Java或C#）所撰寫的應用程式。我們將以Amazon提供的Python範例（[aws-samples/aws-cdk-eks-flux](https://github.com/aws-samples/aws-cdk-eks-flux)）輕鬆部署eks與FluxCD 完成CICD機制。

### 2.2.1必要條件（工作站端）

* Python 3.6或以上版本，用來執行本範例。
* Node.JS 10.13.0或以上版本：用來安裝AWS CDK。
* 設置你的ubuntu, Cloud9 或Windows具有AWS Credential憑證：安裝參考附錄。

### 2.2.2開始安裝AWS CDK

* **步驟一：檢查是否安裝Python 3.6版以上；若沒有，請參考附錄**

$python3

#或者

$python

* **步驟二：檢查是否安裝Node.JS 10.13.0或以上版本；若沒有，請參考附錄**

$node -v

* **步驟三：開始安裝CDK**

$ npm install -g aws-cdk

#安裝後檢查版本，也檢查cdk運作是否正常

$cdk --version

* **步驟四：（選擇性）**

$cd aws-cdk-eks-flux

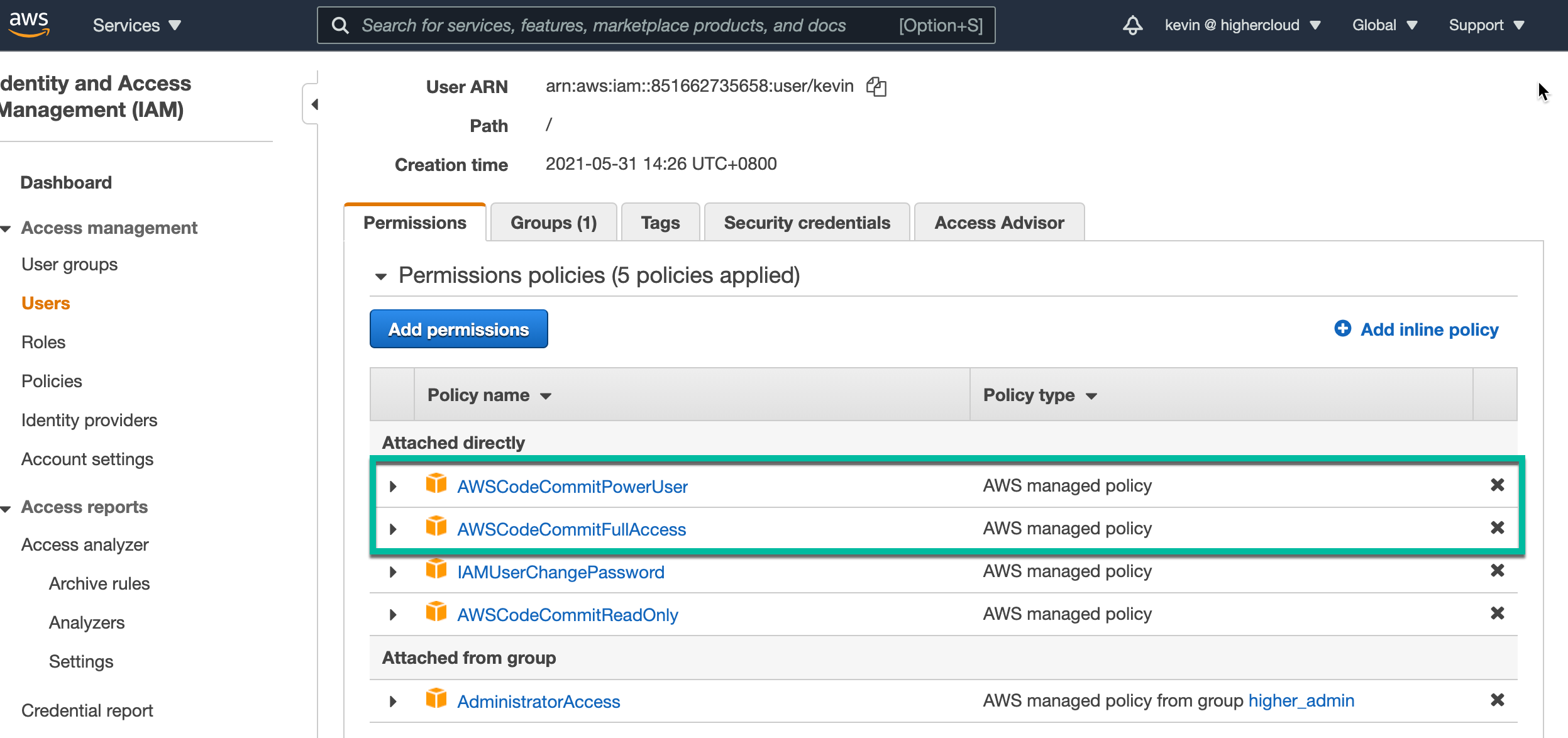
#如果是新建CDK app則執行以下命令，在該目錄初始化CDK

$cdk init app --language python

## 2.3 啟用AWS CodeCommit

若您已知道目前使用者帳號權限可使用AWS CodeCommit所在區域（region）服務，可以跳過2.3.1，該權限允許CDK執行時會在CodeCommit自動建立所需要Repository的權限。

本例而言，至少需要下圖綠色框線內的AWS權限。



### 2.3.1 使用 git 登入 CodeCommit

### 由於在本機ubuntu使用git 命令操作CodeCommit，需要存取repo權限，允許本地修改檔案push到CodeCommit，請參考AWS Commit使用者指南>設定>[對於使用 Git 認證的 HTTPS 使用者設定](https://docs.aws.amazon.com/zh_tw/codecommit/latest/userguide/setting-up-gc.html)，講解如何在ubuntu安裝git、產生CodeCommit的username與password，所產生的憑證亦提供下載檔案（使用者\_codecommit\_credentials.csv）。

### 2.3.2 設定SSH public key存取AWS CodeCommit repo

在本地ubuntu 命令列視窗，用下列命令，輸入檔名codecommit，其他可忽略，可產生ssh金鑰分別為私鑰（private key）與公鑰（public key）：

* 私鑰：/home/your-user-name/.ssh/codecommit\_rsa
* 公鑰：/home/your-user-name/.ssh/codecommit\_rsa.pub

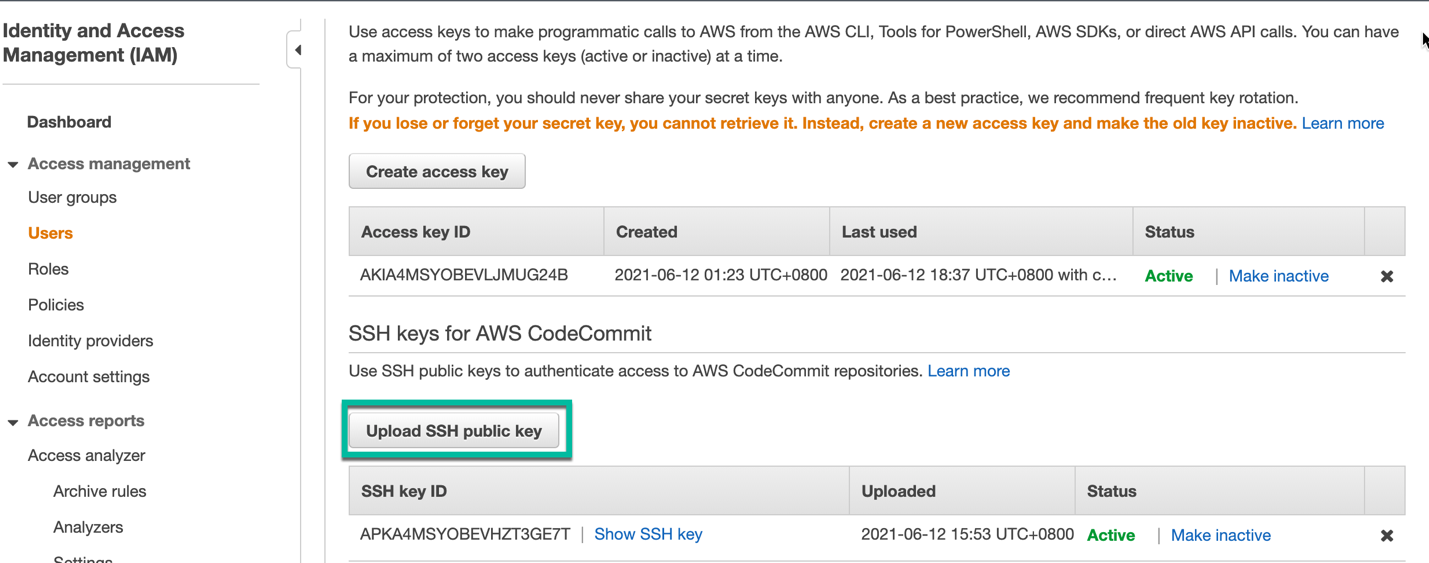
$ssh-keygen

(…)

#查看公鑰

$cat ~/.ssh/codecommit\_rsa.pub

#### 2.3.2.1 將產生的public key上傳到IAM>Users>username>Security credentials>SSH keys for AWS CodeCommit>Upload SSH public key



在本機電腦上，使用文字編輯器在 ~/.ssh 目錄中建立組態檔，然後將下列幾行新增至檔案，其中 *User* 是您之前複製的 SSH 密鑰 ID：

$nano ~/.ssh/config

在2.2.2步驟三所產生的access key ID(使用者名稱＿accessKeys.csv)填入User。

將2.3.2的public key路徑填入IdentityFile。

Host git-codecommit.\*.amazonaws.com

User APKAEIBAERJR2EXAMPLE

IdentityFile ~/.ssh/codecommit\_rsa

授與擁有者（此例為ubuntu）讀寫config權限

$chomd 600 config

2.3.2.2 在執行git命令時，避免詢問username與password作法

詳請參考[使用 AWS CLI 認證協助程式，在 Linux、macOS 或 Unix 上設定 HTTPS 連線到 AWS CodeCommit 存放庫的步驟](https://docs.aws.amazon.com/zh_tw/codecommit/latest/userguide/setting-up-https-unixes.html)>步驟 3：設定認證協助程式

執行下列命令，將產生/home/ubuntu/.gitconfig檔案

$git config --global credential.helper '!aws codecommit credential-helper $@'

$git config --global credential.UseHttpPath true

並產生下列設定於.gitconfig檔案。

[credential]

helper = !aws --profile CodeCommitProfile codecommit credential-helper $@

UseHttpPath = true

編輯/home/ubuntu/.gitconfig

$nano /home/ubuntu/.gitconfig

複製下列設定，貼在.gitconfig內，並儲存檔案。該設定告訴git使用credential helper於特定地區CodeCommit。當使用git命令時不會詢問輸入username與password。

以本例而言，CodeCommit repo建在us-east-1區域則將紅色設定貼入.gitconfig檔案內。

[credential]

helper = !aws codecommit credential-helper $@

UseHttpPath = true

[credential]

helper = !aws --region us-east-1 codecommit credential-helper $@

UseHttpPath = true

[credential "https://git-codecommit.us-east-2.amazonaws.com"]

helper = !aws codecommit credential-helper $@

UseHttpPath = true

**[credential "https://git-codecommit.us-east-1.amazonaws.com"]**

**helper = !aws codecommit credential-helper $@**

**UseHttpPath = true**

[credential "https://git-codecommit.ap-east-1.amazonaws.com"]

helper = !aws codecommit credential-helper $@

UseHttpPath = true

## 2.4 啟用 AWS EKS

只要確認使用者帳號可以使用AWS EKS所在區域（region）服務，CDK執行時會自動建立所需要的EKS以及相關服務。

### 2.4.1 Bootstrapping

將CDK所需模組載入啟動點，並執行：

$cd aws-cdk-eks-flux

$cdk bootstrap aws://<你的aws\_account\_id>/<region>

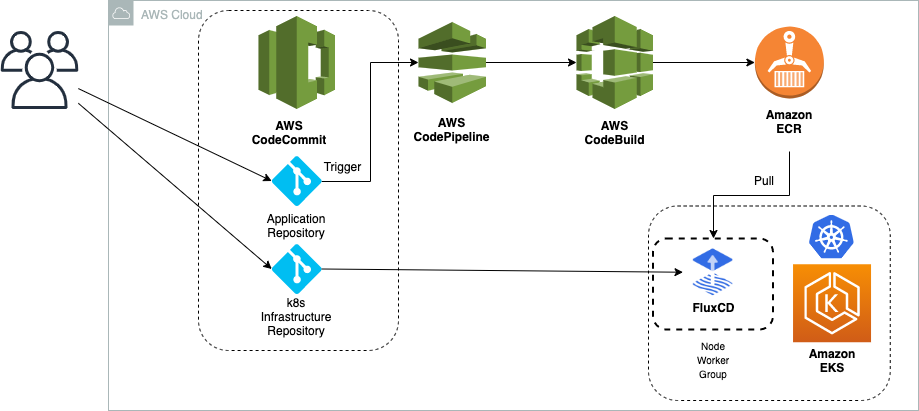
## 2.5 安裝FluxCD

## 2.6 額外學習

### 2.6.1 GitHub

如果不想使用CodeCommit，改用GitHub可操作以下練習。

# 架構說明



# 重現步驟

* + Option A：（本例非必要）將AWS提供的本範例程式Fork一份到你個人github帳號

＃在你的ubuntu 使用者home目錄，執行下列命令，將範例複製到ubuntu本地

$git clone [https://github.com/<](https://github.com/%3c)改成你的github帳號>/aws-cdk-eks-flux.git

* + Option B：（本例必要）直接將本範例程式下載到ubuntu

＃在你的ubuntu 使用者home目錄，執行下列命令，將範例複製到ubuntu本地

## $git clone https://github.com/aws-samples/aws-cdk-eks-flux.git

## 4.1 dd

## 4.1 在cdk.json指定技術堆疊名稱，與AWS EKS服務區域

路徑aws-cdk-eks-flux/cdk.json。第一次執行本範例，region可改成使用者帳號可使用eks的區域。(下列紅色字可以更換)

{

"app": "python3 app.py",

"context": {

"@aws-cdk/core:enableStackNameDuplicates": "true",

"aws-cdk:enableDiffNoFail": "true",

"@aws-cdk/core:stackRelativeExports": "true",

"@aws-cdk/aws-ecr-assets:dockerIgnoreSupport": true,

"name": "**awsome-flux-example**",

"region": "**us-east-1**"

}

}

2.2.4 CDK所需的python模組

由於執行cdk deploy需要10～20分鐘以上時間，按照範例說明執行會跳出缺少aws某些python模組的錯誤，以下說明幾種解決方法：

* **方法一**
* **方法二**
* **方法三**

# 確認叢集

# 清除所建

# 客製化

# 文件授權

這個程式庫是在 MIT-0 許可下許可的。 請參閱許可證文件。

本文件由海爾雲端（股）公司技術處參考Amazon範例（[aws-samples/aws-cdk-eks-flux](https://github.com/aws-samples/aws-cdk-eks-flux)）補充原著說明以完整重現步驟，相關著作權，智慧財產權屬於海爾雲端（股）公司技術處所有。

# 附錄

## 9.1 安裝Python

9.1.1 在ubuntu安裝

$sudo apt install software-properties-common

$sudo add-apt-repository ppa:deadsnakes/ppa

$sudo apt update

$sudo apt install python3.8

9.1.2 安裝pip模組

$python -m ensurepip --upgrade

$python -m pip install --upgrade pip

$python -m pip install --upgrade virtualenv

9.1.3 讓本範例在Python虛擬環境執行

$cd **aws-cdk-eks-flux**

source .venv/bin/activate

執行本範例，可以不用執行

9.1.4 安裝pip模組

## 9.2 安裝Node.JS

* 在ubuntu安裝可參考：<https://andy6804tw.github.io/2019/09/23/ubuntu-indtall-nodejs/>

## 9.3 產生IAM Git憑證

## 9.4 操作AWS CLI手動安裝AWS EKS

## 9.5安裝開放程式FluxCD

## 9.6 安裝AWS-CDK Python模組

按照範例執行cdk deploy會發生缺少aws-cdk模組錯誤，可以依照錯誤訊息所缺的模組，尋找下列模組清單，並執行 pip install <aws-cdk模組名稱>

aws.cdk.core

aws-cdk.alexa-ask

aws-cdk.app-delivery

aws-cdk.assets

aws-cdk.aws-amazonmq

aws-cdk.aws-amplify

aws-cdk.aws-apigateway

aws-cdk.aws-applicationautoscaling

aws-cdk.aws-appmesh

aws-cdk.aws-appstream

aws-cdk.aws-appsync

aws-cdk.aws-athena

aws-cdk.aws-autoscaling

aws-cdk.aws-autoscaling-common

aws-cdk.aws-autoscaling-hooktargets

aws-cdk.aws-autoscalingplans

aws-cdk.aws-backup

aws-cdk.aws-batch

aws-cdk.aws-budgets

aws-cdk.aws-certificatemanager

aws-cdk.aws-cloud9

aws-cdk.aws-cloudformation

aws-cdk.aws-cloudfront

aws-cdk.aws-cloudtrail

aws-cdk.aws-cloudwatch

aws-cdk.aws-cloudwatch-actions

aws-cdk.aws-codebuild

aws-cdk.aws-codecommit

aws-cdk.aws-codedeploy

aws-cdk.aws-codepipeline

aws-cdk.aws-codepipeline-actions

aws-cdk.aws-codestar

aws-cdk.aws-cognito

aws-cdk.aws-config

aws-cdk.aws-datapipeline

aws-cdk.aws-dax

aws-cdk.aws-directoryservice

aws-cdk.aws-dlm

aws-cdk.aws-dms

aws-cdk.aws-docdb

aws-cdk.aws-dynamodb

aws-cdk.aws-dynamodb-global

aws-cdk.aws-ec2

aws-cdk.aws-ecr

aws-cdk.aws-ecr-assets

aws-cdk.aws-ecs

aws-cdk.aws-ecs-patterns

aws-cdk.aws-efs

aws-cdk.aws-eks

aws-cdk.aws-elasticache

aws-cdk.aws-elasticbeanstalk

aws-cdk.aws-elasticloadbalancing

aws-cdk.aws-elasticloadbalancingv2

aws-cdk.aws-elasticloadbalancingv2-targets

aws-cdk.aws-elasticsearch

aws-cdk.aws-emr

aws-cdk.aws-events

aws-cdk.aws-events-targets

aws-cdk.aws-fsx

aws-cdk.aws-gamelift

aws-cdk.aws-glue

aws-cdk.aws-greengrass

aws-cdk.aws-guardduty

aws-cdk.aws-iam

aws-cdk.aws-inspector

aws-cdk.aws-iot

aws-cdk.aws-iot1click

aws-cdk.aws-iotanalytics

aws-cdk.aws-iotevents

aws-cdk.aws-iotthingsgraph

aws-cdk.aws-kinesis

aws-cdk.aws-kinesisanalytics

aws-cdk.aws-kinesisfirehose

aws-cdk.aws-kms

aws-cdk.aws-lakeformation

aws-cdk.aws-lambda

aws-cdk.aws-lambda-event-sources

aws-cdk.aws-logs

aws-cdk.aws-logs-destinations

aws-cdk.aws-managedblockchain

aws-cdk.aws-medialive

aws-cdk.aws-mediastore

aws-cdk.aws-msk

aws-cdk.aws-neptune

aws-cdk.aws-opsworks

aws-cdk.aws-opsworkscm

aws-cdk.aws-pinpoint

aws-cdk.aws-pinpointemail

aws-cdk.aws-ram

aws-cdk.aws-rds

aws-cdk.aws-redshift

aws-cdk.aws-robomaker

aws-cdk.aws-route53

aws-cdk.aws-route53-targets

aws-cdk.aws-route53resolver

aws-cdk.aws-s3

aws-cdk.aws-s3-assets

aws-cdk.aws-s3-deployment

aws-cdk.aws-s3-notifications

aws-cdk.aws-sagemaker

aws-cdk.aws-sam

aws-cdk.aws-sdb

aws-cdk.aws-secretsmanager

aws-cdk.aws-securityhub

aws-cdk.aws-servicecatalog

aws-cdk.aws-servicediscovery

aws-cdk.aws-ses

aws-cdk.aws-sns

aws-cdk.aws-sns-subscriptions

aws-cdk.aws-sqs

aws-cdk.aws-ssm

aws-cdk.aws-stepfunctions

aws-cdk.aws-stepfunctions-tasks

aws-cdk.aws-transfer

aws-cdk.aws-waf

aws-cdk.aws-wafregional

aws-cdk.aws-workspaces

aws-cdk.core

aws-cdk.custom-resources

aws-cdk.cx-api

aws-cdk.region-info

## 9.7 AWS EKS手動安裝