AWS Hack'n'Roll Workshop

AWS User Group Taiwan

費用提醒

- AWS 大多服務都會收費
- 不用的 Stack 記得關閉 (cdk destroy)
    - 特別是 Docker Container (Service) 與 NAT Gateway

HelloWorldStack

```
git switch hello-world
```

CDK

- Concepts
- App Lifecycle
- Environment & Context
- Assets

Concepts

- App
  專案只能有一個，可包含多個 Stack

- Stack
  對應到 CFN Stack

- Construct
  包裝 Resource 或其他 Construct 的容器，
  用於整理並重用 Resources

- Resource
  包含直接對應 CFN 的低階 (CfnXXX) 與高階資源

Concepts

- CDK 如何知道 App 在哪？
- CDK 如何關聯 Resource, Construct, Stack 與 App

Concept

- CDK 如何關聯 Resource, Construct, Stack 與 App

```
new Stack(
    scope?: Construct,
    name?: string,
    props?: StackProps)
new Construct(
    scope: Construct,
    id: string)
new Vpc(
    scope: Construct,
    id: string,
    props?: VpcProps)
```

App LifeCycle

- Construction
- `cdk synth`
    - Prepare
      遞迴調用 construct.prepare()，執行 Aspect 等 hook
    - Validate
      遞迴調用 construct.validate()，回傳所有錯誤
    - Synthesize
      遞迴調用 construct.synthesize() 生成 (CDN) Artifact
- Deploy

https://docs.aws.amazon.com/cdk/latest/guide/apps.html

Construction

- `Vpc.fromLookup(scope, id, options: VpcLookupOptions)`
-
```
interface VpcLookupOptions {
    isDefault?: boolean
    tags?: Map<string, string>
    vpcId?: string
    vpcName?: string
}
```
- 上哪查呀？

Environment

- new Stack(scope?: Construct, name?: string,
  props?: StackProps)

- ```
  props = {
      env: {
          account: process.env.CDK_DEFAULT_ACCOUNT,
          region: process.env.CDK_DEFAULT_REGION
      }
  }
  ```

- 在 Synthesis 階段提供 stack.account, .region, .availabilityZones
  否則走 Token 機制，轉換爲 !Ref AWS::Region

https://docs.aws.amazon.com/cdk/latest/guide/environments.html
  ```

Context

- cdk.context.json

- 儲存透過 Account, Region 取得的資訊

EchoStack

```
git switch echo
```

echoStack

- echoStack 引入 echoConstruct
- 自 echoConstruct 定義 API Gateway 與 Lambda
- 從 Asset 建立 Lambda Function

Assets

- 打包 (Synthesis) 爲 Artifact
- Lambda 爲本地打包上傳，必要時需利用 Docker 部署
- 建議明確引入 AWS SDK，以免 Lambda 內的版本過舊
- Docker Asset 會自動使用 docker 建置

RainFallCrawlerStack

git switch rain-fall-crawler

RainFallCrawlerStack

- 定義 Lambda Function 與 DynamoDB Table
- 隱含 Execution Role

RainFallCrawlerConstruct

- 程式碼：lib/constructs/rainFallCrawlerConstruct.ts

- 任務一
  修正程式碼裡的問題，讓程式依照預期運行

- 任務二
  設置程式碼內的資源使 Lambda Function 每日執行一次

- （開放）任務三
  思考使用場景，可以如何調整資源配置

HelloDocker

```
git switch hello-docker
```

HelloDocker

- 與 HelloWorld 相同，使用 docker 建佈
- 自動創建 ECR Repository 儲存 docker image
- Docker 讓我們輕鬆的在本地開發、執行、驗證

ecsPatterns

- AWS CDK 將常用的模式包裝爲 Construct
- 引入 Construct 便能快速創建該模式

```
ecsPatterns
    .ApplicationLoadBalancedFargateService
```

Amazon ECS Fargate

- Docker Container Scheduler
- Cluster
  VPC {Subnet, Gateway, Route Table, Network ACL}
  Subnet, Security Groups
- Network
- Service
- Task
- Task Definition
  - Container

Amazon ECS

- 支持多種 Log drivers
- awslogs 輸出至 CloudWatch Logs
  - > Insight
  - > Filter > Alarm > Notification
    (Application) AutoScaling

docker pull from ECR

```
$(aws ecr get-login --no-include-email)
```

Delete 好像有點久？

Why?

RainFall

```
git switch rain-fall
```

RainFall

- 設計並實作資料查詢界面
- 資料源：RainFallCrawler
- 使用語言、服務不限

RainFall

- 任務一
  維持 RainFallCrawlerStack，但透過 bin/app.ts 將
  其作爲參數帶入 RainFallStack，透過 cdk 交換資訊

- 任務二
  在 RainFallStack 內引入 RainFallCrawlerConstruct，
  再開發 Lambda Function 或 Fargate Service 查詢

WeatherMonitor

```
git switch weather-monitor
```

WeatherMonitor

- 氣象資料開放平臺–現在天氣觀測報告 (O–A0003–001)
  https://opendata.cwb.gov.tw/dataset/observation/O-A0003-001

- 多維度時序性資料

  - Amazon Timestream

Amazon Timestream

快速、可擴展、全受管的時間序列資料庫

申請預覽版

Amazon Timestream 是一個適用於 IoT 和營運應用程式的快速、可擴展、全受管的時間序列資料庫服務，可輕鬆存放和分析每天數兆個事件，其成本僅為關聯式資料庫的十分之一。在 IoT 裝置、IT 系統和智慧型工業機器的推動下，時間序列資料 – 衡量事物如何隨時間變化的資料 – 是增長最快的資料類型之一。時間序列資料具有特定的特徵，例如通常依時間排序送達、是僅能附加的資料，而且總是查詢一段時間的資料。雖然關聯式資料庫可以存放這些資料，但由於並未針對存放和擷取不同時間間隔的資料進行優化，因此處理這類資料的效率不佳。 Timestream 是一個專門建構的時間序列資料庫，可依時間間隔有效地存放和處理這類資料。使用 Timestream 時，您可以輕鬆存放和分析 DevOps 日誌資料、IoT 應用程式感應器資料，以及設備維護的工業遙測資料。隨著時間資料越來越多，Timestream 調整式查詢處理引擎可知道資料的位置和格式，讓您可以更輕鬆快速地分析資

AWS re:Invent 2018 中的 Amazon
公告 (2:05)

WeatherMonitor

- 氣象資料開放平臺-現在天氣觀測報告 (O-A0003-001)
  https://opendata.cwb.gov.tw/dataset/observation/O-A0003-001

- 多維度時序性資料
  - Amazon Timestream
  - CloudWatch Custom Metrics

CloudWatch

- Logs, Filter, Events
- Metrics 由 Metric-Data 組成
  https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/API_PutMetricData.html
- namespace, name, dimensions
- data, unit

WeatherMonitor

- 任務一
  撰寫程式將氣象觀測值寫至 CloudWatch，
  並設定程式排程執行

- 任務二
  建立 CloudWatch Alarm 在特定值超出標準時
  觸發 SNS Topic 並發送信件通知

WeatherMonitor

```
git switch weather-monitor
```

WeatherMonitor

- 文字資料？

WeatherMonitor

- 文字資料
  - 比照 Enum 或 OneHot，輸出至 CloudWatch Metrics
  - 儲存至 DynamoDB Table
    - 使用 Data Stream 進行分析
    - 利用 Query 取出最新值
- 輸出至 Kinesis，由其他程式分析

WeatherMonitor

- 任務一
  比照 Enum 或 OneHot，輸出至 CloudWatch Metrics

- 任務二
  設計 DynamoDB Table Schema 使後端可快速取得一區域的歷史值，以及所有區域的最新值 (Hint: Secondary Index)

- 任務三
  輸出至 Kinesis Data Firehose，利用 S3 Event 觸發 Lambda 分析，視結果觸發 SNS Topic