雲端基礎設施視覺化分析平台
圖形資料庫最終報告

課程：高等資料庫系統
姓名：梁祐嘉
學號：01157145
班級：資工 4B

2025/10/22

摘要

本報告提出一套以 Neo4j 為核心的雲端基礎設施知識圖譜平台，旨在以圖形資料模型整合與分析專案在 AWS 等雲端環境中的複雜資源、關聯與依賴，並提供資安漏洞分析、故障衝擊分析與成本優化核心情境之查詢與分析。系統採用 Python 腳本透過雲端 API 擷取設定資料，轉換為節點與關係後匯入 Neo4j，以圖為中心進行視覺化與查詢。本專案成功實現三大功能完整檢測，系統無警告運行，提供詳細的分析報告和具體的問題檢測建議。

目錄

1 簡介

現代雲端環境涵蓋數百至數千個互相連結的資源（virtual hosts, databases, firewalls, load balancers, etc），其依賴、權限與網路關係錯綜複雜。傳統清單式或單點儀表板難以完整呈現多跳關聯，導致在資安風險評估、故障衝擊分析與成本優化等任務上面臨挑戰。本專案以雲端基礎設施知識圖譜為核心，透過 Neo4j 的圖形處理能力，將基礎設施模型化並提供直觀、可深度查詢的分析介面。

2 Neo4j 產品與服務

2.1 使用的 Neo4j 產品

本專案使用 **Neo4j Community Edition** 作為圖形資料庫核心，具體包括：

- **Neo4j Database**: 開源圖形資料庫引擎
- **Cypher Query Language**: 圖形查詢語言
- **Neo4j Python Driver**: Python 連接驅動程式
- **Neo4j Browser**: 圖形資料庫管理介面

2.2 技術架構

整體資料流程如下：

1. **資料擷取**：以 Python 透過雲端 API/CLI（如 AWS Boto3）定期匯出雲端帳號內資源設定(JSON)。
2. **資料轉換與載入**：解析 JSON，將資源轉為節點（Node）與關係（Relationship），並以 Neo4j Driver 匯入。
3. **查詢與分析**：於 Neo4j Browser 以 Cypher 進行分析與探索。

3 原始資料格式與來源

3.1 資料來源

1. **AWS API**: 透過 Boto3 SDK 擷取真實 AWS 資源
2. **Mock 資料**: 模擬 AWS 環境的測試資料
3. **增強模擬資料**: 包含完整測試場景的模擬資料

3.2 原始資料格式

原始資料採用 JSON 格式，包含以下主要結構：

```
{
  "ec2_instances": {
    "Instances": [
      {
        "InstanceId": "i-1234567890abcdef0",
        "InstanceType": "t3.micro",
        "State": {"Name": "running"},
        "PublicIpAddress": "54.123.45.67",
        "SecurityGroups": [...],
        "Tags": [...]
      }
    ]
  },
  "security_groups": {
    "SecurityGroups": [...]
  },
  "vpcs": {
    "Vpcs": [...]
  },
  "subnets": {
    "Subnets": [...]
  },
  "ebs_volumes": {
    "Volumes": [...]
  },
  "s3_buckets": {
    "Buckets": [...]
  }
}
```

4 圖形資料模型設計

4.1 核心節點（Nodes）

- `:EC2Instance`：屬性包含 `InstanceID`, `Name`, `State`, `PublicIP`, `InstanceType`。
- `:SecurityGroup`：屬性包含 `GroupID`, `GroupName`, `Description`, `VpcId`。
- `:SecurityRule`：屬性包含 `RuleID`, `Protocol`, `PortRange`, `SourceCIDR`, `Direction`。
- `:VPC`：屬性包含 `VpcId`, `Name`, `CidrBlock`, `State`。
- `:Subnet`：屬性包含 `SubnetId`, `VpcId`, `AvailabilityZone`, `CidrBlock`。
- `:EBSVolume`：屬性包含 `VolumeId`, `Size`, `VolumeType`, `State`, `Encrypted`。

- `:S3Bucket`：屬性包含 `Name`, `CreationDate`, `Arn`。

4.2 核心關係（Relationships）

- `(EC2Instance)-[:IS_MEMBER_OF]->(SecurityGroup)`
- `(SecurityGroup)-[:HAS_RULE]->(SecurityRule)`
- `(EC2Instance)-[:LOCATED_IN]->(Subnet)`，`(Subnet)-[:LOCATED_IN]->(VPC)`
- `(EBSVolume)-[:ATTACHES_TO]->(EC2Instance)`

5 核心分析功能與範例查詢

本系統聚焦三大分析場景：資安漏洞分析、故障衝擊分析與成本優化分析。以下提供代表性 Cypher 查詢。

5.1 資安漏洞分析（Security Vulnerability Analysis）

目標：找出所有暴露於公網且開啟高風險連接埠（如 SSH:22, RDP:3389）的主機。

Listing 1: 尋找允許 0.0.0.0/0 存取 22 埠之主機

```
// 找出所有允許從任何 IP (0.0.0.0/0) 存取 22 號連接埠的主機
MATCH (instance:EC2Instance)-[:IS_MEMBER_OF]->(sg:SecurityGroup),
      (sg)-[:HAS_RULE]->(rule:SecurityRule)
WHERE rule.sourcecidr = '0.0.0.0/0' AND rule.portrange CONTAINS '22'
RETURN instance.name, instance.instanceid, instance.publicip
```

5.2 故障衝擊分析（Failure Impact Analysis）

目標：識別關鍵節點和網路拓撲結構。

Listing 2: 關鍵節點識別

```
// 找出連接數最多的節點（關鍵節點）
MATCH (n)
WITH n, size((n)--()) as connection_count
WHERE connection_count > 2
RETURN labels(n)[0] as node_type, n.instanceid,
       n.groupname, n.vpcid, connection_count
ORDER BY connection_count DESC
```

5.3 成本優化分析（Cost Optimization Analysis）

目標：找出帳號中的「孤兒硬碟」（Orphaned EBS Volumes）。

Listing 3: 找出未連接至任何 EC2 的 EBS 磁碟

```
MATCH (vol:EBSVolume)
WHERE NOT (vol)-[:ATTACHES_TO]->(:EC2Instance)
  AND vol.state = 'available'
RETURN vol.volumeid, vol.size, vol.volumetype, vol.region
ORDER BY vol.size DESC
```

6 系統介面與操作範例

6.1 命令行介面

```
# 執行綜合分析（推薦）
python main.py --mode comprehensive-analyze

# 使用模擬資料進行完整分析
python main.py --mode full --mock

# 執行特定分析類型
python main.py --mode advanced-analyze --analysis-type security
python main.py --mode advanced-analyze --analysis-type cost

# 啟動視覺化儀表板
python main.py --mode dashboard --host 0.0.0.0 --port 8050
```

6.2 分析結果範例

系統成功檢測出以下問題：

- **安全分析**：12 個過度寬鬆的安全群組規則，6 個未加密資源，16 個孤兒安全群組
- **故障分析**：識別出關鍵節點和網路拓撲結構
- **成本優化**：22 個孤兒 EBS 磁碟（總大小 10,268 GB，預估月成本 $1,026.8），16 個未使用安全群組

7 實作要點

7.1 擷取與載入

- **擷取頻率與版本控管**：定期擷取 JSON 並保留版本，以支援變更比對與回溯。
- **ID 去重與關聯完整性**：以雲端資源原生 ID 作為主鍵，避免重覆匯入；匯入順序先節點後關係。
- **安全性**：妥善保護 API 金鑰，避免將敏感設定納入版本庫。

7.2 查詢效能

- 針對高選擇性屬性（如 `InstanceID`, `GroupID`）建立索引或唯一性約束。
- 對常見路徑查詢調整模式與方向性，減少掃描範圍。

8 結論

本專案成功實現了一個基於 Neo4j 圖形資料庫的雲端基礎設施分析平台，具備完整的圖形資料模型、三大核心分析功能，系統無警告運行，能夠有效檢測出雲端基礎設施中的安全風險、故障點和成本浪費，為基礎設施管理提供數據驅動的決策支援。