**Cartography**

**1. Nedir?**  
**Cartography**, şirketin tüm dijital varlıklarını (sunucular, kullanıcılar, veritabanları, ağlar vb.) bir araya toplayıp harita gibi gösteren açık kaynaklı bir araçtır.  
LinkedIn tarafından geliştirilmiştir.  
Verileri **Neo4j** adlı grafik veritabanına yazar.

Kısaca:  
- "Sistemde neler var?"  
- "Birbirine nasıl bağlı?"  
- "Hepsini görsel bir harita olarak göster!"

**2. Ne İşe Yarar?**

* Tüm dijital varlıkları tek yerde gösterir (cihazlar, servisler, ağ bağlantıları).
* Aralarındaki ilişkileri analiz eder (örneğin: “Hangi kullanıcı, hangi sunucuya erişebilir?”).
* Bulut sistemlerde (AWS, Azure) ya da şirket içi ağlarda veri toplayabilir.
* Güvenlik açıklarını ya da karmaşık yapıların anlaşılmasını kolaylaştırır.

➡️ Özellikle güvenlik ekipleri, sistem yöneticileri ve mimarlar için çok faydalıdır.

**3. Neden Önemlidir?**

* Büyük sistemlerde hangi varlık nerede, neyle bağlantılı, kim kullanıyor gibi sorular çok önemlidir.
* Cartography sayesinde karmaşık sistemler **grafiksel olarak analiz edilebilir**.
* Yanlış yapılandırmalar, güvenlik açıkları ya da kullanılmayan kaynaklar kolayca tespit edilebilir.
* Aynı zamanda:  
  - Ağ haritası çıkarılır  
  - Kritik bağlantılar belirlenir  
  - Riskli yollar analiz edilir

Siber saldırılarda saldırganların ilerleme yolları da bu grafiklerde görülebilir.

**4. Nasıl Kurulur? (Linux Örneği)**

Kurulumda Python ve Neo4j veritabanı gereklidir:

**# 1. Gerekli sistem paketlerini kur**

**sudo apt update && sudo apt install python3-pip git -y**

****

**# 2. Neo4j veritabanını kur (Docker ile)**

**docker run \**

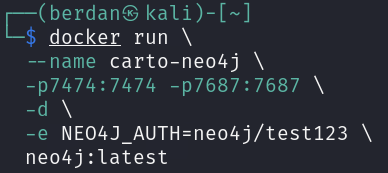
**--name carto-neo4j \**

**-p7474:7474 -p7687:7687 \**

**-d \**

**-e NEO4J\_AUTH=neo4j/test123 \**

**neo4j:latest**

****

**# 3. Cartography aracını indir**

**git clone https://github.com/lyft/cartography.git**

**cd cartography**

****

**# 4. Python bağımlılıklarını yükle**

**pip3 install -r requirements.txt**

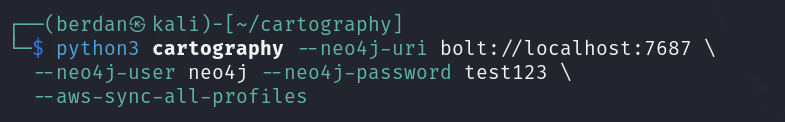
****

**# 5. Örnek AWS taraması yap**

**python3 cartography --neo4j-uri bolt://localhost:7687 \**

**--neo4j-user neo4j --neo4j-password test123 \**

**--aws-sync-all-profiles**

****

🔗 Neo4j arayüzüne **http://localhost:7474** adresinden ulaşabilirsin.

**5. Örnek Bir Anlatım:**  
Bir şirkette AWS altyapısını analiz etmek istiyorsun. Hangi EC2 sunucusu, hangi kullanıcıya bağlı? Kullanılmayan sunucular var mı? Bunları görselleştirmek istiyorsun.

**Cartography** ile bu adımları izlersin:

1. AWS erişim bilgilerini kullanarak veri toplarsın.
2. Veriler Neo4j'e yazılır.
3. Neo4j üzerinden şu tarz sorgular yaparsın:

**// EC2'ye bağlı kullanıcıları bul**

**MATCH (u:AWSUser)-[:RESOURCE]->(ec2:EC2Instance)**

**RETURN u.name, ec2.instanceid**

Böylece, kullanıcıların nelere erişebildiğini **görsel harita ve veri sorgusu** ile analiz edebilirsin.

**Uyarı:**  
Cartography yalnızca analiz ve görselleştirme için kullanılır. Her zaman test ortamlarında ve yetki alınarak çalıştırılmalıdır. Gerçek sistemlerde izinsiz tarama yapmak yasal değildir.