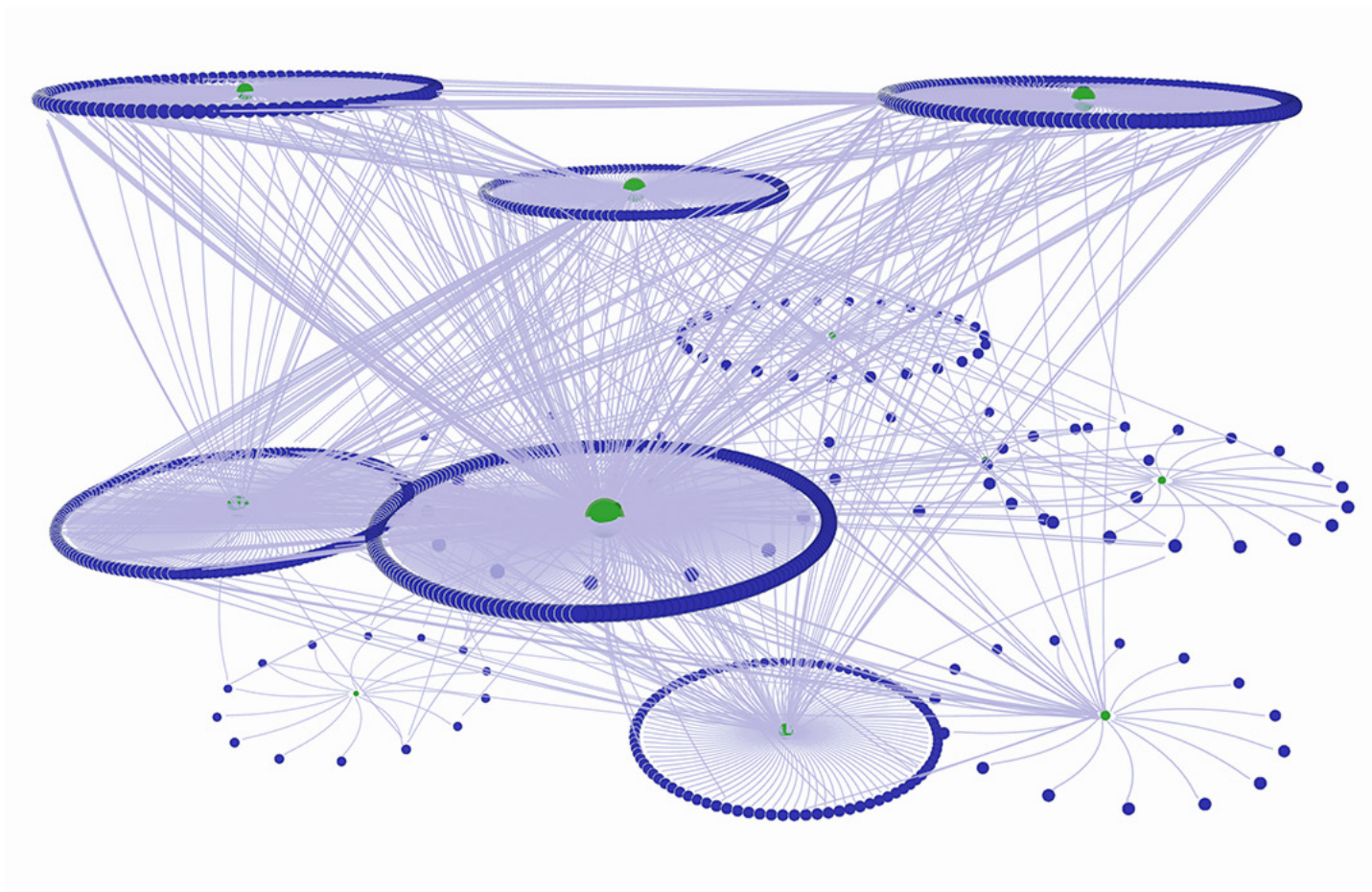


WHITE PAPER

# Cloud Services for Graph Data Provisioning



## Contents

|                                                |   |
|------------------------------------------------|---|
| Overview                                       | 3 |
| Key Benefits of Cloud Services                 | 3 |
| Provisioning Workflow                          | 4 |
| Data Sources                                   | 4 |
| Provisioning Services                          | 4 |
| Mapping to Graph Data                          | 5 |
| Data Enrichment                                | 5 |
| Support for Iterative Exploration and Analysis | 6 |
| Choosing a Provider                            | 6 |
| Performance (Speed)                            | 7 |
| Usability                                      | 7 |
| Cost Savings                                   | 7 |
| Use Cases                                      | 8 |
| References and Further Reading                 | 9 |

### Overview

Cloud services are increasingly available that make it easier and less costly to set up and automate data provisioning workflows for graph data. Companies such as Amazon Web Services (AWS) and others provide serverless storage that can include control of access by user and role, secure backup, and guaranteed speed of retrieval. Cloud microservices priced per query or by processing time let you select, transform, and enrich data on demand or in the background.

Graph data models are both flexible and extensible, and using cloud infrastructure for data provisioning makes it easier to adjust or expand a model when new insights require new data. In addition, cloud services can be deployed only as needed as a graph application progresses from proof of concept to production.

A provisioning workflow can recast data from structured or unstructured sources as entities, relationships, and properties of a graph model, and inject the data into your graph database. By eliminating or reducing the need for in-house servers and IT infrastructure, serverless provisioning can help you focus resources on iterative exploration of the stories that graph data can tell.

### Key Benefits of Cloud Services

- **Cost Savings.** Server infrastructure is owned and managed by the cloud service provider. Pricing per use for both storage and processing reduces cost over an application's life cycle. Economies of scale enable providers to offer low cost or, in some cases, free services.
- **Performance.** Cloud services provide guarantees of reliable storage and secure role-based access. Expected speed of data retrieval and of query or microservice processing is specified.
- **Usability.** Applications for data intake, archival, security are flexible and straightforward to use. Pre-built pipelines can let you transform data without having to use code. Cloud data can support multiple users and browser-based visualization. Support is generally available from both the provider and user communities

## Provisioning Workflow

The generalized workflow illustrated below shows the technical stack and structure for gathering, refining and archiving data in an integrated graph data ecosystem.

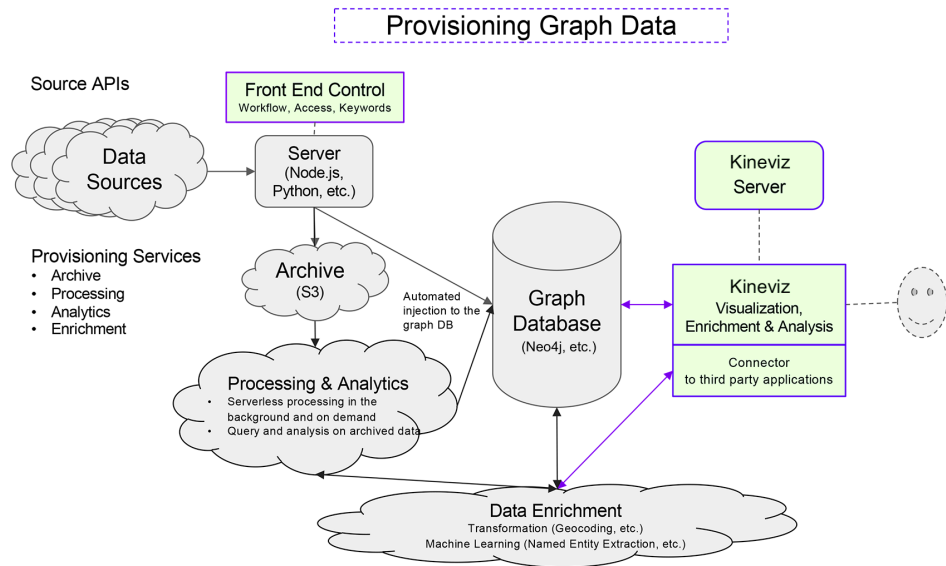


Figure 1. Workflow for provisioning graph data.

## Data Sources

Data can be streamed or queried from public or private data sources. Sources such as social media, government, or academic institutions may make data freely available. Developer APIs to connect to data streams are often available. For example, Twitter provides APIs to its public data through a collection of developer endpoints. Data streams from devices such as sensors, badges, and machinery can be securely accessed as well, through IoT APIs.

The collected data may be stored in private buckets or data lakes. Cloud microservices or other third party applications can be used to further select, process, and analyze data, and to inject data into a graph database such as Neo4j.

## Provisioning Services

Cloud provisioning services fall into two broad areas:

- 1) storage and retrieval infrastructure, and
- 2) programs and interfaces that enable you to collect, analyze, visualize, transform, and enhance data.

## Cloud Services for Graph Data Provisioning

For example, Amazon Web Services (AWS) provides a full-featured and widely used suite of services for collection, storage, data transformation, and enrichment. Microsoft, Google, and others provide broadly similar offerings. Services include:

- Query tools to select data from structured and unstructured sources, store it securely, and transform it into graph data.
- Microservices and interfaces to configure and automate processes both on demand and in the background.
- Machine learning applications to enrich and analyze data.

### *Mapping to Graph Data*

Query tools and APIs available in a cloud environment let you specify a data source, the data to be retrieved, and where to deliver or store the results. You can:

- Map data directly to entities, relationships, and properties in a graph model, and inject it into the graph database.
- Pass data to microservices, for example, an AWS Lambda program, for further processing and injection to your graph database, either on demand or as an automated background process.
- Store the selected data and use query and visualization services (for example, AWS Athena), data definition (AWS Glue) or machine learning (AWS Comprehend) to select, redefine, transform, or enrich data before injecting it into a graph database.

### *Data Enrichment*

Machine learning services based on natural language processing (NLP) let you analyze and enrich data in the cloud (e.g. AWS Comprehend, Microsoft Azure T-NLP, BERT, etc.). Using these services, text documents can be analyzed for overall sentiment or tagged for contextual meaning. These services can provide the data needed for automated market research, monitoring, and AI-enabled applications such as chatbots, to name just a few possibilities.

Textual analysis to add keywords, topics or sentiment is necessarily a batch process. NLP service providers each set their own maximum batch size, currently between 1 and 25 text documents per batch.

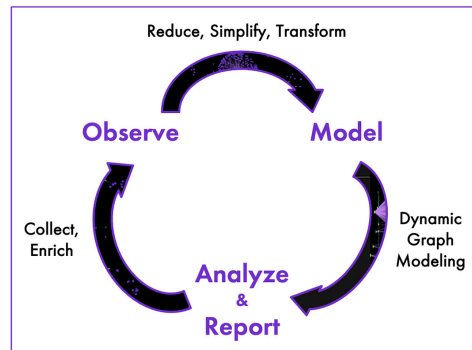
It is also possible to simplify and enrich data within graph environments that provide interfaces to machine learning and transformation applications. Examples include Neo4j's N10s package or the Connector in Kineviz.

## Cloud Services for Graph Data Provisioning

### Support for Iterative Exploration and Analysis

Graph data provides the flexibility to discover new relationships in data, develop new insights, and ask new questions. A generalized process iterates through:

Observation => Modeling => Analysis => New Observation



Depending on the use case, modeling and analysis might lead you to discard, simplify, or transform data, and to add data (nodes, relationships, and properties) from new sources.

The cloud provisioning workflow we've shown provides flexibility to modify data as it's prepared for intake to the graph database as well as during exploration and visualization within the graph environment. Cloud services can be automated, and they can be modified quickly as needs change. Options to pay only for storage used, and per query or per millisecond of processing help control costs.

### Choosing a Provider

Considerations for choosing a cloud service provider center on **performance**, **usability**, and **cost**.

Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and many others provide storage as well as controlled access by user and role, secure backup, and specified speed of retrieval.

Both storage and processing can be purchased as it's used, so there is no need to reserve unused space or processing capacity. This reduces initial storage and IT support costs and improves cost control as projects scale up. For example, Amazon Web Services (AWS) generally provide reliable, secure, technically accessible and cost-effective services.

Some services have a free tier to get started for a relatively low cost of entry. But those that don't offer free services may provide solutions that are a better fit.

## Cloud Services for Graph Data Provisioning

### Performance (Speed)

- **Access.** Processing time for different modes of data access:
  - Streaming, on-demand or in the background
  - Batch processing
  - Retrieval speed from data buckets or longer-term archives
- **Query / Microservices.** Processing time for queries and lambda microservices.

### Usability

Usability depends on factors such as the number and roles of expected users, and the degree of flexibility required. Key considerations include:

- **Automated Provisioning and Archival.** Ease of automating data provision. For example, some services provide pre-built or easily customizable pipelines to select, transform and save data without having to use code.
- **Modifying the workflow.** Ease of gathering additional data from the same or different sources.
- **Enriching data.** Ease of using machine learning services, or of connecting to third party APIs.
- **Secure access by user and role.** Ease of ensuring appropriate user access.

### Cost Savings

The cost of cloud services versus maintaining in-house infrastructure can be compared in terms of:

- **Storage and Computing Hardware**
- **IT and Developer Support**
  - Configuring interfaces, whether pre-built or customized.
  - Maintaining 24/7 access, including backup and failover processes.
  - Managing security and privacy by user and role.

### Use Cases

Serverless provisioning and data management can be a key strategy to jump-start and scale up graph applications in business intelligence, healthcare, cybersecurity, law enforcement, and many others. Even where data security, persistence, or privacy is a consideration, serverless provisioning services can provide much of the needed infrastructure. Some representative use cases are:

- **COVID-19 Contact Tracing**

Employees with internet-connected wearable devices can automatically transmit history of proximity to other employees as well as anonymized COVID-19 infection status. The information is passed to Ki and displayed in tree diagrams that show contact history in near-real time.

- **Sentiment and Influence on Twitter**

Public Twitter data can be streamed to a cloud data store or graph environment (Neo4j and Kineviz) for analysis of user identity, sentiment, and influence. Analysis of unstructured text can be run in the background through APIs to third party machine learning applications.

- **Law Enforcement**

Provide timely information to discover or (ideally) prevent crime. Two basic roles, an Analyst, and Investigator, share similar data but differ in requirements for its use.



## References and Further Reading

### Graph Databases

[Neo4j](#)

[Graph Databases - Overview](#)

[Multi-model databases - Overview](#)

### Cloud Provisioning Services

#### Amazon Web Services (AWS)

[Amazon Web Services](#)

[AWS EC2: Computing Service](#)

[AWS S3: Storage Services](#)

[AWS Glue: Serverless ETL \(Extract, Transform,, Load\)](#)

[AWS Lambda: Processing Microservices](#)

[AWS Athena: On-Demand SQL Query and Analysis](#)

#### Google Cloud

[Google Cloud Services](#)

#### Microsoft Azure

[Microsoft Azure Solutions](#)

### Data Sources

#### Twitter Developer

[Sample a Stream of Tweets](#)

#### Open-Source Intelligence (OSINT)

[OSINT Handbook](#)

[Maltego](#) - OSINT and graphical link analysis

### Data Transformation and Enrichment

#### Transformation

[AddressToGeo](#), [IPToGeo](#) -Geocoding

#### Identity

[WHOIS](#), [Nslookup](#) - IP, Ns lookup

[Shodan](#) - IoT identity

[BotoMeter by OsOMe](#)- Human/bot likelihood

#### Natural Language Processing (NLP) Machine Learning

Sentiment and Topic Analysis with [spaCy](#)

[Sentiment Analysis of Tweets](#)

[AWS Comprehend](#)

[MS Azure Text Analytics](#)

#### spaCy Named Entity Recognition (NER)

[spaCy](#)

[Entity Extraction with spaCy](#)