HELPING THE WORLD MAKE SENSE OF DATA

neo4j

neo4j aura SECURITY

Executive Summary

Security is a priority at Neo4j, and we understand how important your data is to your organization. Customers trust Neo4j to store and process critical, sensitive information. We take that trust – and our responsibility to protect customer data – very seriously. Neo4j is continually working to improve our own security processes and controls while giving our customers security features that protect their information.

With Neo4j Aura and all of our products and services, security is a key design criteria. We stay current on the latest security technologies and industry standard best practices. We continually test to find and fix vulnerabilities throughout the product life cycle. No matter whether data is in flight, at rest, or being processed, we want Neo4j customers to trust the people, systems, and services that handle their critical data.

Neo4j believes that when it comes to security processes and controls, transparency with our customers is essential. We are frequently asked to document our security controls and standards for Neo4j Aura. Customers that store and process data in Aura must know what security controls are in place to protect data and prevent unauthorized access. These are the same concerns that Neo4j has with its own vendors. This white paper provides a detailed understanding of Neo4j Aura's security controls and features as well as a view into our overall approach to information security.

neo4j aura

Neo4j Aura is a fully managed, cloud-native graph data platform that includes Neo4j AuraDB and Neo4j AuraDS. Aura empowers developers and data scientists to quickly build scalable, AI-driven applications and analyze big data with graph algorithms. Aura is fully automated, scalable and secure.

neo4j auraDB

Neo4j AuraDB is a graph database as a service for developers building graph-powered applications. It allows users of our service to build intelligent, context-driven applications faster with lightning-fast queries, real-time insights, using built-in developer tools, data visualization, and integrations supported by the largest graph developer community.

neo4j auraDS

Neo4j AuraDS is a graph data science as a service solution for data scientists building predictive models and analytics workflows. AuraDS unifies the machine learning (ML) surface and graph database into a single workspace and includes pre-tuned algorithms and graph-native ML techniques, making it easy to uncover the connections in big data and answer critical business questions.

Neo4j Aura Data Security

Shared Responsibility Model

Neo4j Aura is a fully managed cloud platform available on multiple commercial cloud service providers and in multiple regions. Offered as SaaS, the platform consists of three major layers, each with a part in the shared responsibility model. As with any modern cloud service, the provider and customers share responsibility for using the service securely.

The foundational layer is the cloud infrastructure that the service runs on. By utilizing state-of-the-art cloud infrastructure providers, Neo4j Aura can rely on unprecedented uptime, security, and reliability in the building blocks that make up the infrastructure underneath Aura.

The middle layer the Neo4j Aura service itself, delivering world-leading graph database services to customers around the world.

The top layer is the customer, including user interaction and configuration. The service configuration is performed by the customer representative, who oversees access and database utilization.

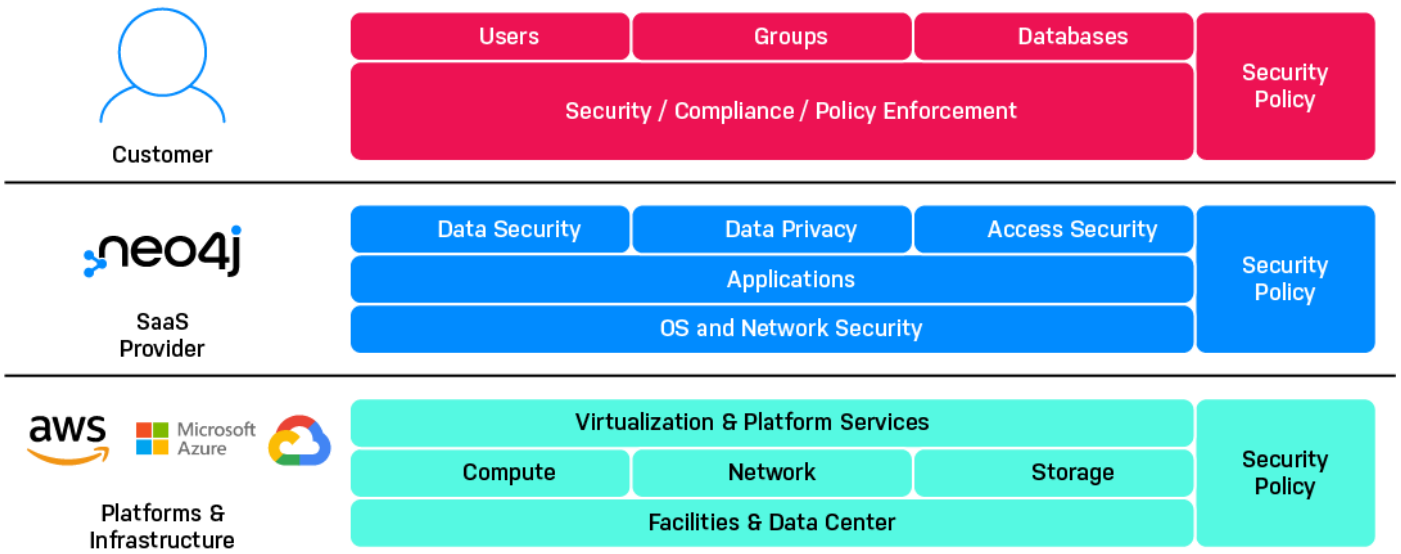

Figure 1. Neo4j Aura Security Shared Responsibility Model

Cloud Service Providers

Neo4j Aura uses infrastructure from Amazon Web Services (AWS), Microsoft Azure, and Google Cloud (GCP), allowing customers to choose the cloud platform they want to use. These providers utilize industry-leading physical and environmental controls, defined through industry standards and frameworks, to protect their data centers. AWS, Azure, and GCP continually undergo security audits and subscribe to a number of certifications to ensure that the data and services they host are safe and secure.

Neo4j Aura can be hosted in any of the provided regions within AWS, Azure, and GCP and configured for customer needs. This allows for ease of use while still maintaining regulatory and customer-specific requirements.

AWS, Azure, and GCP serve a variety of customers, including those in regulated industries. Through their shared responsibility models, they enable customers to manage risk efficiently in the IT environment and provide assurance of effective risk management through their compliance with established, widely recognized frameworks and programs.

A complete list of security and compliance mechanisms can be found on the on the AWS Compliance site, the Azure Compliance site, and the Google Cloud Compliance Resource Center

Identity and Access Management

The Neo4j Aura architecture has two distinct parts:

- **The Aura Control Plane,** which allows customers to administer their Aura services, including creating databases and viewing metrics.
- **The Aura Data Plane,** where the databases run and where their data is stored.

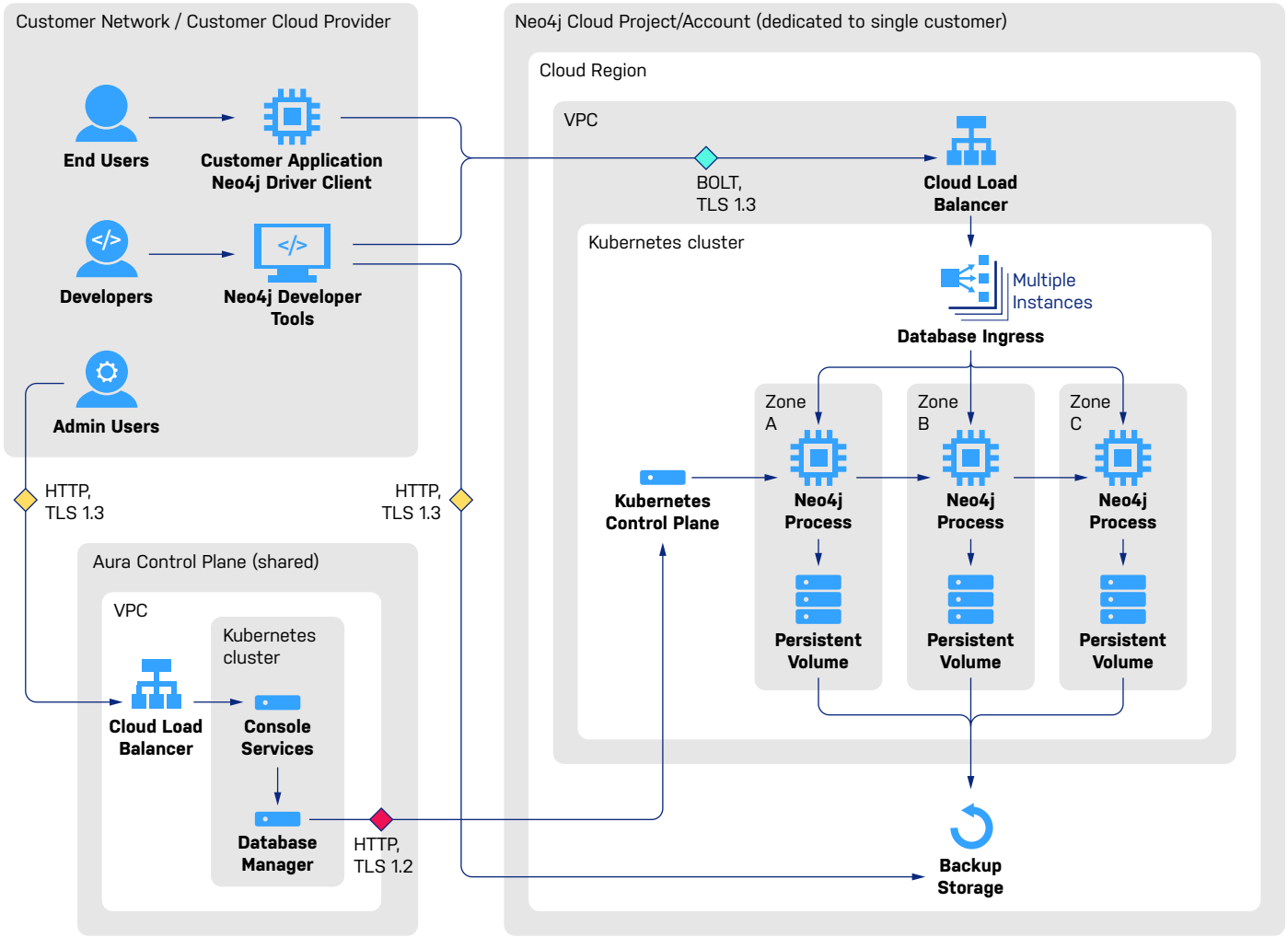

Figure 2. Neo4j Aura Cloud Architecture

Keeping the control plane and the data plane separate makes it easier to reason about their different security responsibilities. The two parts also use separate mechanisms for authentication:

- The Aura Control Plane uses an industry-standard authentication service to secure access for the small number of Neo4j staff required to perform administrative actions.
- Aura uses built-in Neo4j database authentication methods in the data plane. Authentication methods can be configured to allow username and password authentication or SSO services.

Single Sign-On

Single sign-on (SSO) helps customers improve their security posture and reduce risk by allowing them to consolidate users under a single identity with one set of well-protected credentials enabled for all internal and third-party applications.

Aura provides SSO support for the Neo4j Browser and Bloom clients. Aura supports the OpenID Connect (OIDC) protocol and identity providers such as Okta, Active Directory, and Google (authentication only). SSO support is a Neo4j Aura Enterprise feature and requires configuration by the Neo4j support team.

Role-Based Access Control

Aura supports role-based access control (RBAC) within the database. By default, Aura has the following roles:

PUBLIC	All users are granted the public role. By default, this role gives access to the default database and allows execution of procedures and user-defined functions.
reader	The reader role can perform read-only queries on all graphs except for the system database.
editor	The editor role can perform read and write operations on all graphs except for the system database, but cannot create new labels, property keys, or relationship types.
publisher	The publisher role has all the same privileges as the editor, but can also create new labels, property keys, and relationship types.
architect	The architect role has all the same privileges as the publisher, but can also create and manage indexes and constraints.
admin	The admin role has all the same privileges as the architect, but can also manage databases, aliases, users, roles, and privileges.

In addition to these predefined roles, Aura Enterprise customers have the ability to create custom roles for their specific access and security requirements.

Schema-Based Access Control

For fine-grained access control, Aura Enterprise offers a schema-based permissions system. This capability enables administrators to control exactly which parts of a graph are accessible to each role, based on the structure of the graph. This capability can restrict access to specific properties on all nodes with a specific label, which is comparable to schema-based access controls in relational databases. However, the Neo4j database can also restrict the ability to traverse relationships and therefore reach other parts of the graph. This powerful capability has no equivalent in relational or other data models. Learn how to use schema-based access control.

Aura Data Security

Neo4j Aura protects customer data at each stage of its journey through the system:

- Connecting to the Aura database for creating, reading, and updating data
- Importing bulk data into the Aura database
- Storing data in the Aura database
- Backups and exports of the Aura database

The following sections describe how Aura protects data at each stage of the journey.

Connecting to an Aura Database

In the first step of the data journey, users or applications establish a secure connection to their Aura database. This connection is used to create, read, and update data.

Bolt Binary Protocol

All connections to the database use a Neo4j-specific binary protocol called Bolt. It is based on PackStream serialization and supports the Cypher type system, protocol versioning, authentication, and encryption. To use this protocol, customer applications depend on one of the provided native language drivers. Neo4j owns and maintains the driver code and the server code, which gives Neo4j a high degree of control over how the protocol operates, especially in security-critical areas such as encryption and authentication.

Enforced Encryption

While self-hosted Neo4j databases can support unencrypted connections and do not need to validate certificates, Neo4j Aura always requires encrypted connections and ensures that clients validate server certificates when establishing a connection. This means users and applications can be confident that network traffic flowing to and from Neo4j Aura is encrypted.

Server Name Indication (SNI)

Server Name Indication (SNI) is an extension to Transport Layer Security (TLS). SNI allows the client to pass the hostname that the client intends to talk to in the TLS handshake negotiation. SNI is implemented in all of the Neo4j native language drivers, allowing applications to verify the hostname in the server certificate that they receive from Neo4j Aura. This verifies that the server is a valid server (with a valid certificate) and ensures that the server is the one that the application intended to talk to (verifying that the hostname in the CN field of the server certificate matches the host in the application's connecting request URL).

VPC Connections

In Neo4j Aura Enterprise, each database runs inside a dedicated VPC network for each customer, managed by Neo4j. A customer's application will typically run in a separate VPC network, managed by the customer. Neo4j Aura Enterprise offers an optional feature to allow dedicated, private secure connections from applications in the customer's VPC to their database running in the customer-specific Neo4j-managed VPC. This feature gives an extra level of reassurance; not only is the connection encrypted, but it also stays within the private cloud provider network. This feature is currently available only for Neo4j Aura Enterprise and leverages Private Connection capabilities from AWS, Azure, and GCP (see architecture diagrams in Figures 3, 4, and 5). The Neo4j documentation provides more information on this topic.

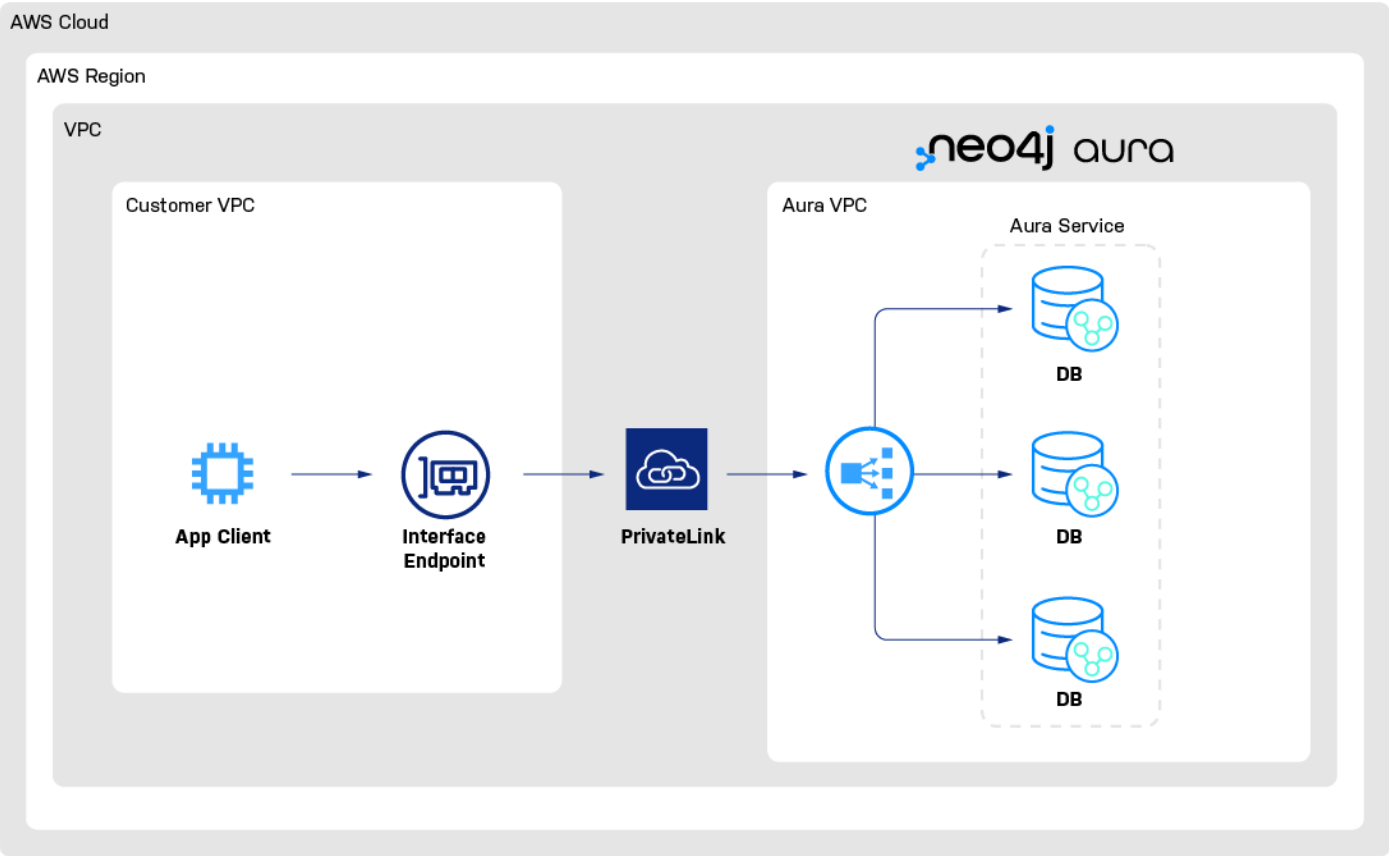

Figure 3. Aura Enterprise on AWS with AWS PrivateLink

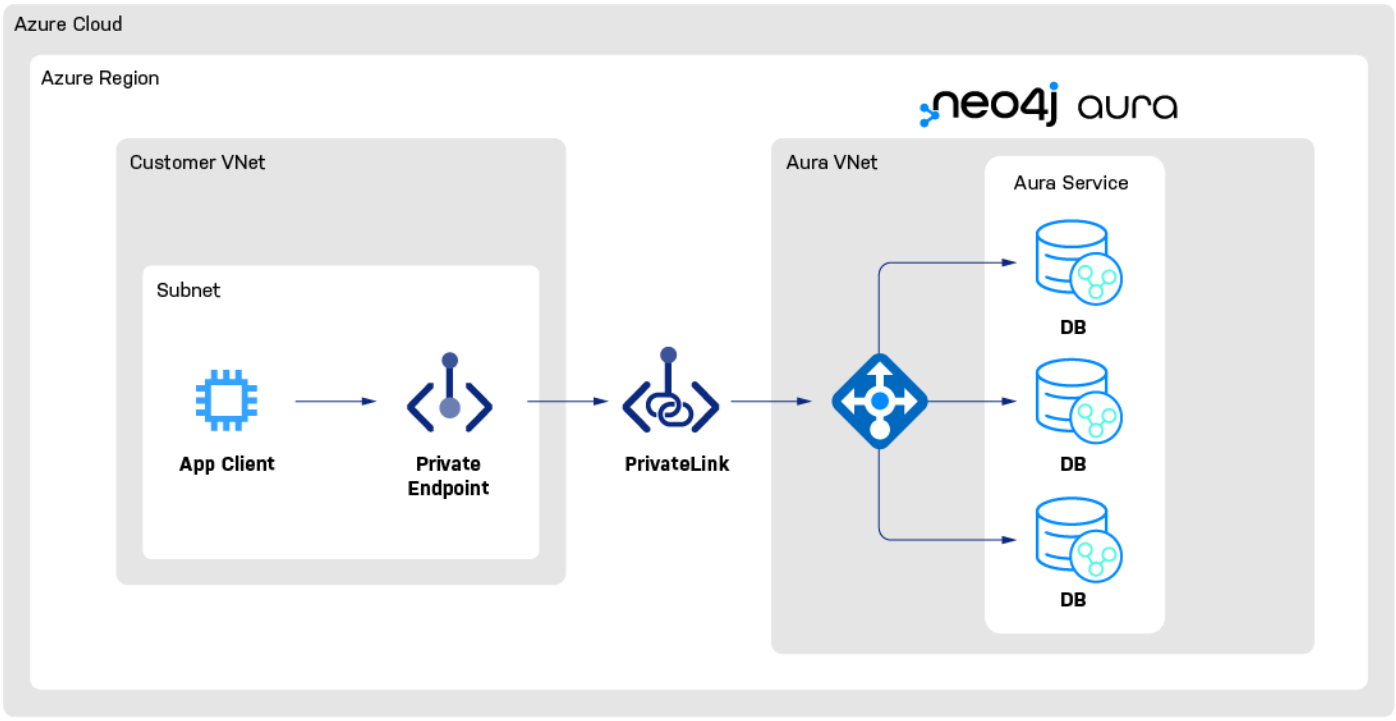

Figure 4. Aura Enterprise on Azure with Azure Private Link

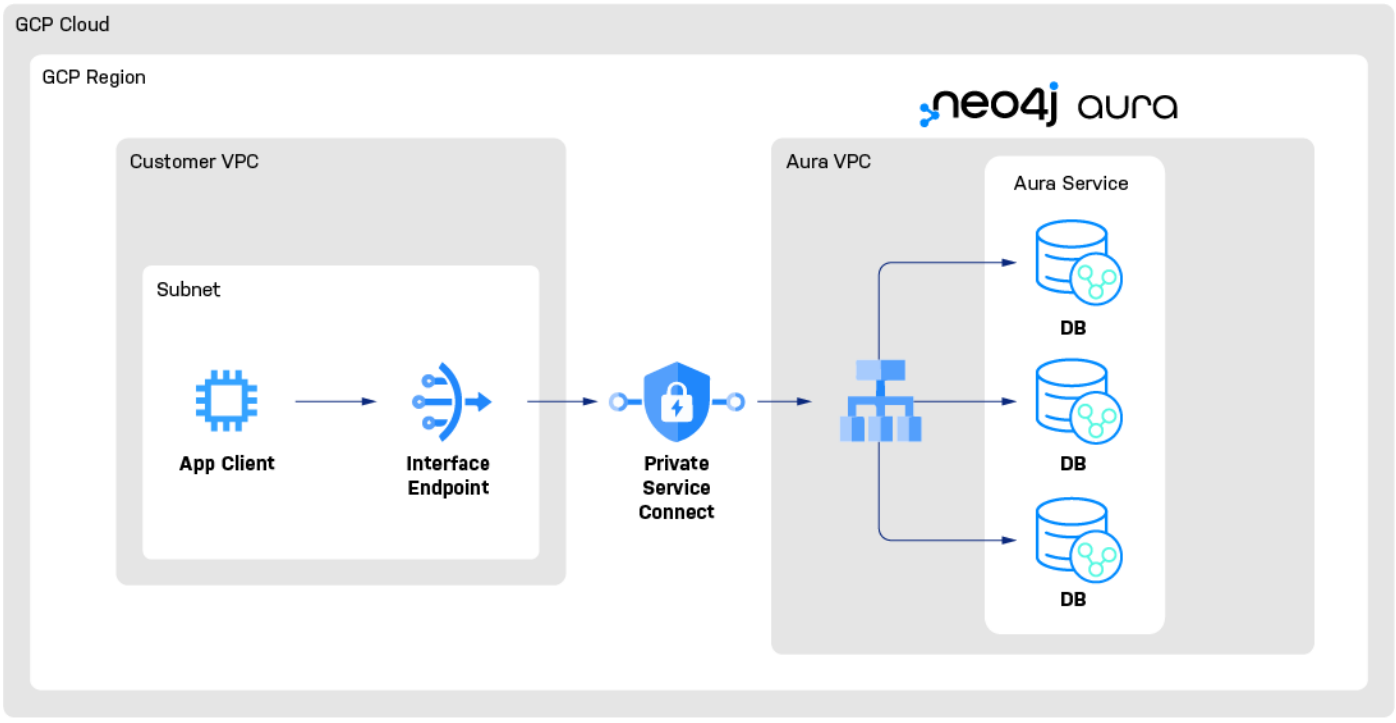

Figure 5. Aura Enterprise on GCP with GCP Private Service Connect

Importing Data into an Aura Database

Neo4j Aura supports bulk import of existing Neo4j databases. In this scenario, the imported data entirely replaces any previous data in the customer's database.

There are two methods for bulk import: uploading data through the Aura Console or at the command line using the database-upload command. Both methods use the Bolt protocol mentioned earlier.

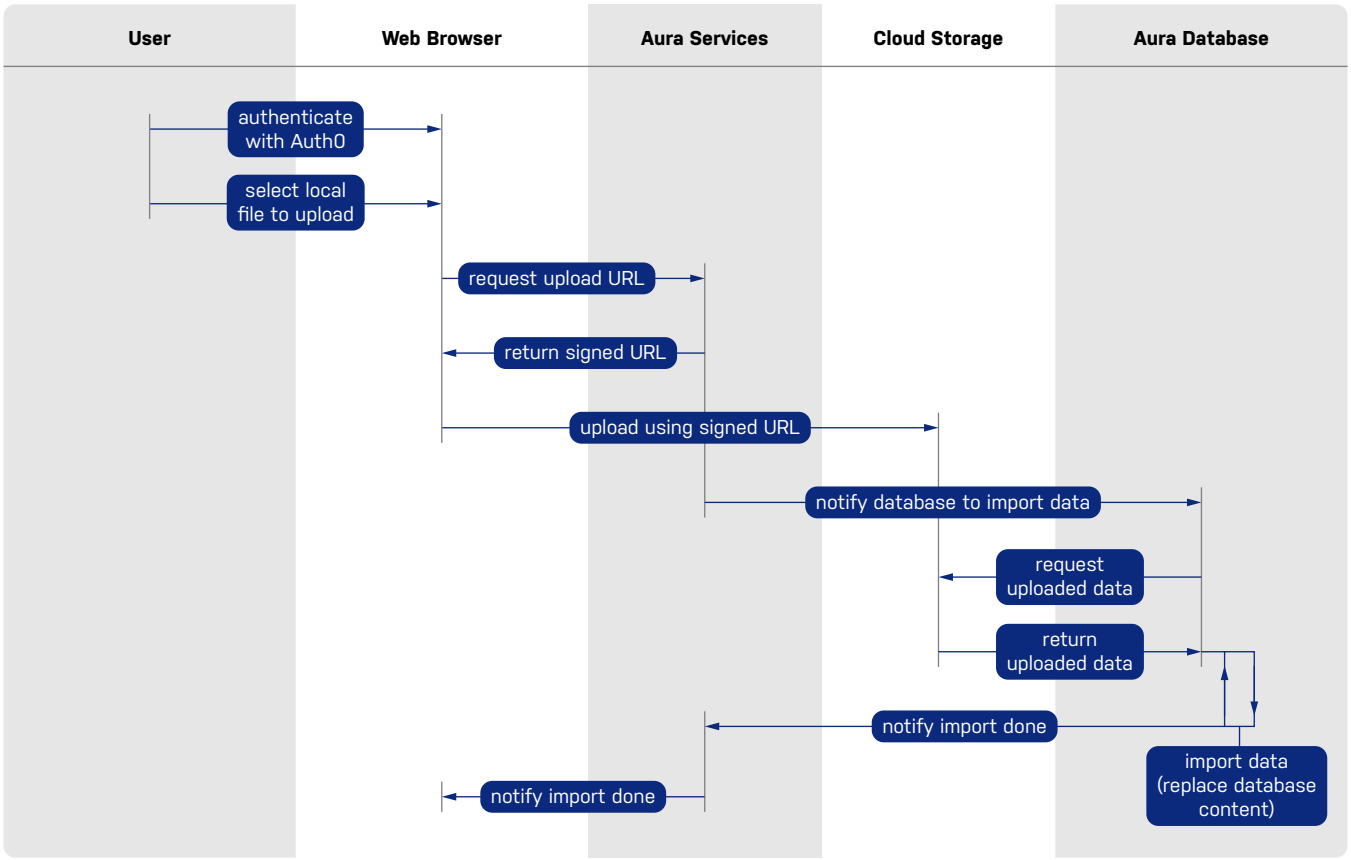

Figure 6. Uploading Data Through the Aura Console

Uploading Data through the Aura Console

The simplest method for importing databases is the Aura Console, which runs in a web browser. This method makes use of standard Auth0 authentication methods, making authentication generally transparent for the user. Behind the scenes, the Aura Console uses its authenticated and validated connection to backend Aura services to obtain a signed URL which it then uses to upload the provided data, as shown in Figure 6.

Uploading Data from the Command Line

Self-managed Neo4j includes a special command called [neo4j-admin database-upload](). This command uses a Bolt connection for bulk file upload. The command efficiently uploads a whole database into Neo4j Aura. This method is suitable for automation and handles larger databases than you can upload through the Aura Console.

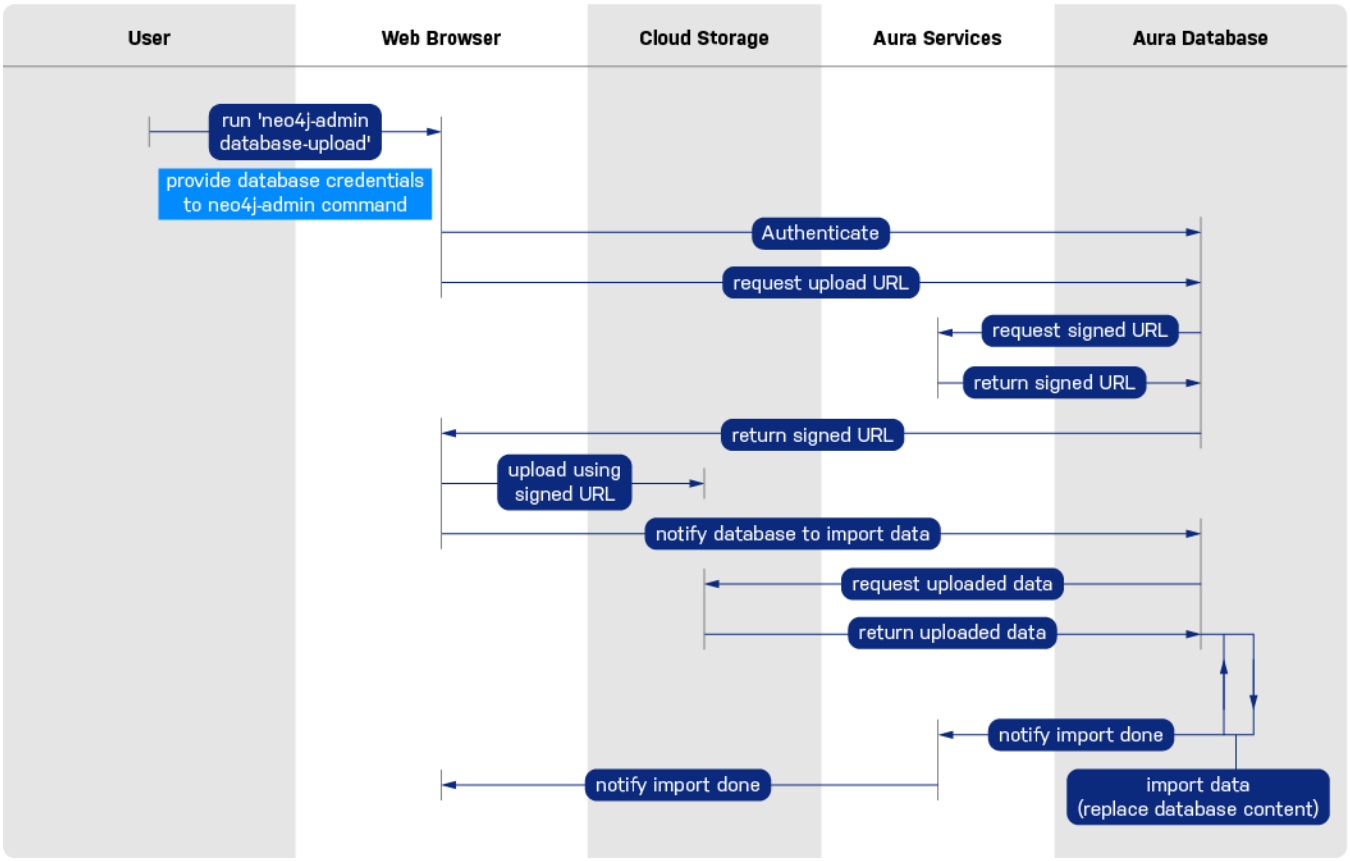

Figure 7. Uploading Data to Aura at the Command Line

Securely Storing Data in an Aura Database

To protect data at rest, Aura uses encrypted data storage capabilities offered by the cloud providers. Whether customers choose to host in AWS, Azure, or GCP, each object store provides server-side encrypted "buckets" for data at rest encryption. By default, AWS, Azure, and GCP encrypt all backup buckets (including the objects stored inside) with AWS SSE-S3 encryption, Azure Storage Encryption (SSE), or Google-managed encryption.

VPC Isolation

In Neo4j Aura Enterprise, each customer's database cluster runs inside a dedicated virtual private cloud (VPC) network with dedicated infrastructure, managed by Neo4j. This VPC serves a single customer, effectively creating a single-tenant system that does not share logical data storage or processing with other customers.

To provide a seamless experience, potentially across multiple VPCs, the Aura Control Plane is shared between multiple customers. This part of the system does not handle any data. (All data is handled by the Aura Data Plane as mentioned earlier.)

Backups and Exports of an Aura Database

Neo4j Aura Enterprise automatically creates backups of each database at regular intervals. Aura stores the data securely in cloud storage buckets.

Users access backups through the Aura Console web application. In the web application, the user can:

- See a list of previous backups
- Choose to restore a backup
- Download a backup (which serves as the export mechanism)

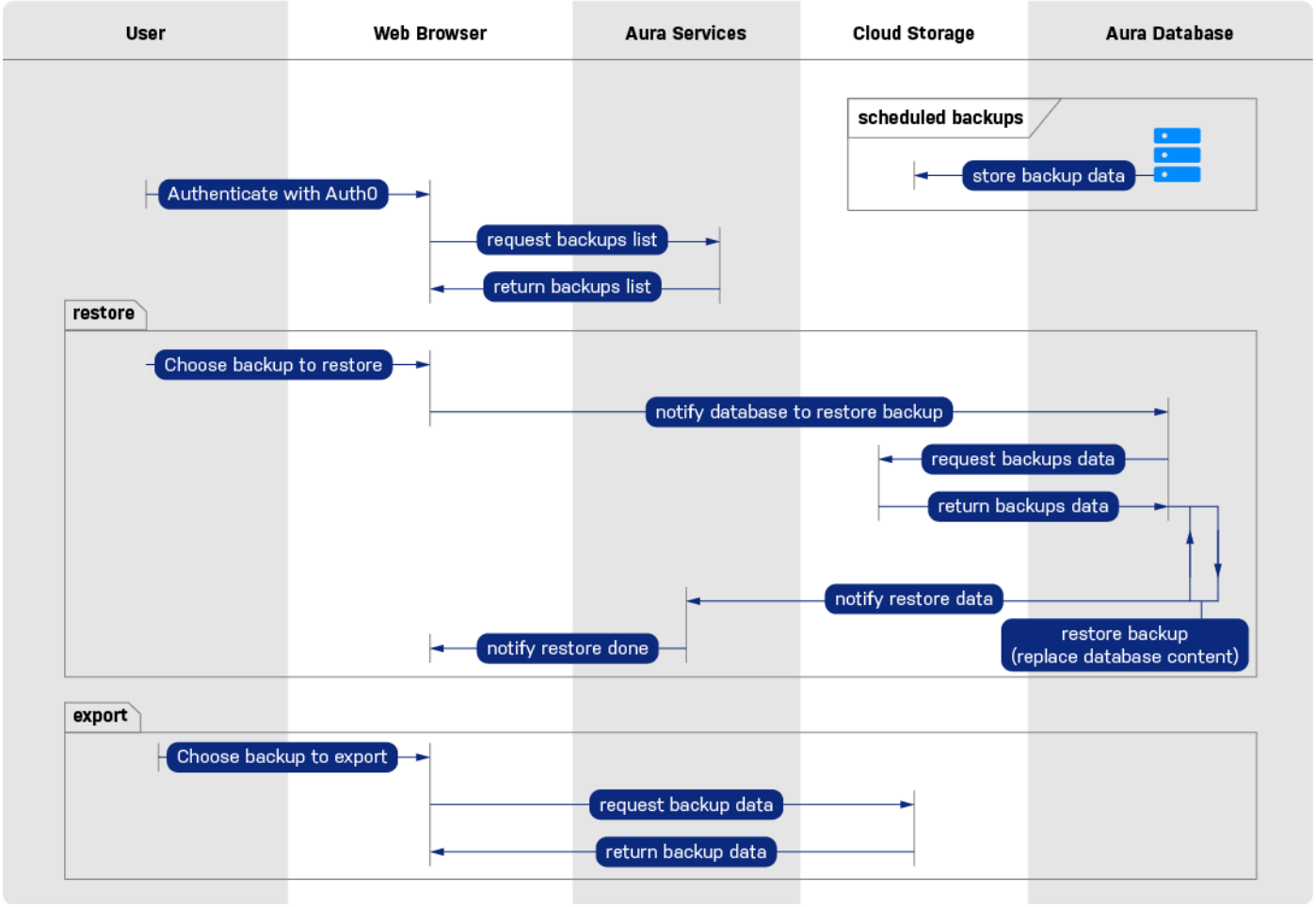

Figure 8. Backups and Exporting Data

Data Access

Data Access by Neo4j Staff

Neo4j Site Reliability Engineers have limited access to Neo4j customer data. Access is granted only to certain authorized personnel within Neo4j. Multifactor authentication is required for those accessing production and customer data.

Access to production and customer data is reviewed regularly and revoked immediately and automatically for employees who leave Neo4j. The Aura Site Reliability Engineering team works with IT and HR to cross-reference and validate the appropriateness of user access to production and customer data. Any issues identified are remediated in collaboration with the customer as needed.

Customer Data Access

Customers connect to Aura by going to console.neo4j.io/, which resolves to the IP addresses of a public cloud load balancer. This load balancer forwards TCP traffic to the Aura Console service. This service handles requests from customers by negotiating an encrypted connection utilizing TLS v1.2 or v1.3 depending on the client support. The documentation lists cipher suites accepted during the TLS handshake.

Customers can also connect directly to their databases using the Neo4j driver for their chosen application or programming language. The driver communicates with Aura using Neo4j's Bolt protocol. Customers connect to their database using a URL that resolves to the public IP addresses of a load balancer in the cloud provider (AWS, Azure, or GCP) where their database is hosted. This load balancer forwards requests to the DB Ingress service running in Kubernetes, which negotiates an encrypted connection with the client using TLS v1.2 or v1.3.

Details about the public IP address and port information needed to provision a firewall configuration is available for both Aura Enterprise and Aura Professional.

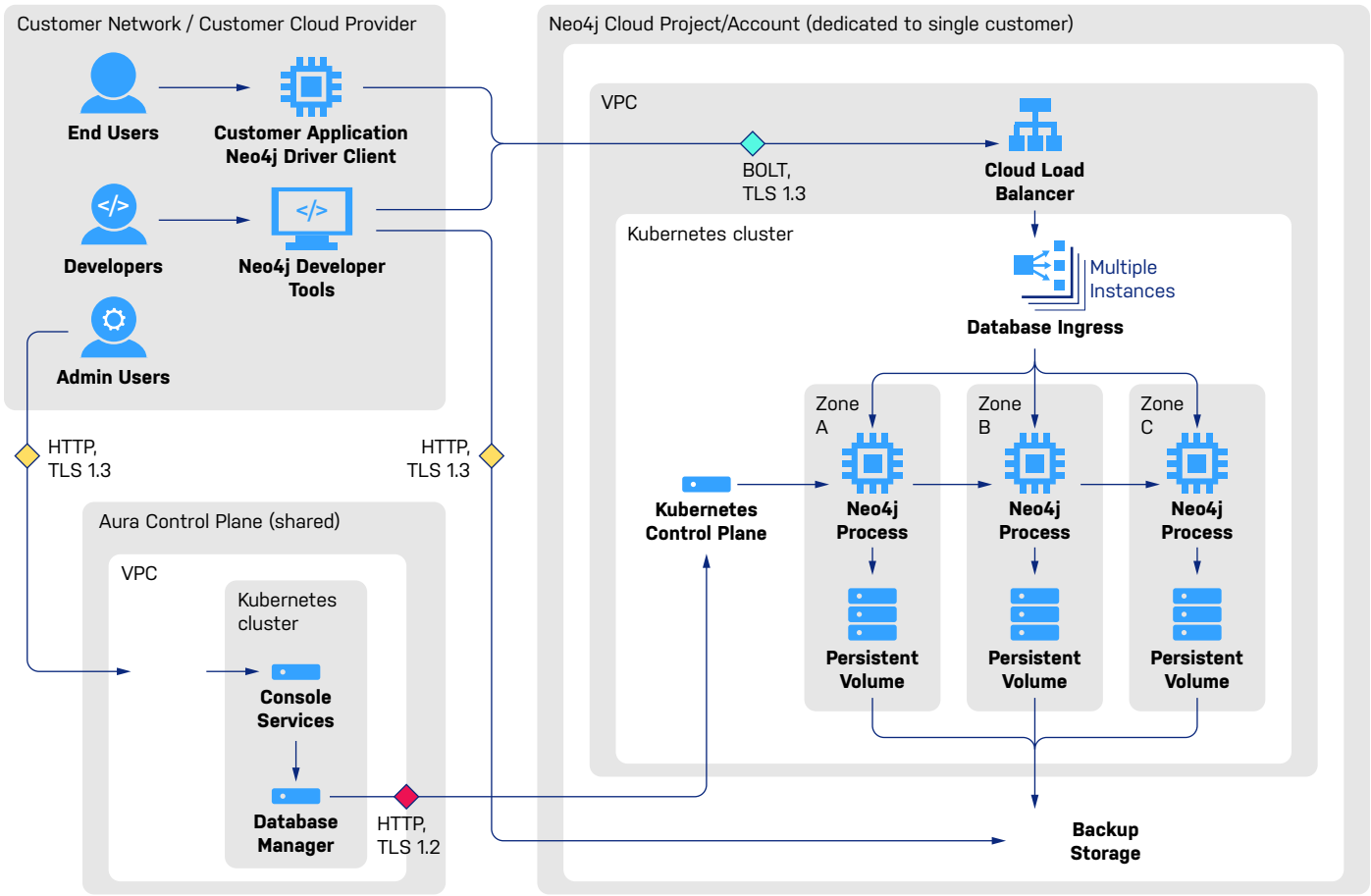

Figure 9. Customer Access to Data in Aura

Security Logs

Security logs are available via the Aura Console. Security logs allow you to identify authenticated users who have connected to and run queries against your Neo4j Aura Database instance in a given timeframe. Security logs are available for up to 30 days per standard data retention policy.

Query Logs

Query logs are available via the Aura Console. Query logs allow you to identify queries that have been run against your Neo4j Aura Database instance in a given timeframe. Query logs are available for up to 30 days per standard data retention policy.

Secure Development

Neo4j follows a secure development policy, which defines security requirements to adhere to during active development and maintenance. The full life cycle of software development is covered, and security principles are applied using a secure-by-design approach.

The Neo4j secure software development life cycle (secure SDLC) captures risk early in the planning stage, where the security requirements are taken into consideration during product and project management.

During development, topics like secure coding practices, vulnerability management, and dynamic security assessment are covered. Neo4j uses automatic code scanning tools built into the CI/CD process to continually evaluate the security of developed code. Tools also scan selected third-party libraries for security flaws.

Verification activities, as well as security assessments and penetration tests, are performed regularly by external parties. The results are reviewed and acted on as appropriate.

Neo4j Aura logically separates production environments from development and testing environments. Aura's cloud environment is both logically and physically separate from Neo4j's corporate offices and networks.

Configuration Management

All production changes are tracked in the source code management system, as well as in an internal ticketing system. A description of each change, communications regarding the change, approvals, and test results are documented in the ticketing system. Tickets are closed when changes are successfully completed. Changes are prioritized and scheduled for deployment to production with minimal impact on the customer. Application code changes require a peer review prior to being implemented into production. Internally, Neo4j personnel are notified of production system changes.

Neo4j uses the full SDLC methodology, which governs the development, acquisition, and implementation of changes. Software is version-controlled, tested, and securely deployed.

Version control software is used to maintain source code versions and migrate source code through the development process to the production environment. The version control software maintains a history of code changes to support rollback capabilities and associates changes with developers.

Input Validation

Neo4j Aura supports the Cypher query language. Cypher injection is a way for maliciously formatted input to jump out of its context, and, by altering the query itself, hijack the query and perform unexpected operations on the database. This is managed by submitting inputs as parameters to the query. See the documentation for more about ways to prevent Cypher injection and about Cypher parameters.

Another way to validate inputs and prevent malicious attacks is to use schema constraints. Neo4j supports a variety of schema constraints to ensure data integrity.

Business Continuity and Disaster Recovery

Availability

Neo4j monitors the capacity utilization of physical and computing infrastructure both internally and for customers to ensure that service delivery matches service level agreements. Neo4j evaluates the need for additional infrastructure capacity in response to growth of existing customers and the addition of new customers. Infrastructure capacity monitoring includes (but is not limited to) disk storage, CPU utilization, and memory utilization.

Cloud Backup

Neo4j has automated daily backups of customer data, which are backed up and retained according to the customer-configured retention policy. Neo4j maintains a disaster recovery plan that defines and documents Neo4j's procedure for recovery of systems containing production and customer data in the event of a disaster. Testing the plan includes restoring backups to validate the integrity of the backup data.

Backups are performed and stored in the cloud infrastructure, where the data is encrypted by the cloud provider.

Incident Response

An incident response plan is in place to guide personnel responding to events affecting integrity, availability, and confidentiality of services or data. Neo4j conducts a root cause analysis as part of the incident management process.

Service Recovery

Neo4j Aura creates, configures, and operates dedicated clusters on infrastructure provided by AWS, Azure, and GCP. In the event of a service outage, Neo4j has a business continuity and disaster recovery plan that goes into effect to bring the service back online. This plan includes working with Neo4j's cloud providers in the event that their services caused the outage. Our infrastructure providers hold numerous certifications and audit reports for these processes and controls. For more information, see the AWS Compliance, Azure Compliance, and GCP Compliance sites.

Customer Onboarding and Offboarding

Customer Onboarding

Each customer environment is provisioned in accordance with the product tier purchased. We create dedicated user accounts for each named user to enable access to the appropriate services (such as the Aura Console and the Support portal). Each user account is restricted to a single named environment and is not shared.

Customer Offboarding

Upon receipt of written confirmation from the customer that services are no longer required or upon the formal end of a defined subscription period, the environments and all associated user accounts are decommissioned based on the agreed timeframes with the customer. All associated user accounts, database instances, storage disks, backups, query logs, cloud projects, provisioning records, configuration details, and all other dedicated cloud services are deleted from all systems, including internal Neo4j systems and identity management systems. The customer data is deleted after successfully transferring it to an approved client repository. In addition, customer access to the Neo4j portal is revoked.

Customer Support

Neo4j Aura offers the following levels of support based on the chosen plan.

	Free	Professional	Enterprise
Aura product documentation	●	●	●
Aura Support Portal	●	●	●
Aura Feedback Portal + Changelog	●	●	●
Neo4j Community Site	●	●	●
Ticket-based support (with no SLA), via the Support Portal		●	●
Premium ticked-based support, via the Support Portal			●

Support Channels

Support Portal - Zendesk
Customers will receive an account for the Neo4j Aura Support Portal. Customers raise support tickets that are mapped to their own project and organization. The Support Portal allows customers to create tickets, view their status, and engage with the Aura Support team.

Within the Support Portal, customers have access to knowledge base articles. Customers can provide feedback on the quality of the content.

Chat via Slack
Customers can opt to engage with Aura Support through a shared Slack channel. This channel is best suited for simple and light touch queries or for synchronization and immediate attention.

For example, an authorized user can request that privileges be revoked in accordance with any staff changes.

Discord
Neo4j offers customer support via a dedicated Aura Discord channel. The community of Discord content as well as some members of the developer relations and support teams actively participate in these exchanges.

Community
Users and customers can post issues and provide feedback by posting on the Neo4j Community forum. Customer Support and Developer Relations team members act as moderators and regularly take part in answering questions and providing solutions and guidance to the community.

Customer Support Service Level Agreements

Neo4j Aura will achieve a Monthly Uptime Percentage of at least 99.95% for each calendar month. Response times for Aura Enterprise customers vary according to the severity of the issue.

Response Times	Initial Response
Severity 1	1 hour
Severity 2	4 business hours
Severity 3	8 business hours
Severity 4	8 business hours

Severity 1: A problem that severely impacts your use of the software in a production environment (such as loss of production data or production systems not functioning). The situation halts your business operations and no procedural workaround exists.

Severity 2: A problem where the software is functioning but your use in a production environment is significantly affected. The situation causes a high impact to portions of your business operations and no procedural workaround exists.

Severity 3: A problem that involves partial, noncritical loss of use of the software in a production environment or a development environment. For production environments, there is a medium-to-low impact on your business, but your business continues to function, including by using a procedural workaround. For development environments, the situation causes your project to no longer function or prevents you from migrating into production.

Severity 4: A general usage question, reporting of a documentation error, or recommendation for a future product enhancement or modification. For production environments, there is low-to-no impact on your business or the performance or functionality of your system. For development environments, there is a medium-to-low impact on your business, but your business continues to function, including by using a procedural workaround.

Find out more about these support terms.

Information Security Program

Neo4j's executive leadership team looks to continuously improve the security of the business and Neo4j products. In an uncertain world, with a range of cyber threats that constantly change and adapt, Aura security risk is regularly reviewed. The executive team has an executive champion, backed by a security team that monitors and provides support across the business, within an evolving governance regime. Neo4j senior leadership is briefed regularly on the status of information security. The board of Neo4j has signed off on a rolling security program to enhance information security, and the Neo4j CEO has endorsed the program internally.

The Neo4j information security program enables the business by:

- Creating and maintaining a set of information security policies, standards, and procedures
- Maintaining a risk management regime to identify and address risks to the integrity, availability, and confidentiality of information and assets in the custody of Neo4j
- Overseeing the implementation and maintenance of security controls to make sure that information security policies, standards, and procedures are adhered to appropriately
- Driving security awareness in the organization

The program is designed, implemented, and maintained by a dedicated security team, reporting to the Neo4j Chief Information Security Officer and under the governance of the Information Security Management System Committee. The security program ensures that Neo4j appropriately manages security risk and that security is optimized to support the business.

Compliance

Compliance and certification activities provide assurance that an appropriately managed security regime is in place and evolves with the changing business environment. Our compliance program is an important initiative to strengthen Neo4j's information security posture.

ISO 27001

Neo4j has built an information security program with a focus on a stable foundation. To achieve this, Neo4j maintains a certified information security management system based on the ISO 27001 standard. This ensures a continuous approach to managing, developing, and maintaining safe and secure products and services.

We believe that having a proper security management system in place allows for a mature security posture. It paves the way for quicker and more accurate decision making, and builds the necessary channels of communication between product development, information security, and the Board.

SOC2

The Service Organization Controls (SOC) framework establishes a standard for controls that safeguard the confidentiality and privacy of information stored and processed in the cloud. Neo4j is SOC2 Type 2 certified.

GDPR

The General Data Protection Regulation (GDPR) standardizes data protection laws across all countries in the EU and imposes strict rules for controlling and processing personally identifiable information. Neo4j manages its data responsibilities with its partners and customers, ensuring that important data is appropriately protected with clearly identified lines of responsibility and the management and deployment of appropriate measures.

CCPA

The California Consumer Privacy Act (CCPA) is a state statute intended to enhance privacy rights and consumer protection for residents of California. Neo4j is compliant with CCPA.

EU-US Privacy Shield

Neo4j maintains its commitment to the Privacy Shield principles and its certification under Privacy Shield is current. Privacy Shield is not a valid international data transfer mechanism at present due to court action in the EU. Neo4j will transition to any new international data transfer frameworks as they become available for EU-to-US data transfers.

Training & Awareness

The Neo4j information security program includes training and awareness to ensure that everyone at Neo4j is actively working to keep information safe and secure and is aware of current threats and risks. Today the training program offers training as needed, together with annual awareness sessions.

We expanded our training program for developers in 2022, which included the delivery of a Neo4j security academy, with on-site secure development training tailored to Neo4j engineers. We plan to expand this program in 2023 and provide a continuous training and awareness platform providing targeted training for developers to help maintain individual skill sets and make security part of the DNA of our development cycle. We also maintain a general security awareness regime.

Vendor Management

Neo4j has a vendor management policy to oversee the onboarding of vendor services and products. Vendors are asked to complete a security questionnaire, which allows the security team to review their security posture and their capability to protect the data they are entrusted with.

Non-disclosure agreements (NDA) are used to protect the agreement, and the information shared, between Neo4j and its vendors.

HR Security

As part of the interview process, HR performs appropriate screening of candidates to ensure their suitability for a role at Neo4j. This screening may differ depending on regulatory requirements and limitations in the country of hire. As part of joining Neo4j, all employees will sign a series of documents, including a nondisclosure agreement and a code of conduct, to name a few.

When leaving a role at Neo4j, departing employees are required to return all assets used during their employment. When a user is offboarded, their user account is disabled in our single sign-on system, which automatically suspends access to Neo4j systems and services.

Conclusion

This document provides information about Neo4j Aura security as well as some of the ways Neo4j addresses the evolving challenges of today's cybersecurity environment. For more information, please contact us at security@neo4j.com or reach out to your Neo4j representative.

Questions about Neo4j? Contact us around the globe:

info@neo4j.com
neo4j.com/contact-us