Secure connections

NOTE

This content is for the classic Neo4j Aura console. Use the version selector on the left to see the new console documentation.

VPC isolation

AuraDB Virtual Dedicated Cloud   AuraDS Enterprise

AuraDB Virtual Dedicated Cloud and AuraDS Enterprise run in a dedicated AWS cloud account, Azure subscription, or GCP project to achieve complete isolation for your deployment. Additional Virtual Private Cloud (VPC) boundaries enable you to operate within an isolated section of the service, where your processing, networking, and storage are further protected. The Aura console resides in a separate VPC, isolated from the rest of the Aura services.

Network access

An Aura instance can be publicly available, completely private, or allow both public and private access.

If public traffic is enabled, your Aura instances are public and traffic to them is allowed to traverse the public internet and they are accessible with the correct username and password.

To make your instance completely private, you need to disable public traffic, use the cloud provider's network, and create a private endpoint inside your VPC, which gives you a private connection to Aura. The only way to connect to your database is from inside your network (your VPC in your AWS/Azure/GCP account) using an internal IP address you choose and DNS records you create.

To configure network access, you need to be authorized to access the part of the infrastructure that runs and handles these instances as well as the networking used to establish sec

AI search

between the database and the application's VPC. This includes the ability to connect over the cloud provider's private link and private endpoint.

To configure settings for network access to your instance, go to **Aura console** > **Security** > **Network access** > **New network access configuration**.

From there, you can either set up a new network access configuration, or edit current configuration settings.

The Aura console provides a step-by-step configuration guide to:

1. Choose your Aura instance details
2. Create an endpoint
3. Accept endpoint connection requests and enable private DNS in the cloud provider's console
4. **Disable public traffic (optional)** If you disable public traffic it is highly recommended to test connectivity through the private endpoint before disabling public traffic.

You can return to Step 4 at any time to disable public traffic, even if you've already completed the network access configuration and initially allowed public traffic. To do this, click through the steps in the network access configuration guide until you reach Step 4, where there is the option to disable public traffic. Disabling public traffic does not take effect immediately. You can monitor the status change in the console to confirm when the process is complete.

To continue accessing Browser and Bloom, you can configure a VPN in your VPC and connect to these services over the VPN.

Private endpoints

Private endpoints are network interfaces inside your own VPC, which can only be accessed within your private network. The cloud provider connects them over their network to Neo4j Aura. By design they are not exposed to the public internet, ensuring that critical services are accessible only through private, secure networks.

A single private link connection applies to all instances in a region. So if you've set one up for `us-east-1` then those network connections will apply to all instances in that region. You can set up a second private link connection to applications that are hosted in a second region (for example `us-west-1`) but still housed inside the same Aura project.

AWS private endpoints

AuraDB Virtual Dedicated Cloud AuraDS Enterprise

AuraDB Virtual Dedicated Cloud and AuraDS Enterprise support private endpoints on AWS using AWS PrivateLink → .

Once activated, you can create an endpoint in your VPC that connects to Aura.

For a step-by-step guide, see the How to Configure Neo4j Aura With AWS PrivateLink blog article.

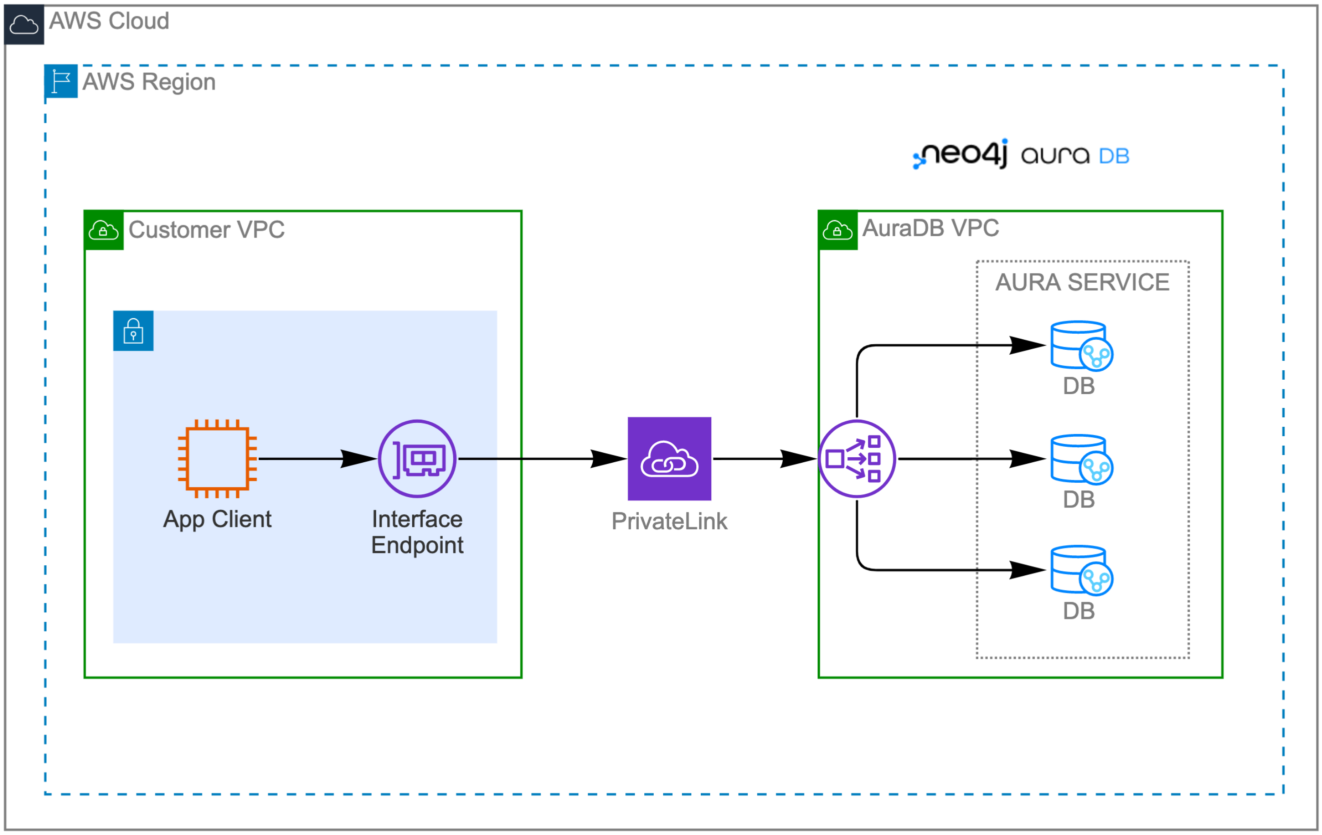

Figure 1. VPC connectivity with AWS PrivateLink

All applications running Neo4j workloads inside the VPC are routed directly to your isolated environment in Aura without traversing the public internet. You can then disable public traffic, ensuring all traffic to the instance remains private to your VPC.

NOTE

NOTE

- PrivateLink applies to all instances in the region.
- When activated, a **Private Connection** label, shield icon, and dedicated **Private URI** will appear on any instance tile using PrivateLink in the Aura Console.
- If you disable public traffic, you must use a dedicated VPN to connect to your instance via Browser or Bloom.
- Connections using private endpoints are one-way. Aura VPCs can't initiate connections back to your VPCs.
- In AWS region us-east-1, we do not support the Availability Zone with ID use1-az3 for private endpoints.

Browser and Bloom access over private endpoints

To connect to your instance via Browser or Bloom, you must use a dedicated VPN. This is because when you disable public access to your instance, this applies to all connections, including those from your computer when using Browser or Bloom.

Without private endpoints, you access Browser and Bloom over the internet:

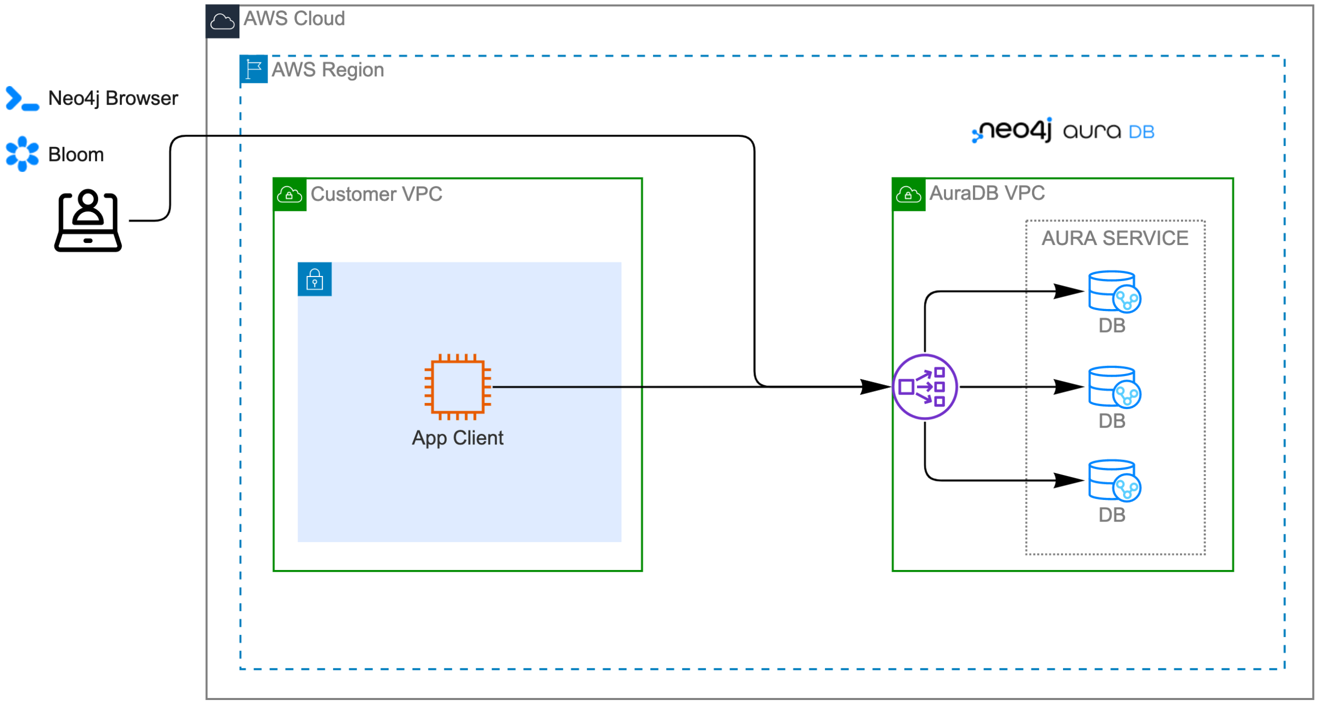

Figure 2. Architecture overview before enabling private endpoints

When you have enabled private endpoints **and** disabled public internet access, you can no longer connect Browser or Bloom to your instances over the internet:

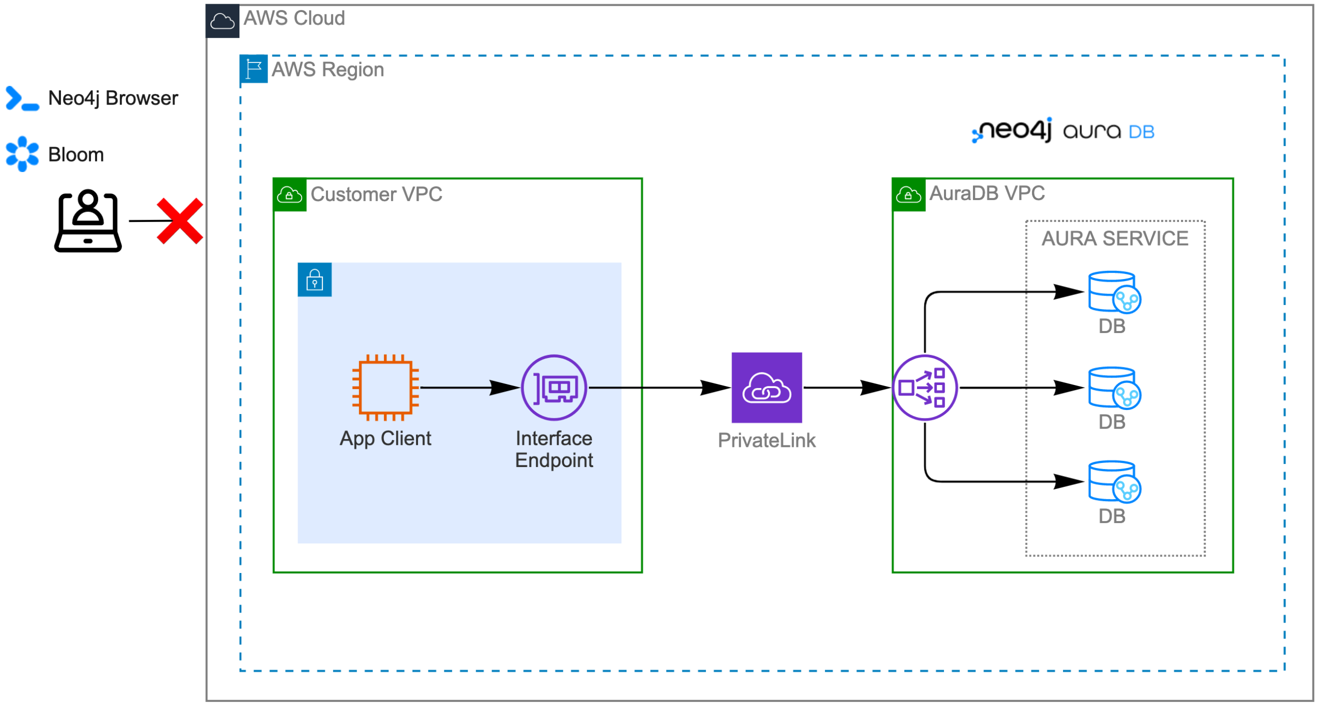

Figure 3. Architecture overview with private endpoints enabled and public traffic disabled

To continue accessing Browser and Bloom, you can configure a VPN (Virtual Private Network) in your VPC and connect to Browser and Bloom over the VPN.

NOTE

To access Bloom and Browser over a VPN, you must ensure that:

- The VPN server uses the VPC's DNS servers → .
- You use the **Private URI** shown on the instance tile and in the instance details. It will be different from the **Connection URI** you used before.

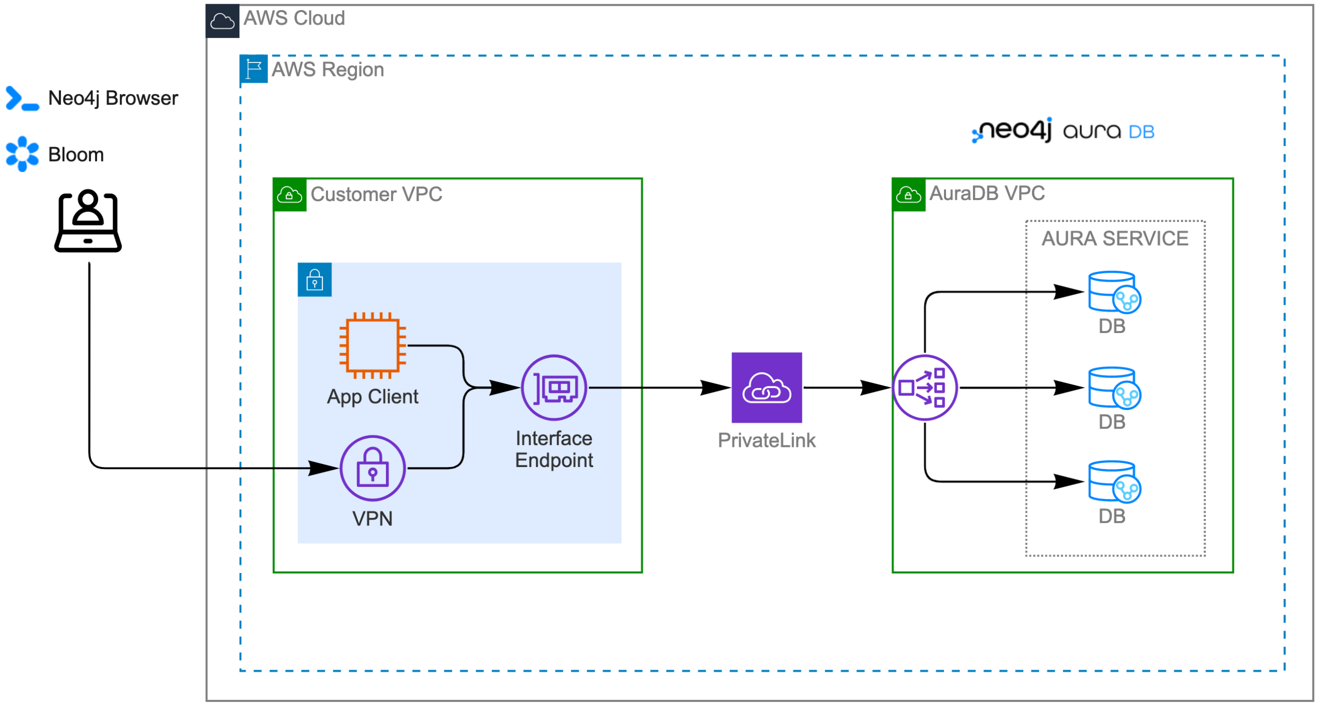

Figure 4. Accessing Browser and Bloom over a VPN

Enable private endpoints

To enable private endpoints using AWS PrivateLink:

1. Select **Network Access** from the sidebar menu of the Console.
2. Select **New network access configuration** and follow the setup instructions.

You will need an AWS account with permissions to create, modify, describe and delete endpoints. Please see the AWS Documentation → for more information.

GCP private endpoints

AuraDB Virtual Dedicated Cloud AuraDS Enterprise

AuraDB Virtual Dedicated Cloud and AuraDS Enterprise support private endpoints on GCP using GCP Private Service Connect → .

Once activated, you can create an endpoint in your VPC that connects to Aura.

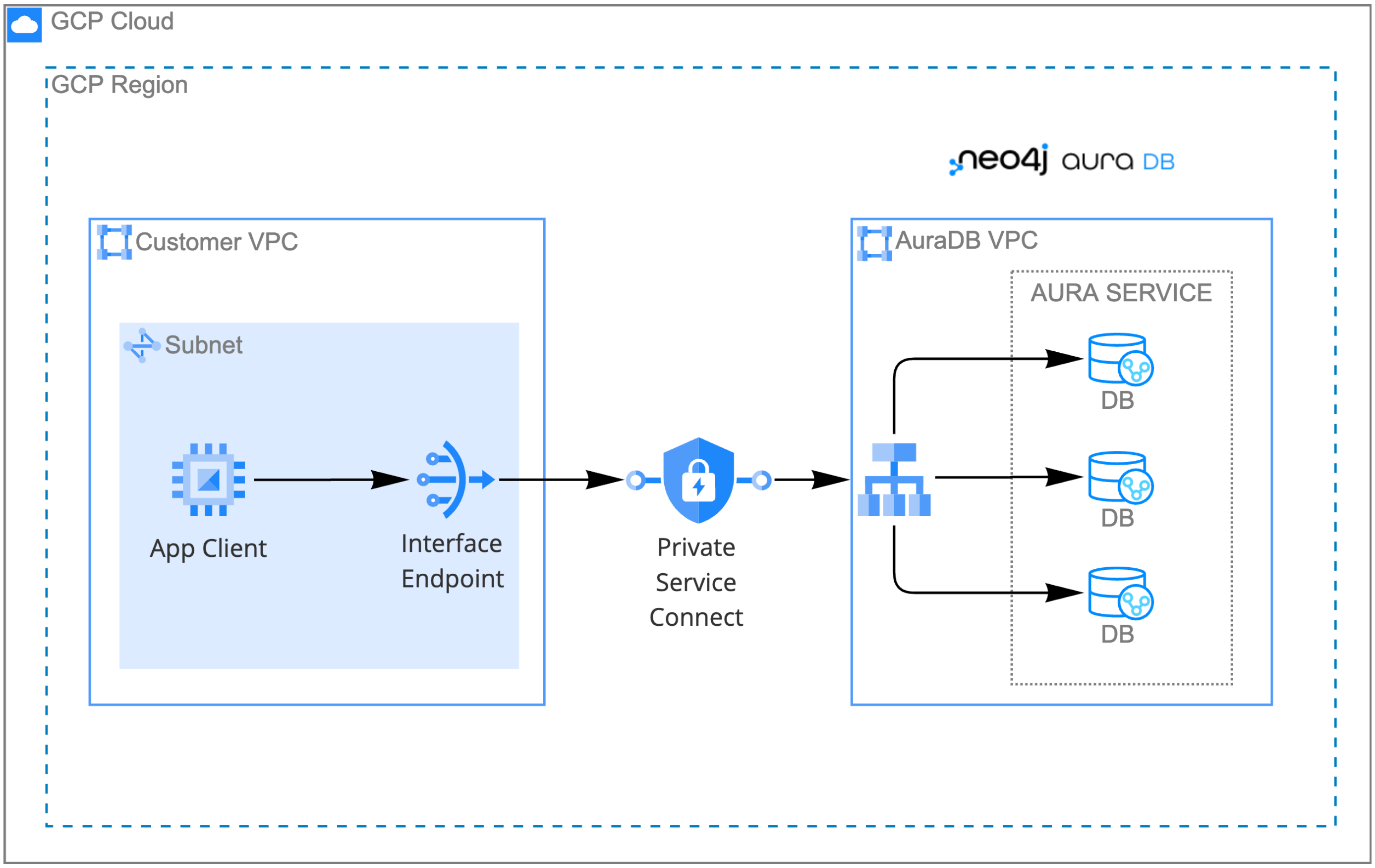

Figure 5. VPC connectivity with GCP Private Service Connect

All applications running Neo4j workloads inside the VPC are routed directly to your isolated environment in Aura without traversing the public internet. You can then disable public traffic, ensuring all traffic to the instance remains private to your VPC.

NOTE

- Private Service Connect applies to all instances in the region.
- When activated, a **Private Connection** label, shield icon, and dedicated **Private URI** will appear on any instance tile using Private Service Connect in the Aura Console.
- If you disable public traffic, you must use a dedicated VPN to connect to your instance via Browser or Bloom.
- Connections using private endpoints are one-way. Aura VPCs can't initiate connections back to your VPCs.

Browser and Bloom access over private endpoints

To connect to your instance via Browser or Bloom, you must use a dedicated VPN. This is because when you disable public access to your instance, this applies to all connections, including those from your computer when using Browser or Bloom.

Without private endpoints, you access Browser and Bloom over the internet:

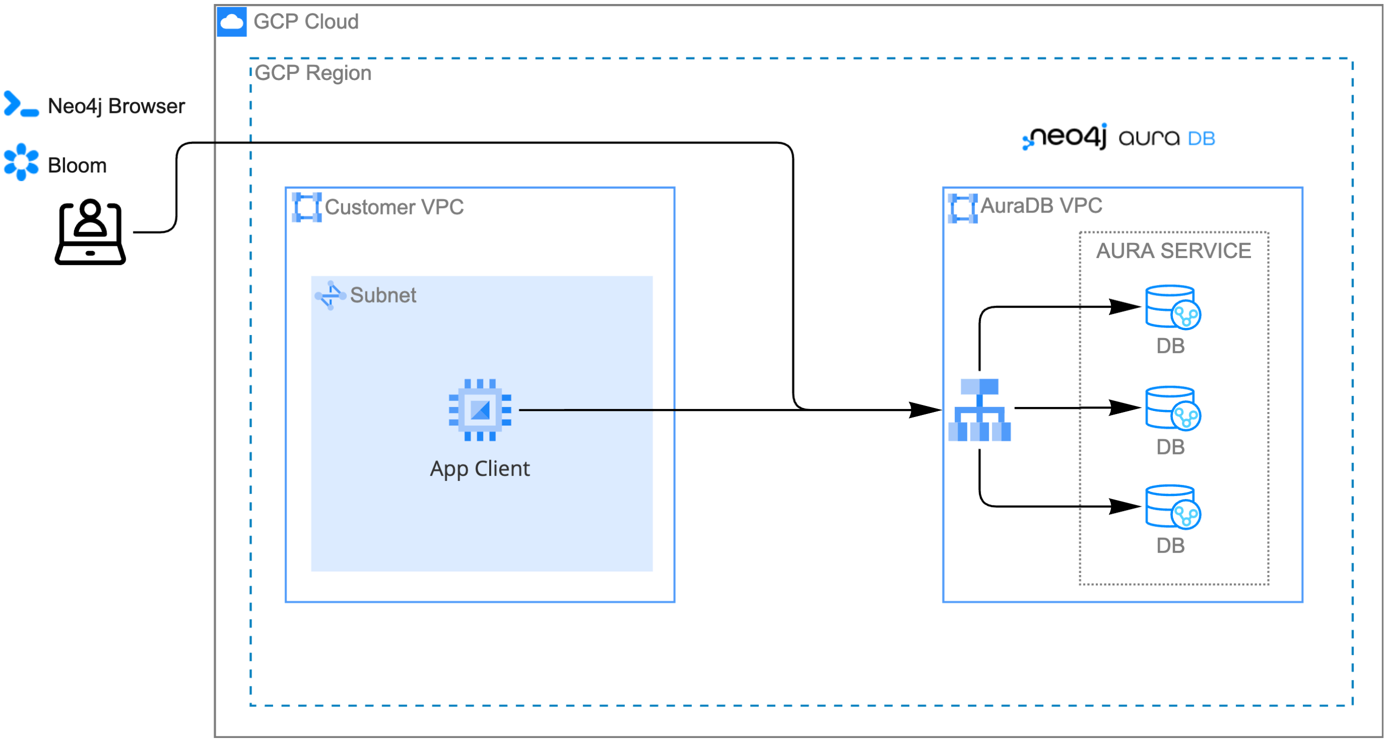

Figure 6. Architecture overview before enabling private endpoints

When you have enabled private endpoints and disabled public internet access, you can no longer connect Browser or Bloom to your instances over the internet:

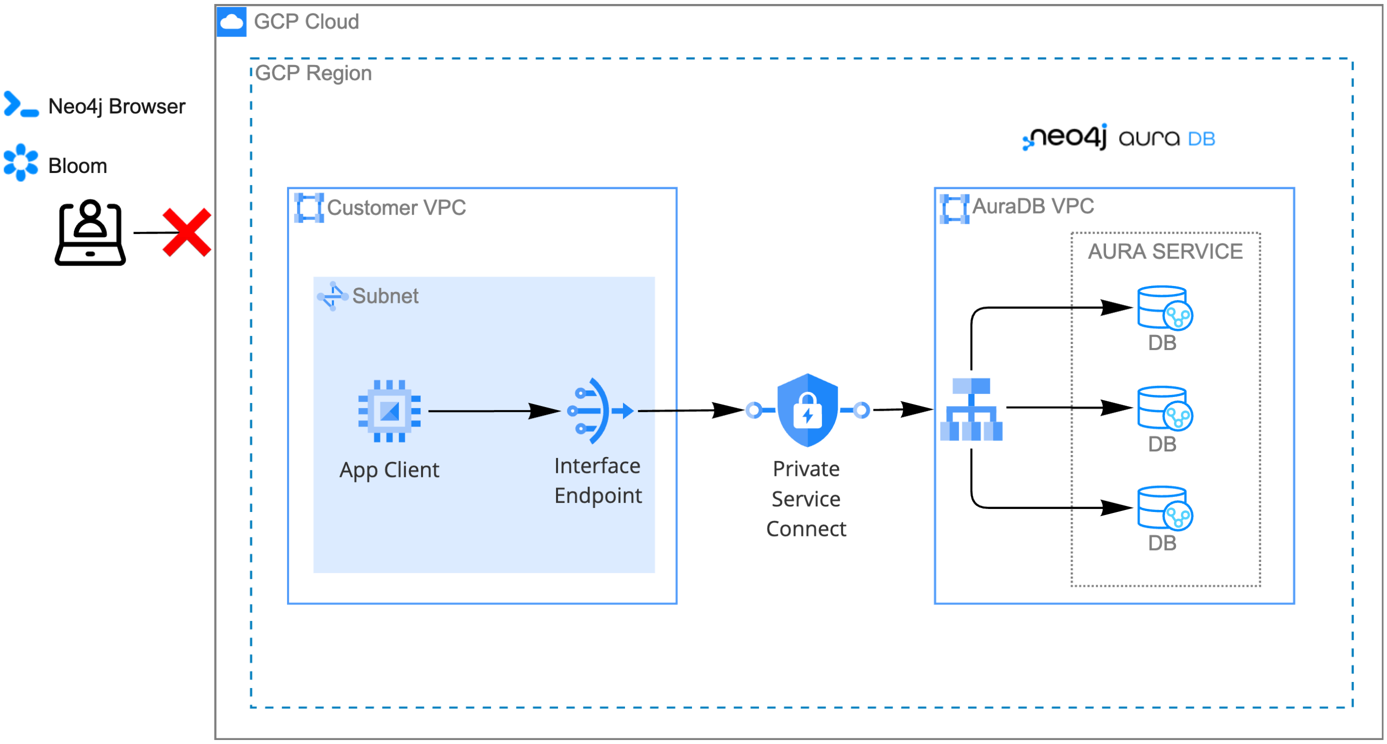

Figure 7. Architecture overview with private endpoints enabled and public traffic disabled

To continue accessing Browser and Bloom, you can configure a GCP Cloud VPN → (Virtual Private Network) in your VPC and connect to Browser and Bloom over the VPN.

NOTE

To access Bloom and Browser over a VPN, you must ensure that:

- You have set up GCP Response Policy Zone → , or an equivalent DNS service, inside of the VPC.
- You use the **Private URI** shown on the instance tile and in the instance details. It will be different from the **Connection URI** you used before.

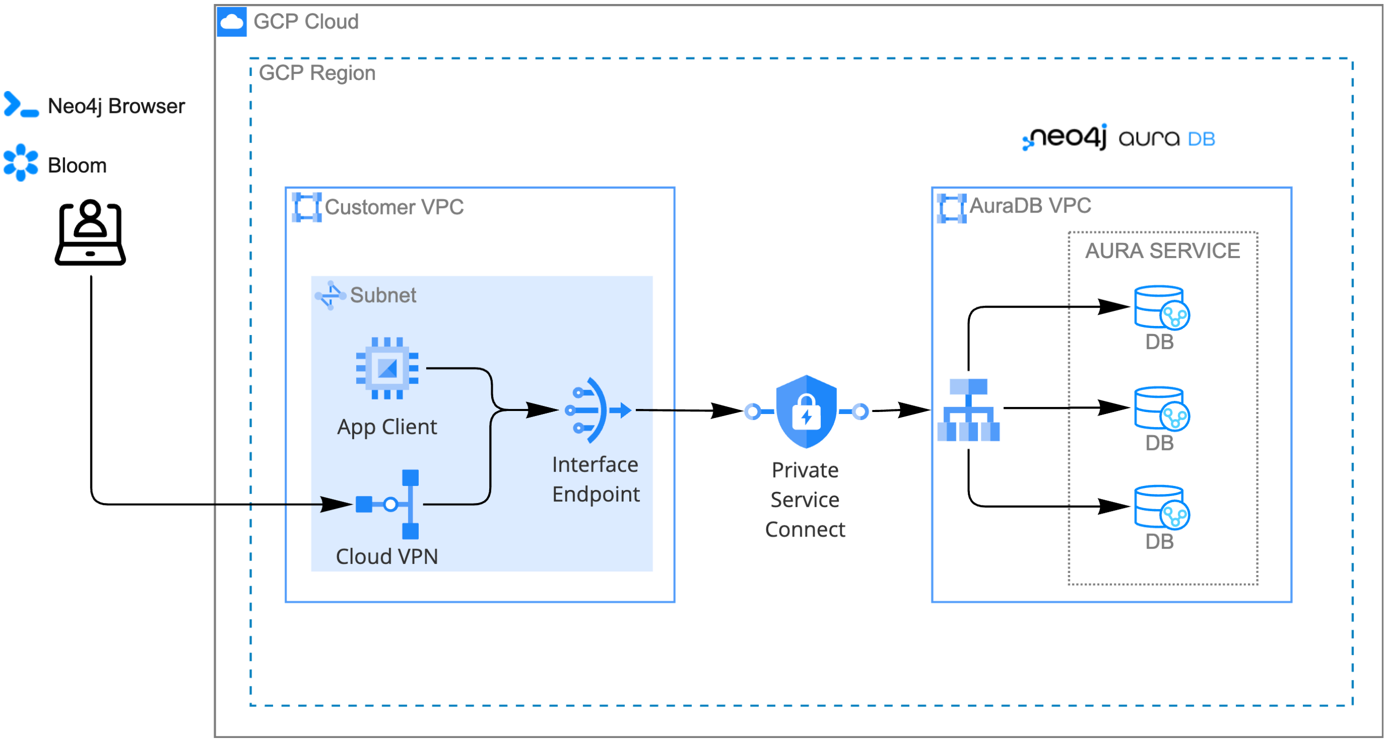

Figure 8. Accessing Browser and Bloom over a VPN

Enable private endpoints

To enable private endpoints using GCP Private Service Connect:

1. Select **Network Access** from the sidebar menu of the Console.
2. Select **New network access configuration** and follow the setup instructions.

Please see the GCP Documentation → for required roles and permissions.

Azure private endpoints

AuraDB Virtual Dedicated Cloud AuraDS Enterprise

AuraDB Virtual Dedicated Cloud and AuraDS Enterprise support private endpoints on Azure using Azure Private Link → .

Once activated, you can create an endpoint in your Virtual Network (VNet) that connects to Aura.

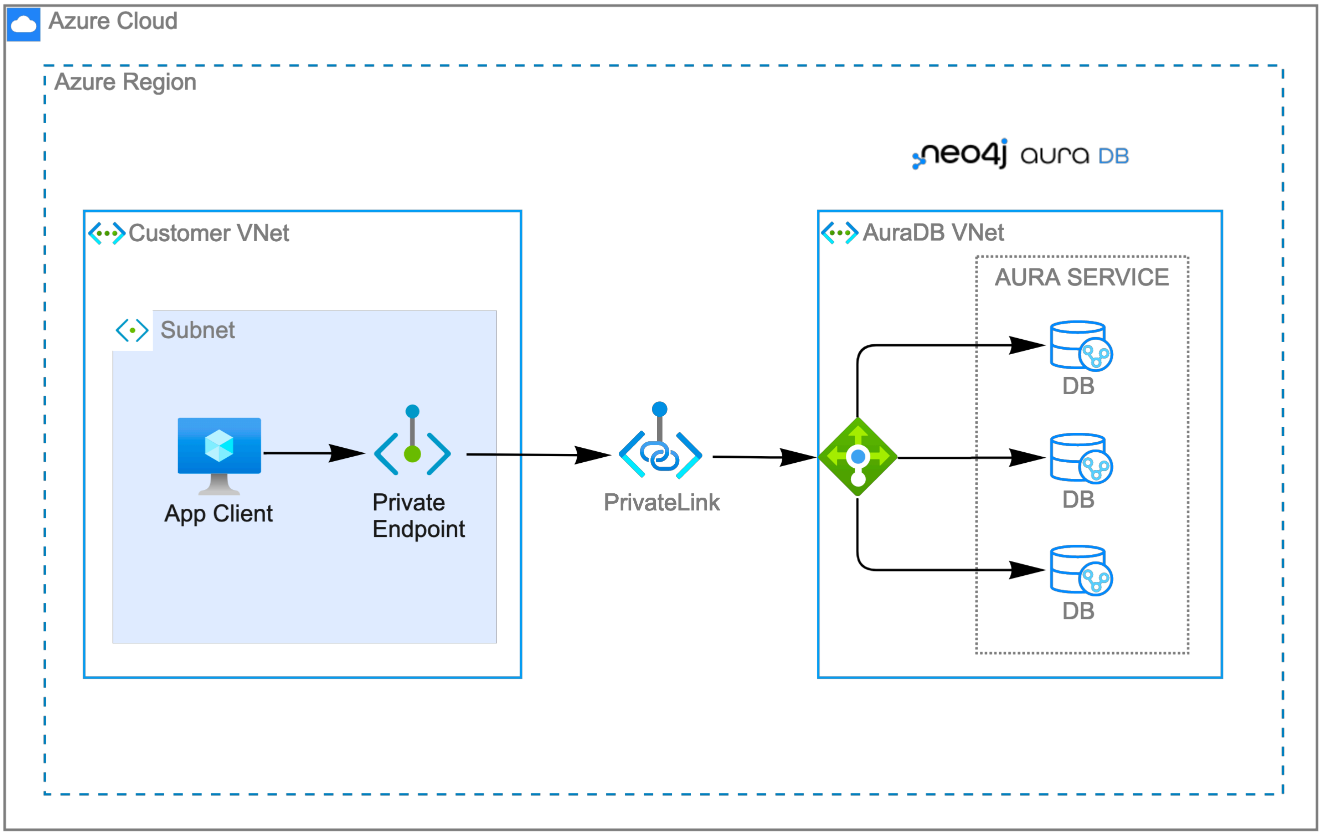

Figure 9. VNet connectivity with Azure Private Link

All applications running Neo4j workloads inside the VNet are routed directly to your isolated environment in Aura without traversing the public internet. You can then disable public traffic, ensuring all traffic to the instance remains private to your VNet.

NOTE

NOTE

- Private Link applies to all instances in the region.
- When activated, a **Private Connection** label, shield icon, and dedicated **Private URI** will appear on any instance tile using Private Link in the Aura Console.
- If you disable public traffic, you must use a dedicated VPN to connect to your instance via Browser or Bloom.
- Connections using private endpoints are one-way. Aura VNets can't initiate connections back to your VNets.

Browser and Bloom access over private endpoints

To connect to your instance via Browser or Bloom, you must use a dedicated VPN. This is because when you disable public access to your instance, this applies to all connections, including those from your computer when using Browser or Bloom.

Without private endpoints, you access Browser and Bloom over the internet:

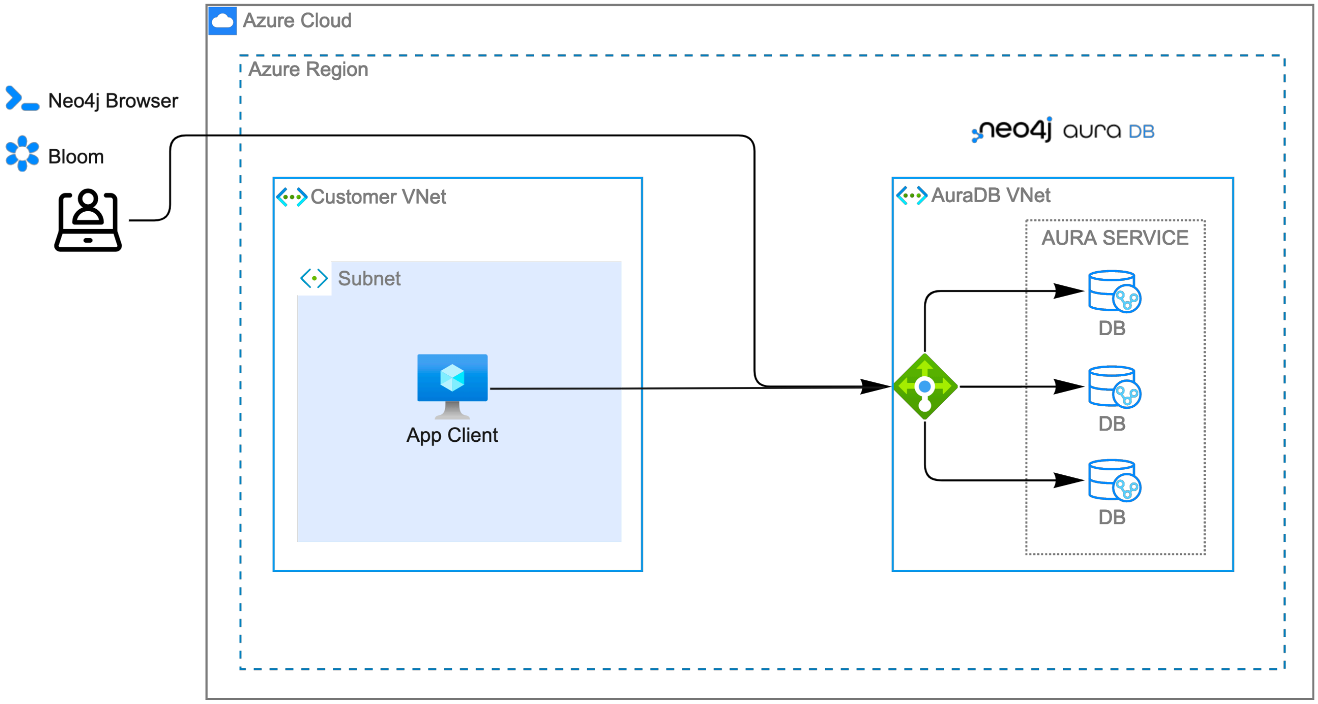

Figure 10. Architecture overview before enabling private endpoints

When you have enabled private endpoints and disabled public internet access, you can no longer connect Browser or Bloom to your instances over the internet:

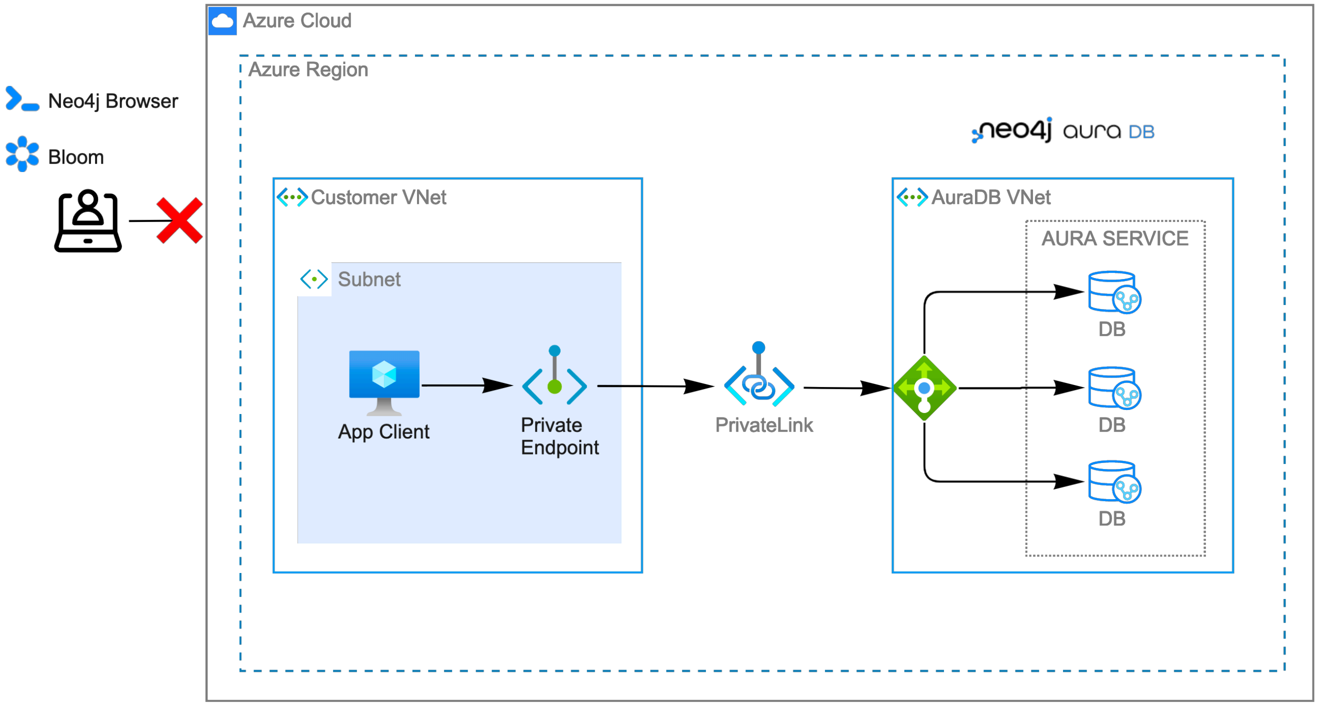

Figure 11. Architecture overview with private endpoints enabled and public traffic disabled

To continue accessing Browser and Bloom, you can configure a VPN (Virtual Private Network) in your VNet and connect to Browser and Bloom over the VPN.

NOTE

To access Bloom and Browser over a VPN, you must ensure that:

- You have setup Azure Private DNS → , or an equivalent DNS service, inside of the VNet.
- You use the **Private URI** shown on the instance tile and in the instance details. It will be different from the **Connection URI** you used before.

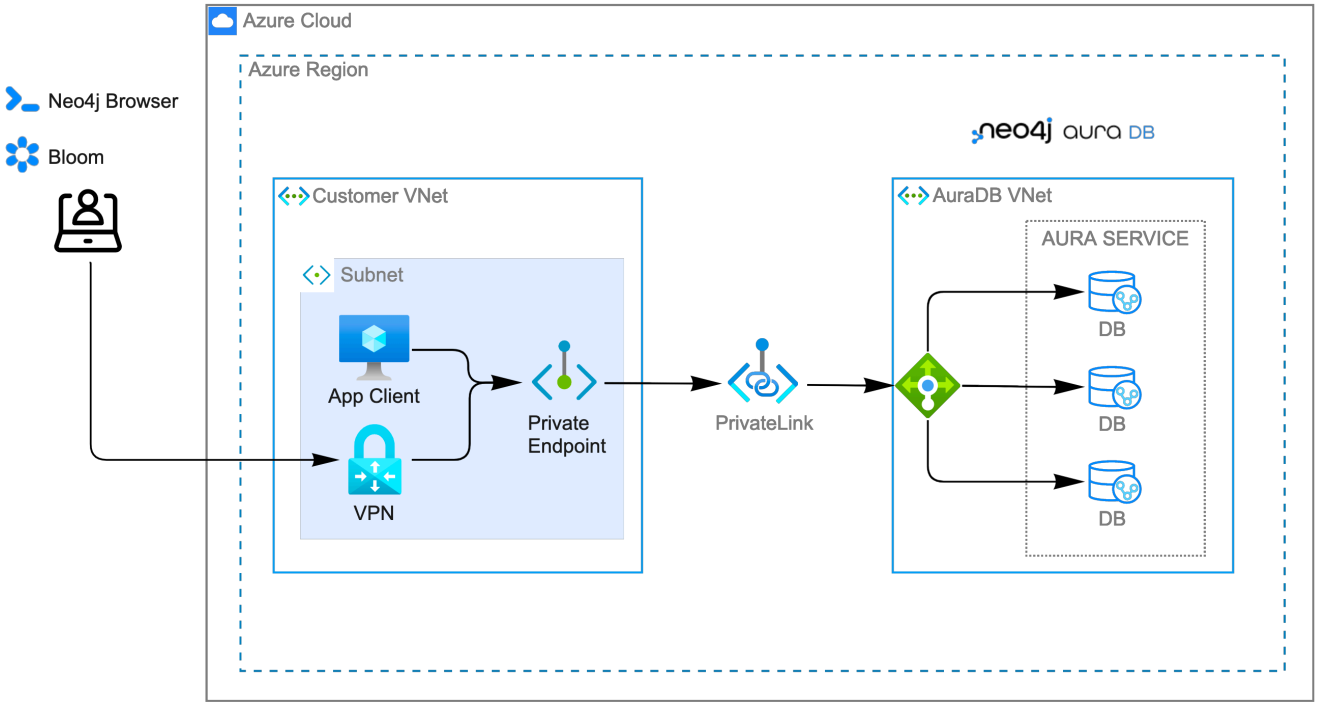

Figure 12. Accessing Browser and Bloom over a VPN

Enable Azure Private Endpoints for Aura

1. To enable private endpoints using Azure Private Link:

   a. From the sidebar menu in the Aura console, select **Security > Network Access > Network Access**

   b. Select **New network access configuration** and follow the setup instructions.

2. Configure Network Access in the Aura console

   a. Select your product from the available options.

   b. Select the appropriate region for your deployment. (Azure Private Link applies to all instances in the region.)

   c. Enter the **Target Azure Subscription IDs**.

   d. Select **Enable Private Link**.

3. Obtain a Private Link service name

   a. After enabling Private Link, you receive a Private Link service name in the Aura console.

   b. Copy this service name, you need it in the next step.

4. Create Private Link endpoint in the Azure portal

    a. Log in to your Azure portal.
    b. Navigate to your cloud VPC and create a new Private Link endpoint.
    c. Use the Private Link service name obtained in step three for the configuration.

5. Accept Endpoint in Aura console

    a. Return to the Aura Console.
    b. Check for the newly created Private Link endpoint.
    c. Accept the endpoint to complete the connection process.
    d. **At this point, it is highly recommended to test connectivity through the private endpoint.**

6. Disable public traffic

    a. Before disabling public traffic, test all your application connectivity with Private Link to ensure everything is functioning correctly.
    b. Once verified, you can disable public traffic by toggling off the public access option.
    c. Note: If needed, you can postpone disabling public traffic.

7. Monitor Private Link status

    a. You can monitor the status of your Private Link configuration in the Aura Console.
    b. Ensure that all services are running as expected and troubleshoot any issues if necessary.

Please see the Azure Documentation → for required roles and permissions.

Private links

AuraDB Virtual Dedicated Cloud

This private link section is cloud-agnostic and therefore applicable to all clouds. A private link provides secure network connectivity between your application and AuraDB without exposing traffic to the public internet.

The term “private link” refers to:

- Private Service Connect = Google Cloud platform
- PrivateLink = AWS
- Private Link = Azure

The following steps explain the process of establishing a private link connection to securely connect your application to an AuraDB Virtual Dedicated Cloud environment.

NOTE

The dbid: abcd1234 and orch-id: 0000 are used in this example. These are different in your AuraDB Virtual Dedicated Cloud environment.

1. The application initializes a driver connection to neo4j+s://abcd1234.production-orch-0000.neo4j.io.
2. The network layer then queries the DNS server to resolve the fully qualified domain name (FQDN) (in this case, abcd1234.production-orch-0000.neo4j.io) to its corresponding IP address.
3. The Cloud Virtual Network private DNS is queried, and it resolves the FQDN to 10.10.10.10, based on the wildcard DNS A record created: *.production-orch-0000.neo4j.io → 10.10.10.10
4. The application's connection is directed to 10.10.10.10, which is the private link endpoint. From there, the private link endpoint forwards the network connection to the private ingress through the private link.
5. The private ingress extracts the dbid from the FQDN and directs the connection to the appropriate Aura instance (dbid: abcd1234).
6. The Aura instance responds by sending the Neo4j cluster routing table back to the application, which contains information about the instances and their roles.
7. Based on the type of transaction (read or write) the driver selects an appropriate instance to execute a read or write transaction. The code has the ability to direct the transaction to the appropriate instances this way.
8. Similar to before, the Cloud Virtual Network private DNS is queried and resolves the FQDN to 10.10.10.10. The application's connection is sent to the private link endpoint (10.10.10.10), which forwards the network connection to the private ingress through the private link. The private ingress then directs the connection to the Aura instance with dbid: abcd1234.
9. Finally, the write transaction is received and executed within the Aura instance with dbid: abcd1234.

Table 1. Available instances and their roles

abcd1234.production-orch-0000.neo4j.io	role: write
abcd1234.production-orch-0000.neo4j.io	role: read
abcd1234.production-orch-0000.neo4j.io	role: read

Custom endpoints with private link

In addition to the production-orch-<orch>.neo4j.io DNS records configured for your private link databases, you must add the following records in order for a Custom Endpoint assigned to a Private Link database to work. When configuring a custom endpoint with a URI like `my-endpoint-abcdef-123456.endpoints.neo4j.io` , you must add the following DNS records for the custom endpoint to function properly:

```
my-endpoint-abcdef-123456.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
a-my-endpoint-abcdef-123456.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
b-my-endpoint-abcdef-123456.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
c-my-endpoint-abcdef-123456.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
d-my-endpoint-abcdef-123456.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
```

Alternative wildcard approach

Instead of adding individual records for a custom endpoint, it is possible to use a wildcard:

```
*.endpoints.neo4j.io IN A <ip-address-of-your-endpoint>
```

This would automatically cover any custom endpoint created for that region. Note that similarly to the individual records, this wildcard record must also be added in addition to the `production-orch-<orch>.neo4j.io` DNS records as mentioned above.

IMPORTANT

If users have regions with different private link endpoints, but have linked those endpoints to one client VPC , then the wildcard record would direct all traffic for custom endpoints to only one region — whichever is associated with the IP address used in the DNS records. This breaks routing for custom endpoints located in the other regions, and therefore, if you do not have a simple private link setup, it is recommended to use the individual custom endpoint records, rather than the wildcard.

Test connectivity through the private endpoint

Use the `nslookup` command to confirm whether the Fully Qualified Domain Names (FQDNs) of your Aura instances are directed to the IP address of the PrivateLink endpoint (usually represented by an internal IP address, such as 10.0.0.0).

```
nslookup <dbid>.production-orch-<orchestra>.neo4j.io
```

Use cURL from a VM instance or a container located in the related VPC network.

```
curl  https://<dbid>.production-orch-<orchestra>.neo4j.io
```

Use nc commands on one of your VM instances or container located in the related GCP Project VPC network, and make sure you get a successful response for all commands

```
nc -vz <dbid>.production-orch-<orchestra>.neo4j.io 443
nc -vz <dbid>.production-orch-<orchestra>.neo4j.io 7687
nc -vz <dbid>.production-orch-<orchestra>.neo4j.io 7474
# if you are using AuraDS
nc -vz <dbid>.production-orch-<orchestra>.neo4j.io 8491
```

On Windows, you can get Netcat → or use PowerShell

```
Test-NetConnection <dbid>.production-orch-<orchestra>.neo4j.io -Port 7687
Test-NetConnection <dbid>.production-orch-<orchestra>.neo4j.io -Port 7474
Test-NetConnection <dbid>.production-orch-<orchestra>.neo4j.io -Port 443
# if you are using AuraDS
Test-NetConnection <dbid>.production-orch-<orchestra>.neo4j.io -Port 8491
```

Supported TLS cipher suites

For additional security, client communications are carried via TLS v1.2 and TLS v1.3.

AuraDB has a restricted list of cipher suites accepted during the TLS handshake, and does not accept all of the available cipher suites. The following list conforms to safety recommendations from IANA, the OpenSSL, and GnuTLS library.

TLS v1.3:

- `TLS_CHACHA20_POLY1305_SHA256 (RFC8446)`
- `TLS_AES_128_GCM_SHA256 (RFC8446)`
- `TLS_AES_256_GCM_SHA384 (RFC8446)`

TLS v1.2:

- `TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (RFC5288)`
- `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (RFC5289)`
- `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (RFC5289)`
- `TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (RFC7905)`
- `TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (RFC5288)`

Prev
‹ Cloud provider marketplaces

Next
Single Sign-On (SSO) ›

Contents

Join the User Research panel

Influence the future of Neo4j products by sharing your experiences with a researcher.

Learn more

LEARN

- Sandbox
- Neo4j Community Site
- Neo4j Developer Blog
- Neo4j Videos
- GraphAcademy
- Neo4j Labs

SOCIAL

- Twitter
- Meetups
- Github
- Stack Overflow

Want to Speak?

CONTACT US →

US: 1-855-636-4532

Sweden +46 171 480 113

UK: +44 20 3868 3223

France: +33 (0) 1 88 46 13 20