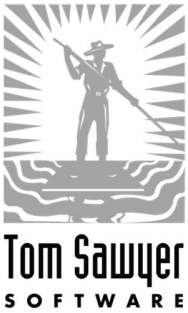


Tom Sawyer Graph Database Browser 8.3.0

AMI Deployment Guide

Introduction

The Graph Database Browser is a web-based data visualization application for graph databases. It is used for easily and quickly viewing and analyzing connections, networks, and dependencies with clean, interactive graph layout. The Graph Database Browser can connect to and view data and schema from popular databases such as Amazon Neptune, Neo4j, Apache TinkerPop, JanusGraph, and Stardog. It is a multiuser system where each user can have his or her own user preferences, view data in a unique way, and connect to and view data from multiple databases.

Customer deployment is supported as a single-server deployment. Since only one EC2 instance is required, it is a very simple deployment. There are no additional options to consider, nor is a CloudFormation template necessary. The deployment of this software can be completed in about 10 minutes for an expert to about an hour for a novice.

The Graph Database Browser is available as an AMI on AWS Marketplace. To get started, you need to subscribe to the Graph Database Browser on AWS Marketplace and launch the AMI into an EC2 instance. For the simplest experience, take the recommended seller settings as they are presented in the launch instance wizard and this guide, particularly in the Sizing section below.

Quick Start Instructions

■ Launching the Server

1. Go to the Graph Database Browser subscription page on AWS Marketplace.
2. Click **Continue to Subscribe** and then click **Continue to Configuration**.
   a. Leave the default fulfillment option.
   b. Select the latest version of the software.
   c. Select the region you want to deploy to.

      It is best to select the same region as the database you want to visualize.

   d. Click **Continue to Launch**.
3. On the next page, from the **Choose Action** menu, select **Launch through EC2** and then click **Launch**.
4. On the **Choose an Instance Type** page:
   a. Scroll down and select **t3.large** and then click **Next: Configure Instance Details**.
   b. Accept all of the default selections except:
      — **Network**: Leave the default VPC setting selection unless you have a Neptune database to connect to, in which case, pick the same VPC that Neptune is in.
      — **Subnet**: Select your subnet. If you do not know the subnet, leave the default selection.
      — **Auto-assign Public IP**: Select **Enable**.
5. When you have the settings you need, click **Next: Add Storage**.

   The default of 100 GiB of General Purpose (gp2) storage should be enough for most uses of the Graph Database Browser. See the Sizing section below if you have a large user base.
6. Click **Next: Add Tags** and then click **Next: Configure Security Group**.

   On the **Configure Security Group** page, the default security group creates a new security group based on Tom Sawyer Software recommendations for the Graph Database Browser.

   The new security group has everything you need to get started. A warning about opening it up to allow all public IP addresses displays. You can lock this security group down now to known IP address ranges, or you can follow the instructions below later to refine your security group.
7. Click **Review and Launch**.
8. On the **Review Instance Launch** page, scroll down and click **Launch**.
   a. For **Key Pair Settings**, select a key pair that you have access to so that you can ssh into the instance for additional configuration and routine system administration tasks. If you do not have any existing key pairs, create a new one.
   b. Click the check box acknowledging that you have access to the key file.
   c. Click **Launch Instances**.

   You should see a message indicating a successful launch!
9. On the **Launch Status** page, click the instance ID to go to your EC2 console.

   You should see the new unnamed EC2 instance with an initializing hourglass in the Status Checks column. Wait a few minutes until the icon changes to a green circle with a checkmark and the two initialization checks have passed.
10. Give the instance a name, such as Tom Sawyer Graph Database Browser, so you can easily identify it later in your list of EC2 instances.

■ Using the Application

1. To access the application, open up a web browser and go to http://{instance-url}/databasebrowser.

   Substitute your server name for {instance-url} in the URL, either the Public DNS name or Public IP of the instance you just created.

2. To sign in for the first time, use these credentials:

   Username: admin

   Password: {instance-id}

   The instance-id is available in the Description tab in your EC2 console.

3. When prompted, enter your e-mail address for your username and change your password.

4. In the Account Information screen, update the First Name and Last Name fields with your information.

5. Sign out and sign back in with your new credentials.

6. For information on how to add a database and use the application, click the help icon.

■ Allowing Multiple Users

By default, only one user is able to use the Graph Database Browser. To allow multiple users to create their own accounts on your instance, you must enable self-registration. There is no extra charge in AWS for this as it is all local to your instance.

- To enable user self-registration, run the script `/home/ec2-user/enable-user-registration.sh`. This adds a Sign Up link on the sign-in page.
- Later on, if you want to restrict user registration, run the script `/home/ec2-user/disable-user-registration.sh`. This removes the Sign Up link and only registered users can sign in.

The user account information stays on the instance itself, encrypted in a local database and is not transmitted anywhere else. User accounts are required to keep user preferences, as this is a multiuser platform for all of your graph database users. Once logged in, users can click the help icon for help using the application.

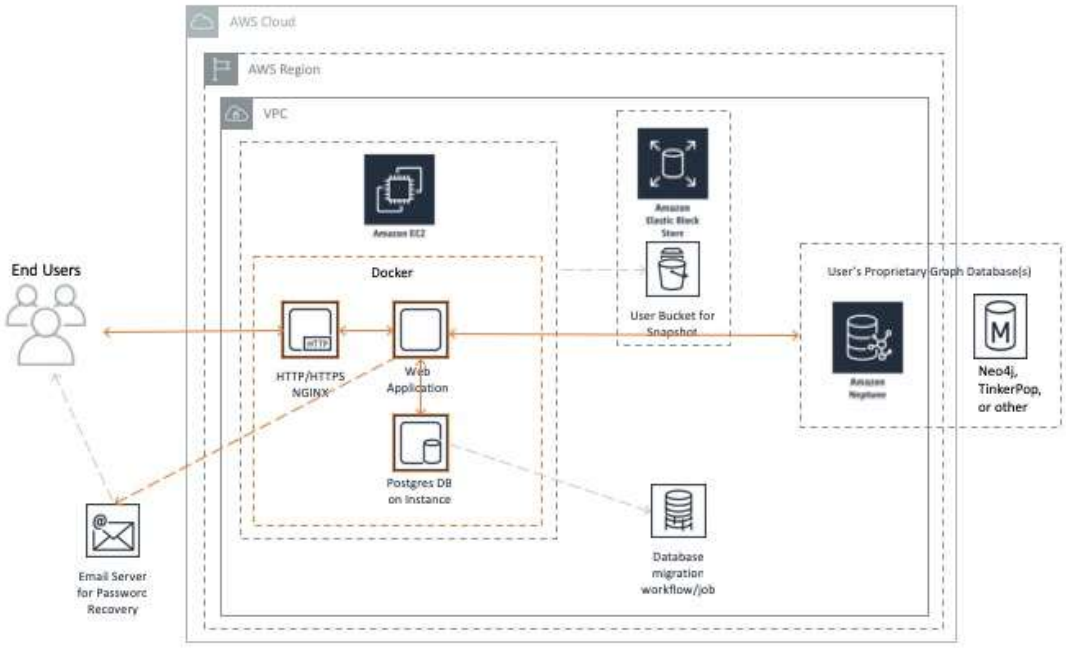
Prerequisites and Requirements

To successfully deploy and maintain the Graph Database Browser AMI, the system administrator needs a basic understanding of the security and networking concepts of launching an EC2 instance that is accessible to the users' graph databases and S3 buckets. This requires knowledge of security groups and VPCs in AWS.

To configure HTTPS using your own certificates, and password recovery through your own mail server, the system administrator needs a basic knowledge of Linux configuration and running commands in order to follow the instructions provided in this guide.

Architecture Diagrams

The following diagram shows the big picture of the Graph Database Browser as launched in an EC2 instance. It includes suggestions for backup, the configuration of an external mail server for password recovery, and the interaction of the system with the users' proprietary graph databases. The components outlined in orange are managed by the application, and the components and relationships depicted in grayscale must be managed by you, the system administrator.



Tom Sawyer Graph Database Browser is a multi-Docker container application that runs with a Docker compose file:

`/home/ec2-user/gdbb-AMI-docker-compose/docker-compose.yml`

It is composed of the following containers:

- Nginx

  As the web server, it handles HTTPS certificates and virtual hosts.

  `/home/ec2-user/gdbb-AMI-docker-compose/lic-docker-gen/ssl`

- jwilder/docker-gen

  The file generation handler for nginx config files.

- Postgres

  Stores Tom Sawyer Graph Database Browser AMI data; users, connections, and appearance rules.

  `/home/ec2-user/gdbb-AMI-docker-compose/.postgres-data`

- Tom Sawyer Licensing AMI Docker containers
  - ts-lic-derby-server

    Stores Tom Sawyer Licensing AMI data.

    `/home/ec2-user/gdbb-AMI-docker-compose/.data`

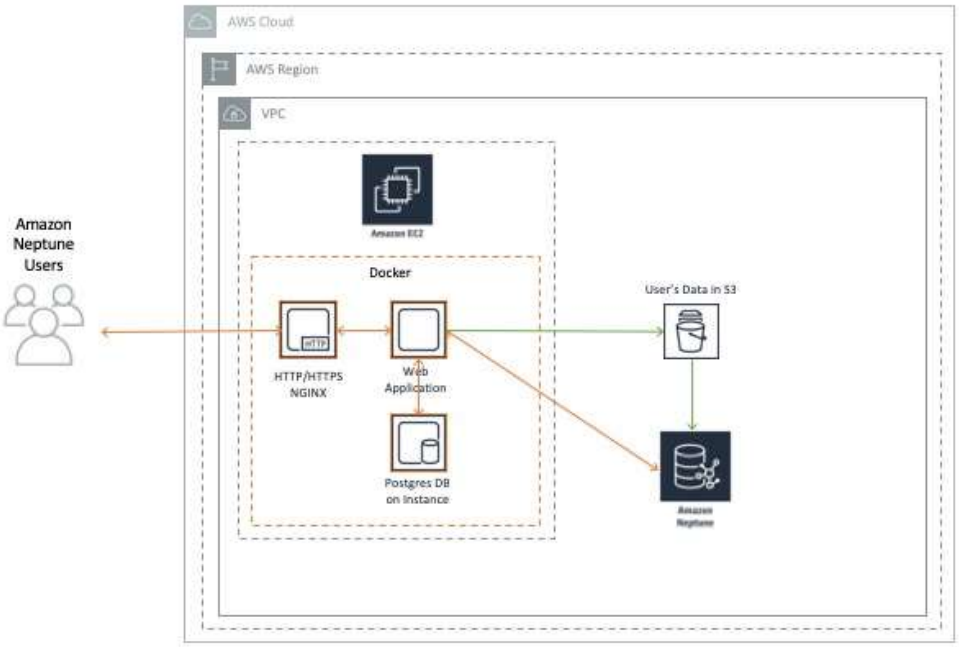
  - ts-lic-db-service
  - ts-lic-administration

- Tom Sawyer Graph Database Browser AMI Docker container
  - ts-dbbrowser-webapp

The Nginx container acts as a web server and redirects traffic to the Tom Sawyer Graph Database Browser AMI Docker container. The Tom Sawyer Graph Database Browser AMI consumes from the Postgres and Tom Sawyer Licensing AMI administration containers.

■ For Amazon Neptune Users

The following diagram shows a feature of the Graph Database Browser specific to Neptune—loading data from an S3 bucket directly to a Neptune database without executing queries. This can be done via the user interface of the Graph Database Browser. A prerequisite for this feature is having the Neptune database and the Graph Database Browser instance reside in the same VPC.



In versions 8.2.1 and 8.2.2, the Neptune instance must have IAM disabled in order to connect. In 8.3.0, the Graph Database Browser supports connection to Neptune with IAM credentials. The necessary credentials are AWS Access Key ID, AWS Secret Access Key, and Services Region.

Planning Guidance

■ Security

Access Management

Access to your Graph Database Browser resources should be managed through AWS Identity and Access Management (IAM) policies. While setting up security roles in your organization through IAM is always recommended, it is not required for a simple deployment of the Graph Database Browser. You can get started quickly without IAM configured, and then at a later time, improve your security policies to match the level of security required by the databases your users are accessing.

The IAM security policies configured for the single EC2 node should be compliant with the level of security appropriate for the data being accessed in the users' proprietary databases. You can create IAM roles for users and then configure the roles to provide access rights to your AWS computing resources based on the needs of your user base. For more information, see the AWS IAM Getting Set Up documentation.

Configuring Rules

At a minimum, port 80 must be open and accessible to the user for application use. You can set this in your security group configuration.

If you want a high level of security, the EC2 instance should be locked down appropriately with rules for port access, along with ingress and egress rules.

The recommended security group settings are:

1. Port 22 for SSH access by the administrator. Inbound and outbound IP addresses should be restricted to the ranges from which the administrator can access the instance. SSH keys are configured at EC2 launch time through the launch instance wizard.
2. Port 80 for HTTP access to the web application. If the data is not proprietary, inbound and outbound IP addresses can be left wide open using the default. If the data is highly proprietary and the access needs to be restricted, our recommendation is to restrict access to known IP address ranges.
3. Port 443 for HTTPS access to the web application. If the data is not proprietary, inbound and outbound IP addresses can be left wide open using the default. If the data is highly proprietary and the access needs to be restricted, our recommendation is to restrict access to known IP address ranges.

   In order to have HTTPS access with your own certificates, some configuration must be done. To add your SSL certificates:

   a. Name your nginx certificate `.crt` and `.key` files as `default.crt` and `default.key` and place them in the directory `/home/ec2-user/gdbb-AMI-docker-compose/lic-docker-gen/ssl`.
   b. Restart the Graph Database Browser by restarting the instance or running the update script `/home/ec2-user/tsgddb.sh`.

User Accounts in the Graph Database Browser

Once the EC2 is launched and a user connects his or her browser to the web application's URL, the new user is prompted to create an account on the server. This allows for a multiple user environment, where each user can have his or her own user preferences for the Graph Database Browser stored. The user credentials for each user are stored encrypted in a Postgres database local to the EC2 instance and managed therein. Password recovery via e-mail can be configured by following the steps in the next section, Password Recovery. No data whatsoever is transmitted outside of this application or server, including user credentials or user preferences.

Password Recovery

To enable password recovery, you need to know the SMTP settings of your own mail server. You also need to obtain `javax.mail.jar`, a library necessary for enabling e-mail.

To configure the mail server for password recovery:

1. Download the file `javax.mail.jar` from javaee.github.io/javamail/ - Download_JavaMail_Release.
2. Add `javax.mail.jar` to the directory:
   `/home/ec2-user/gdbb-AMI-docker-compose/libraries`
3. In the same directory, edit the `spring.mail` properties in the file `javax.mail.properties` with information for your mail server.
4. Restart the instance or run the update script `/home/ec2-user/tsgddb.sh`.

Graph Database Integration

Since this application allows users to connect to externally hosted graph databases, each graph database's networking security must also be configured to be accessible by this instance. When connecting to Amazon Neptune, both Neptune and the Graph Database Browser instance must reside in the same VPC.

Password Policy

Follow best practices for password and key rotation by following the policies of databases being accessed with the Graph Database Browser.

■ Costs

The cost of running Tom Sawyer Graph Database Browser as a single EC2 node deployment—the only supported configuration—can be easily calculated with the published rates on AWS Marketplace. At time of this writing, the cost for the smallest recommended size—t3.large for up to 5 concurrent users—is $0.53 per hour or $382 per month. There will be some additional EBS costs for up to about $10 per month. If you are backing up data, depending on how you store backups, there will be an additional cost there as well. For more detail, see the AWS EBS pricing page.

■ Sizing

You select the EC2 instance size and type when launching an EC2 from your AMI purchase on AWS Marketplace. For less than five concurrent users, we recommend selecting a t3.large. For more than 20 users, we recommend that you launch another instance dedicated to a separate group of users. While the current instances are for single-node deployment only, in the future, we plan to provide load-balanced instances for greater scalability and reliability.

This table provides guidelines when choosing EC2 instance size and configuring storage for your EBS.

Concurrent Users	Instance Type	EBS Volume
0-5	t3.large, m5.large	General Purpose SSD 100 GiB
6-10	t3.xlarge, m5.xlarge	General Purpose SSD 100 GiB
11-20	t3.2xlarge, m5.2xlarge	General Purpose SSD 100 GiB
Over 20	Additional instance(s), and/or m5.4xlarge, m5.12xlarge	General Purpose SSD 100 GiB or higher

Your mileage may vary due to the usage of the tool. If many users are loading high volumes of data into the applications, or the server becomes sluggish or unresponsive, we recommend that you increase the instance size in memory and the number of processors.

Deployment Guidance

■ Deployment Assets

In order to maximize uptime and reliability for this single-node deployment configuration, monitoring and alarms should be set on the instance to alert the system administrator, or to alert a script to restart services or reboot the instance if necessary. At this time, we do not support auto-scaling or multi-availability zone configurations.

These are the most common issues faced by our users:

- Security groups are not set up correctly. Make sure port 80 is reachable for HTTP access.

- In the initial version of the Graph Database Browser AMI in overseas deployments, very long hostnames resulted in a 503 error. Since this can be difficult to troubleshoot, it is a good idea to create a custom hostname instead of the auto-generated hostname to test the deployment. For more information, see the Usage Instructions.
- Public DNS names fail for region ap-southeast-1 because the region is appended to the hostname and it becomes too long. However, in version 8.2.2 and above, the hostname can be up to 128 characters.
- Intermittent connection errors occur when end users access the application and begin connecting to their own databases. When this occurs, users may experience timeout exceptions. To resolve this problem, ensure that the database is accessible to the Graph Database Browser, with appropriate network, port, and protocol access. It is best if the user database resides in the same VPC as the Graph Database Browser EC2 instance, but if not, it must be accessible to it.

Operational Guidance

Health Check

We recommend the following health checks for monitoring purposes with Amazon CloudWatch:

1. Availability zone fault
   - Set a CloudWatch alarm.
   - Restore from backup to the same availability zone when it recovers, or to a new availability zone. VPC and security groups need to be configured to work with the users' databases.
2. Instance fault
   - Set a CloudWatch alarm and respond with an automated reboot of the instance or a restore from the latest snapshot.
3. Application fault
   - Respond to application-level alarms with an automated script to automatically restart services on the instance.
   - If the sign-in page is unreachable, it should be restarted. The sign-in page is http://{instance-url}/databasebrowser.
4. Storage capacity on the application instance internal database
   - Monitor the disk space available on the instance. If it reaches a certain threshold, increase the instance size of the EC2.
5. Storage capacity alarm for EBS
   - Monitor the memory available for snapshots. If it reaches a certain threshold, increase the EBS memory.
6. Security certificate expiration
   - Set reminders to renew certificates in ample time before they expire.

Backup and Recovery

In case of a failure in a region or availability zone, or a failure on the instance itself, the user data to recover and restore is in the Postgres database that resides on the instance. Use this location to back up the data:

```
/home/ec2-user/gdbb-AMI-docker-compose/.postgres-data
```

The data contained in the Postgres database is for user preferences and settings essential to the user experience. It stores account information, configured connections to their databases, and can contain detailed preferences on how to view proprietary data.

Backup

At a minimum, we recommend that you back up the data in Postgres once a day, but always follow your own company's policy. One easy option for data backup in AWS is to store your backups on S3. Another way to back up would be to make a snapshot of your EBS volume.

When setting up the deployment of the Graph Database Browser, if any custom configurations were made to the instance, it would be good to create your own image from this. This can be done via snapshotting the EBS volume or saving to an AMI.

If your application is mission critical and cannot wait for the region or availability zone to be available again, back up and store your data in at least one more region so you can easily and quickly recover from this type of failure.

Recovery

To recover the instance after a destructive failure:

- If custom configurations were made to the instance, you can restore the instance from your own custom AMI. See the AWS instructions on how to create this.
- If no custom configurations were made, all you need to do is launch a new AMI of the same version from AWS Marketplace.
- Use the latest backup copy of the data—snapshot or Postgres backup—to restore to the file location listed above.
- If there was a region or availability zone failure and you cannot wait for it to become available again, you must set up in another region.

■ Routine Maintenance

Certificate management and key rotation should be followed in accordance with your security policy's best practices. It should be as stringent as necessary to protect your graph database data, as this application has access to any databases configured as connections. We suggest a 90-day key rotation policy to protect access to your sensitive data.

Use these locations to place certificates and keys:

- For web server configuration:

  Nginx Docker image, configuration template: `/lic-docker-gen/nginx.tmpl`

- For key rotation, SSL certificates:

  `/home/ec2-user/gdbb-AMI-docker-compose/lic-docker-gen/ssl/default.crt`

  `/home/ec2-user/gdbb-AMI-docker-compose/lic-docker-gen/ssl/default.key`

To assist you when you upgrade to another version of the Graph Database Browser, migration instructions are provided at the time of purchase in AWS Marketplace. The upgraded version includes all of the latest software including support for the latest operating system and patches.

During an upgrade, a new instance is created through the AMI launch template. Then, you need to shut down the old instance, export the application data, and import the data into the new instance.

Use this storage location for migration of application data, which is the same as the backup and recovery location:

`/home/ec2-user/gdbb-AMI-docker-compose/.postgres-data`

Emergency Maintenance

Due to the single-node deployment scenario, you should follow the backup and recovery procedures for any failure scenarios.

For when the web application is unreachable:

1. Make sure all the Docker containers are running:
   `/home/ec2-user/gdbb-AMI-docker-compose/docker-compose ps`
2. Restart all the Docker containers:
   `/home/ec2-user/gdbb-AMI-docker-compose/docker-compose up -d`

If the steps above do not bring the web application back online, try these backup and recovery procedures:

1. Migrate the application data to a new instance:
   `/home/ec2-user/gdbb-AMI-docker-compose/docker-compose stop`
2. Compress and export this directory to a new instance:
   `/home/ec2-user/gdbb-AMI-docker-compose/.postgres-data`

Support

Support is included when you purchase the Tom Sawyer Graph Database Browser AMI. Users must sign up with Tom Sawyer in order to submit a support request at www.tomsawyer.com/aws-marketplace-support.

Support Costs

Support for the Graph Database Browser is free of charge for AWS Marketplace customers.

For help with custom projects, contact sales@tomsawyer.com for an individualized consultation.