Tom Sawyer Graph Database Browser

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 and view data and schema from popular databases such as Amazon Neptune, Neo4j, and Apache TinkerPop. It is a multi-user system where each user can have his or her own user preferences, view data in a unique way, and connect and view data from multiple databases.

Customer deployment is supported as a single server deployment. It requires only one EC2 instance and is, therefore, a very simple deployment. There are no additional options to consider, nor is there 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 in AWS. To get started using the Graph Database Browser, you must subscribe on the AWS Marketplace page and launch the AMI into an EC2 instance, using a wizard to guide you through the process. For the simplest experience, take the recommended seller settings as they are presented in the wizard and this guide, particularly in the Sizing section below.

Quick Start Instructions

Launching the Server

1. Visit the Graph Database Browser subscription site on the AWS Marketplace.
2. Click **Continue to Subscribe**.
3. Click **Continue to Configuration**.
   a. The Fulfillment option is filled in for you.
   b. Select the latest version of the software.
   c. Select the region you wish to deploy to.

      Note that it is best to select the same region as the database you wish to visualize.

   d. Then, click **Continue to Launch**.
4. On the next page, select **Choose action: Launch through EC2**, and click **Launch**.
5. On the **Step 2: Choose an Instance Type** page:
   a. Scroll down the list 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**.

6. When you have the settings you desire, click **Next: Add Storage**.
   a. 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.

   b. Click **Next: Add Tags**, and then click **Next: Configure Security Group**.
7. On the **Configure Security Group** page, the default Security Group creates a new security group, based on Tom Sawyer recommendations for the Graph Database Browser.

   It has everything you need to get started. You will see a warning about opening it up to allow all public IP addresses. 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.
8. Click **Review and Launch**.
   a. On the **Review Instance Launch** page, scroll down and click **Launch**.
   b. 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.
   c. Click the checkbox acknowledging that you have access to the key.
   d. Click **Launch Instances**.

   You should see a message indicating a successful launch!
9. Click on the instance ID to take you to your EC2 Console. You should see the new, unnamed EC2 Instance that shows that it is initializing with an hourglass icon next to it. Wait a few minutes until it shows that it is running and has passed the two initialization checks. 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 given above, either the Public DNS name or the Public IP of the instance you just created.
2. To start using the software, create a user account on the login page.

   The 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 multi-user platform for all of your graph database users. Once logged in, users can click the help icon (?) to for help using the application.

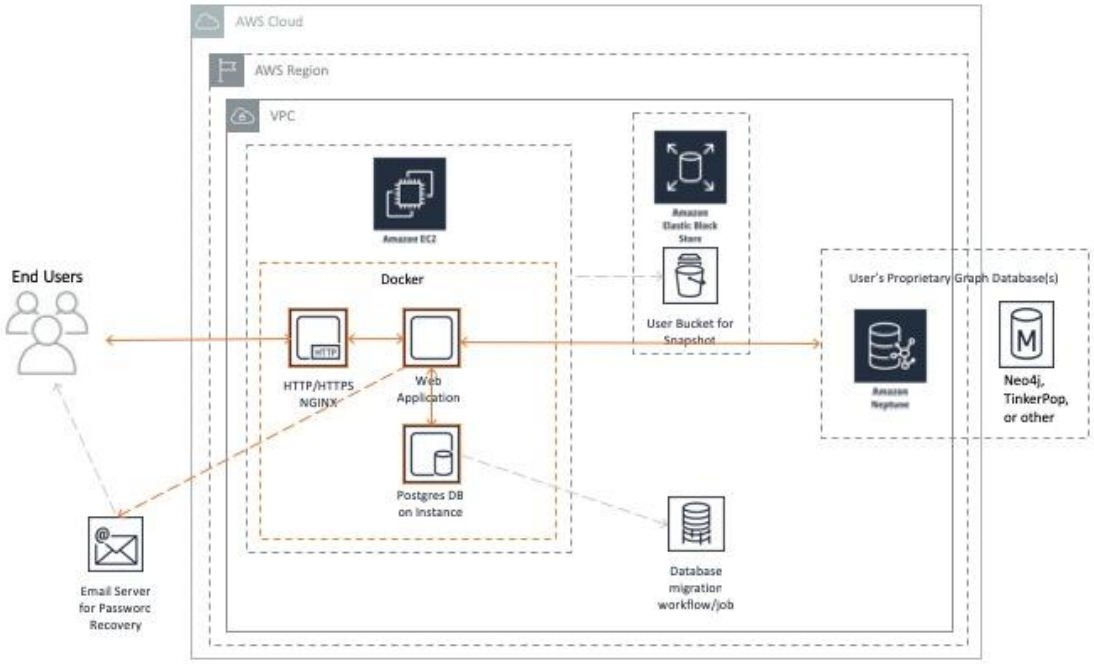
Prerequisites and Requirements

To successfully deploy and maintain the Tom Sawyer Graph Database Browser AMI, the system administrator should have some basic understanding of the security and networking concepts of launching and EC2 instance that will be accessible to the user’s graph database(s) and S3 buckets. This requires knowledge of Security Groups and VPCs in AWS.

To configure https using own certificates, and password recovery through own mail server, the system administrator should have basic knowledge in Linux configuration and running commands in order to follow the simple Usage Instructions provided.

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 email server for password recovery, and the interaction of the system with the user's proprietary graph database(s). Note that 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 web server, handles HTPPS certificates and virtual hosts.

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

- jwilder/docker-gen

  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

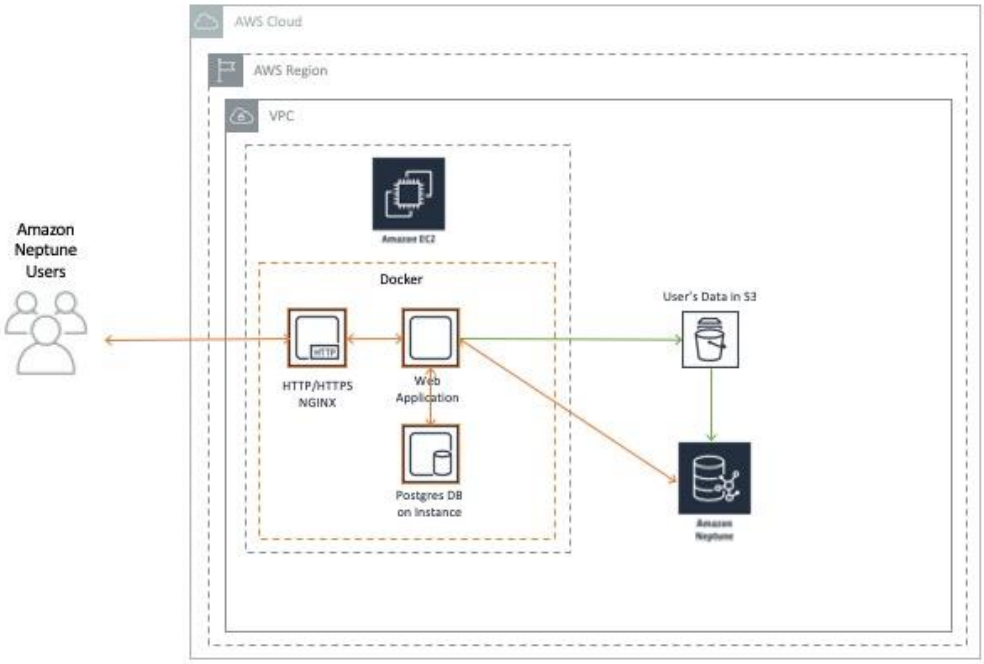
    `/home/ec2-user/gdbb-AMI-docker-compose/.data`

  - 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 Tom Sawyer Licensing AMI administration container and Postgres container.

■ For Amazon Neptune Users

The following diagram shows a feature of the Graph Database Browser specific to Neptune, which is 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. Note that it necessary for the Graph Database Browser and the Neptune database to be 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, Graph Database Browser will begin supporting connection to Neptune with IAM credentials.



Planning Guidance

■ Security

Access Management

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

The Identity and Access Management (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 user's proprietary database. IAM roles can be created for users, roles, and can be configured very specifically to provide access rights to your computing resources in AWS based on the needs of your user base. More information can be found in AWS IAM documentation. Refer to this guide to get you started.

Configuring Rules

To use the application, at a minimum, port 80 must be open and accessible to the user. This can be set up in your Security Group configuration.

If a high level of security is desired, 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 AWS launch 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 certificate, some configuration must be done. See the Usage Instructions for how to configure your SSL certificate.

User Accounts within Graph Database Browser

Once the EC2 is launched, and a user connects their 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 their own user preferences for the Graph Database Browser to be 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 email can be configured by following step 4 in the Usage Instructions to specify a mail server. No data whatsoever is transmitted outside of this application or server, including user credentials or user preferences.

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

Best practices for password and key rotation should be followed in accordance with the policies for the database(s) being accessed by 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 the AWS Marketplace. At time of this writing, the costs 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/month. If you are backing up data, depending on how you store backups, there will be an additional cost there as well. See the AWS EBS pricing page for more detail.

■ Sizing

The EC2 instance size and type must be selected during the process of launching an EC2 from your AMI purchase on the Marketplace. For less than five concurrent users, we recommend selecting a t3.large. For more than 20 users, we recommended that you launch another instance dedicated to a separate group of users. In the future, we aim to provide load-balanced instances for greater scalability and reliability, but the current version is a single node deployment only.

The following table shows guidelines when choosing EC2 instance size and configuring storage for your EBS. 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.

# 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

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-AZ configurations.

The most common issues faced by our users:

1. Security Groups are not set up correctly. Make sure port 80 is reachable for http access.
2. Very long hostnames have been an issue in overseas deployments in initial version of the Graph Database Browser AMI. This results in a 503 Error and 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. See Usage Instructions for more information.

   Public DNS names work fine for region us-east-1, but fail for region ap-southeast-1 because the region is appended to the hostname, and it becomes too long. In version 8.2.2 and above, however, the hostname can be up to 128 characters.

3. Sometimes there are connection errors 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 recommended the following health checks for monitoring purposes with Amazon CloudWatch:

1. Availability zone fault
   - Set a CloudWatch alarm.
   - Restore from backup to the same AZ when it recovers, or to a new AZ. VPC and Security Groups will need to be configured to work with the user's databases.
2. Instance fault
   - Set a CloudWatch alarm and respond with an automated reboot of the instance or a restore from latest snapshot.
3. Application fault
   - Respond to application-level alarms with an automated script to automatically restart services on the instance.
   - If the login page is unreachable, it should be restarted.

     The login page is: http://{instance-url}/databasebrowser.
4. Storage capacity on the application instance internal database
   - Monitor the disk space available on the instance. If 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 an availability zone, region, 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 database(s), and can contain detailed preferences on how to view proprietary data.

Backup

We recommend that you back up the data in Postgres once a day at the minimum, but always follow your own company's policy. One easy option for data backup within 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 as detailed in the Usage Instructions, it would be good to create your own image from this. It 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 AZ 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 as detailed in the Usage Instructions, the instance could be restored from your own custom AMI. See AWS instructions on how to create this.
- If no custom configurations were made, this type of restore is not necessary. Launching a new AMI from the Marketplace of the same version would be sufficient.
- The latest backup copy of the data—snapshot or Postgres backup—would be used to restore to the file location listed above.
- If there was a region failure or an AZ failure, and your application is mission critical and cannot wait for the region or AZ to be 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 and as stringent as you would protect your Graph Database data, as this application would have 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:

- Web Server Configuration:

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

- 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`

Upon upgrade to another version of the Graph Database Browser, migration Instructions are provided at the time of purchase in the Marketplace to assist the user. The upgraded version includes all of the latest software including the support for the latest OS and patches. During an upgrade, a new instance is created through the AMI Launch template. Then, you will 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.

Follow these procedures 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 docker containers:
   ```
   /home/ec2-user/gdbb-AMI-docker-compose/docker-compose up -d
   ```

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

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