







 **m-mohamedin** Fixed Receiving Malformed URI error on startup (docker-drawio-125) 

59cb704 · 10 months ago  History

This branch is **119 commits ahead of, 39 commits behind** `fjudith/docker-draw.io:master` .

Name	Name	Last commit date
 ..		
 fonts	Added self-contained standalone docker-compose file	5 years ago
 README.md	Fixed Receiving Malformed URI error on startup (docker-drawio...	10 months ago
 docker-compose.yml	Adding DRAWIO_SERVER_URL new variable	last year

README.md 

This docker compose file runs draw.io diagram editor without depending on any draw.io online services (e.g., export service, plantUml, ...) and with support of Google Drive, Microsoft OneDrive, ...

### Adding fonts to improve generated PDFs and images

The docker-compose file bind the `fonts` volume into the running container system fonts.

The best option for Windows users is to copy the contents of `windowsdrive:/Windows/Fonts` into `fonts` folder. These fonts are copyrighted and cannot be re-distributed freely.

## Configuration

You can customize the application by setting the following environment variables.

- `DRAWIO_SERVER_URL` : Your deployment base URL. For example, `https://drawio.example.com/` or `https://www.example.com/drawio/` if it is deployed into a folder. **Note:** Must end with `/`
- `DRAWIO_BASE_URL` : Your deployment base URL but used with the viewer, lightbox and embed. Usually the same as `DRAWIO_SERVER_URL` but does **not** end with `/` .
- `DRAWIO_CSP_HEADER` : (Optional) Your website Content-Security-Policy if you want to customize it.
- `DRAWIO_VIEWER_URL` : (Optional) If you want to host a draw.io viewer also, set the viewer URL. For example, `https://drawio.example.com/js/viewer.min.js`
- `DRAWIO_LIGHTBOX_URL` : (Optional) If you want to host a draw.io viewer also, set the lightbox URL. For example, `https://drawio.example.com`
- `DRAWIO_CONFIG` : (Optional) draw.io configuration JSON. [Documentation](#)
- `DRAWIO_USE_HTTP` : (Optional and INSECURE) If your setup uses http only and you understand the risks (for example, sending OAuth tokens over http), set `DRAWIO_USE_HTTP=1` . **Caution: Use at your own risk.**

## Google Drive

You will need to create a project at [Google API Console](#) and create [Credentials](#) of type "Create OAuth client ID" -> Web Application. This option will be disabled until you create "OAuth consent screen" from the link in warning message bar. In the "OAuth consent screen" configuration, enter the "Application name" and "Authorized domains". In the "Create OAuth client ID" configuration, enter the required information and note that "Authorized redirect URIs" is `[your-draw.io-hostname]/google` . For example, if you host draw.io at `https://drawio.example.com` , then "Authorized redirect URIs" whould be `https://drawio.example.com/google` and "Authorized JavaScript origins" would be `https://drawio.example.com` .

Set the values of the generated "Client ID" and "Client secret" into environment variables `DRAWIO_GOOGLE_CLIENT_ID` and `DRAWIO_GOOGLE_CLIENT_SECRET` . Also, set `DRAWIO_GOOGLE_APP_ID` environment variable to your Google App ID. APP ID is the number before the dash in the CLIENT ID. For example, if CLIENT ID is `123456789-abc...` , then APP ID is `123456789`

If you want to host a draw.io viewer also, you can create another client ID for the viewer. Viewer has read-only access to Drive files.

Set the values of the generated "Client ID" and "Client secret" into environment variables `DRAWIO_GOOGLE_VIEWER_CLIENT_ID` and `DRAWIO_GOOGLE_VIEWER_CLIENT_SECRET` . Also, set `DRAWIO_GOOGLE_VIEWER_APP_ID` environment variable to your Google App ID.

## Microsoft OneDrive

You will need to create and application in order to use MS Graph APIs. Follow the information on [how to register your app](#) and [how to use the APIs](#).

Once you registered your application, from Microsoft Azure UI, select your new app, then "Authentication". From Authentication, enter your redirect URIs. draw.io requires two redirect URIs `[your-draw.io-hostname]/microsoft` and `[your-draw.io-hostname]/onedrive3.html` . For example, if you host draw.io at `https://drawio.example.com` , then redirect URIs would be `https://drawio.example.com/microsoft` and `https://drawio.example.com/onedrive3.html`

In "Advanced settings" on the same page, enable "Access tokens" and "ID tokens" check boxes. To get the "Client secret", select "Certificates & secrets" page from the menu, then click "+ New client secret" button. Finally, from the "Overview" page in the menu, you can find the "Application (client) ID".

Set the client ID and secret into environment variables `DRAWIO_MSGRAPH_CLIENT_ID` and `DRAWIO_MSGRAPH_CLIENT_SECRET` . If your Azure app is a single tenant, set `DRAWIO_MSGRAPH_TENANT_ID` to your tenant ID.

## Gitlab

Create a new OAuth app (Settings -> Applications). Set "Redirect URI" (e.g, `https://drawio.example.com/gitlab` ) and "Scopes" (e.g, `api` , `read_repository` , `write_repository` ). Then, set the following environment variables, using information from the app, to enable Gitlab integration.

- `DRAWIO_GITLAB_ID` : Your Gitlab ID
- `DRAWIO_GITLAB_SECRET` : Your Gitlab Secret
- `DRAWIO_GITLAB_URL` : Your Gitlab URL, for example, `https://gitlab.com/oauth/token` when the gitlab.com is used

## EMF Converter

This service is currently used by VSDX importer for converting EMF files in VSDX files. If you don't plan to use VSDX importer or your VSDX files don't contain EMF files, then this service is not important to you.

This service is based on [Cloud Convert](#). You will need to register for an account and set the environment variable `DRAWIO_CLOUD_CONVERT_APIKEY` to the API KEY. We use API **V1** API KEY.

## AWS Deployment

You can deploy this docker compose easily to AWS ECS. Follow the instructions in this [tutorial](#) to install Amazonn ECS CLI, create a cluster, and deploy "self-contained" docker compose file to it. We recommend EC2 deployment. You will need to chnage port mapping to 80 and 443 to support standard HTTP and HTTPS ports in `docker-compose.yml` . Don't forget to allow access to these ports in the security group inbound rules. Also, it is required to set `DRAWIO_BASE_URL` environment variable in order to have a fully functional deployment. Set the other environment variables as described above to enable other services and features as needed.

Refer to the main [README](#) file for how to configure **Let's Encrypt**.