Set up your development environment

This folder contains useful scripts and configuration so you can:

- Configure data sources in Grafana for development.
- Configure dashboards for development and test scenarios.
- Create docker-compose file with databases and fake data.

Install Docker

Grafana uses Docker to make the task of setting up databases a little easier. If you do not have it already, make sure you install Docker before proceeding to the next step.

Developer dashboards and data sources

```
./setup.sh
```

After restarting the Grafana server, there should be a number of data sources named `gdev-<type>` provisioned as well as a dashboard folder named `gdev dashboards`. This folder contains dashboard and panel features tests dashboards.

Please update these dashboards or make new ones as new panels and dashboards features are developed or new bugs are found. The dashboards are located in the `devenv/dev-dashboards` folder.

docker-compose with databases

This command creates a docker-compose file with specified databases configured and ready to run. Each database has a prepared image with some fake data ready to use. For available databases, see `docker/blocks` directory. Notice that for some databases there are multiple images with different versions. Some blocks such as `slow_proxy_mac` or `apache_proxy_mac` are specifically for Macs.

```
make devenv sources=influxdb,prometheus,elastic5
```

Some of the blocks support dynamic change of the image version used in the Docker file. The signature looks like this:

```
make devenv sources=postgres,openldap,grafana postgres_version=9.2
grafana_version=6.7.0-beta1
```

Notes per block

Grafana

The grafana block is pre-configured with the dev-datasources and dashboards.

Jaeger

Jaeger block runs both Jaeger and Loki container. Loki container sends traces to Jaeger and also logs its own logs into itself so it is possible to setup derived field for traceID from Loki to Jaeger. You need to install a docker plugin

for the self logging to work, without it the container won't start. See https://grafana.com/docs/loki/latest/clients/docker-driver/#installing for installation instructions.

Graphite

version	source name	graphite-web port	plaintext port	pickle port
1.1	graphite	8180	2103	2103
1.0	graphite1	8280	2203	2203
0.9	graphite09	8380	2303	2303

Debugging setup in VS Code

An example of launch.json is provided in `devenv/vscode/launch.json`. It basically does what Makefile and .bra.toml do. The 'program' field is set to the folder name so VS Code loads all *.go files in it instead of just main.go.

Troubleshooting

Containers that read from log files fail to start (Mac OS)

If you are running Mac OSX, containers that read from the log files (e.g. Telegraf, Fileabeat, Promtail) can fail to start. This is because the default Docker for Mac does not have permission to create `grafana` folder at the `/var/log` location, as it runs as the current user. To solve this issue, manually create the folder `/var/log/grafana`, then start the containers again.

```
sudo mkdir /var/log/grafana
```