Install and setup graylog in docker

Graylog5 docs

I have a router that has both a syslog and a firewall log, the routers interface for logs is cumbersome and paged with about 100 lines per page.
I would like to view my logs a different way and I am therefore trying graylog. My install env is docker running on a spare computer, i have portainer to manage containers and stacks (docker compose).

Inspiration is from this video from Lawrence Systems
Graylog: Your Comprehensive Guide to Getting Started Open Source Log Management

Installing graylog

1. I am basing my install on this docker compose file from this github repo
2. Generate variables for the environment using this example

```
# You MUST set a secret to secure/pepper the stored user passwords here.
# Use at least 64 characters.
# Generate one by using for example: pwgen -N 1 -s 96
# ATTENTION: This value must be the same on all Graylog nodes in the
cluster.
# Changing this value after installation will render all user sessions
# and encrypted values in the database invalid. (e.g. encrypted access
tokens)
GRAYLOG_PASSWORD_SECRET=""

# You MUST specify a hash password for the root user (which you only need
to initially set up the
# system and in case you lose connectivity to your authentication backend)
# This password cannot be changed using the API or via the web interface.
If you need to change it,
# modify it in this file.
# Create one by using for example: echo -n yourpassword | shasum -a 256
# and put the resulting hash value into the following line
# CHANGE THIS!
GRAYLOG_ROOT_PASSWORD_SHA2=""
```

3. Create a new stack in portainer and paste the docker cmpose file contents
4. Add generated pw secret and root pw as portainer stack variables
5. Change any port assignments that might be conflicting with other service
6. Deploy the stack

7. Access the web interface on `localhost:9000` or another port if you changed it in the stack setup

The stack i ended up using was this:

```
# From https://github.com/Graylog2/docker-compose/tree/main/open-core

version: "3.8"

services:
  mongodb:
    image: "mongo:5.0"
    volumes:
      - "mongodb_data:/data/db"
    restart: unless-stopped

  opensearch:
    image: "opensearchproject/opensearch:2.4.0"
    environment:
      - "OPENSEARCH_JAVA_OPTS=-Xms1g -Xmx1g"
      - "bootstrap.memory_lock=true"
      - "discovery.type=single-node"
      - "action.auto_create_index=false"
      - "plugins.security.ssl.http.enabled=false"
      - "plugins.security.disabled=true"
    ulimits:
      memlock:
        hard: -1
        soft: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      - "os_data:/usr/share/opensearch/data"
    restart: unless-stopped

  graylog:
    hostname: "server"
    image: "${GRAYLOG_IMAGE:-graylog/graylog:5.1.5}"
    depends_on:
      opensearch:
        condition: "service_started"
      mongodb:
        condition: "service_started"
    entrypoint: "/usr/bin/tini -- wait-for-it opensearch:9200 --  /docker-
entrypoint.sh"
    environment:
      GRAYLOG_NODE_ID_FILE: "/usr/share/graylog/data/config/node-id"
      GRAYLOG_PASSWORD_SECRET: "${GRAYLOG_PASSWORD_SECRET:?Please configure
GRAYLOG_PASSWORD_SECRET in the .env file}"
      GRAYLOG_ROOT_PASSWORD_SHA2: "${GRAYLOG_ROOT_PASSWORD_SHA2:?Please
configure GRAYLOG_ROOT_PASSWORD_SHA2 in the .env file}"
      GRAYLOG_HTTP_BIND_ADDRESS: "0.0.0.0:9100"
      GRAYLOG_HTTP_EXTERNAL_URI: "http://localhost:9100/"
      GRAYLOG_ELASTICSEARCH_HOSTS: "http://opensearch:9200"
      GRAYLOG_MONGODB_URI: "mongodb://mongodb:27017/graylog"
    ports:
```

```
      - "2055:2055/udp"   # Netflow udp
      - "5044:5044/tcp"   # Beats
      - "514:5140/udp"   # Syslog
      - "5140:5140/tcp"   # Syslog
      - "5555:5555/tcp"   # RAW TCP
      - "5555:5555/udp"   # RAW TCP
      - "9100:9100/tcp"   # Server API
      - "12201:12201/tcp" # GELF TCP
      - "12201:12201/udp" # GELF UDP
      - "13301:13301/tcp" # Forwarder data
      - "13302:13302/tcp" # Forwarder config
    volumes:
      - "graylog_data:/usr/share/graylog/data/data"
      - "graylog_journal:/usr/share/graylog/data/journal"
    restart: unless-stopped

volumes:
  mongodb_data:
  os_data:
  graylog_data:
  graylog_journal:
```

New user

The root users timezone is UTC and also the timestamp that logs are stored as.
I prefer to view the logs in my local timezone and to do that i create a new user.

TIP to see the configured timezones click your user and select `System->Overview` then scroll down to see the configured timezones for:

User admin: 2023-09-30 13:42:53 +02:00
Your web browser: 2023-09-30 13:42:53 +02:00
Graylog server: 2023-09-30 11:42:53 +00:00

1. Got to `System->Users and Teams`
2. Create a user with admin priviledges, change timezone to your timezone
3. log out and then in with your new user creds, confirm timezone in `System->Overview`

Configure inputs

In order to get data to graylog you need to create inputs, my router is configured to send `syslog` data on UDP port 514 and `netflow` data on UDP port 2055.

1. Select `System->Inputs`
2. Click select input and select your source type, this example uses `SYSLOG UDP`
3. Give it a title, something sensible that yu choose

4. Select the port, this is the internal port of the docker container, i have mapped `UDP port 514` from outside the container to `UDP port 5140` inside the container (see stack config above for details)
5. Save
6. Check that data is recieved by clicking on the inputs `Show received messages` button, messages should appear

Create indicies

In order to route your data to another stream than the `Default Stream` you need to configure indices

1. Select `System->Indices`
2. Click `Create Index Set`
3. Fill out `Title` `Description` `Index Prefix` (no spaces in these names)
4. Configure `Index Rotation Configuration` and `Index Retention Configuration` to your preferences, remeber that logs can grow big!
5. Confirm with `Create Index Set` at the bottom of the page

Create stream

Putting data in specific streams makes it easier to navigate data later

1. Click `Streams` and `Create Streams`
2. Give the stream a name and optional a description
3. Select the index set to get data from (your index set from before should appear in the dropdown menu)
4. Check `Remove matches from 'Default Stream'` to remove matching data from the `Default Stream` (no need to store it twice)
5. Confirm with `Create Stream`

The status of your stream is `paused` and before you can use it you need to filter what data is received in the stream, to do this you need to grab the input `Field:Value` pair

1. Go to `System->Inputs` and click `Show Received Messages`, in the next window copy the string next to the magnifying glass example: `gl2_source_input:6517dd3c0b3aa72ee5489355`
2. Go back to streams and click `More->Manage Rules` on your corresponding stream
3. Click `Add Stream Rule` to create a new rule

4. In the Field input paste the field part of the `Field:Value` pair, example `gl2_source_input`
5. in the Value input paste the value part of the `Field:Value` pair, example `6517dd3c0b3aa72ee5489355`
6. Test the rule by selecting the input and `Load Message`, confirm that the filter is matching the message without errors
7. Click the stream name and confirm that data is collected in the new stream
8. Look at the default stream to confirm that older data is removed and not present anymore

Now, happy monitoring :-) Play around and see how you can query and filter your data.
I find it quite nice, ~a million times better than the DD-WRT interface log viewer!

Last update: 2023-09-30 12:44:15