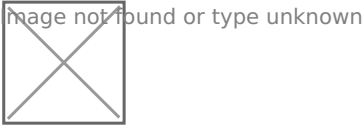


Install Umami to Monitor your Website Traffic



Umami is a simple, easy to use, self-hosted web analytics solution. The goal is to provide you with a friendlier, privacy-focused alternative to Google Analytics and a free, open-sourced alternative to paid solutions. Umami collects only the metrics you care about and everything fits on a single page. You can view a live demo [here](#) or read more about Umami [here](#).

Install Filebrowser (if you want to cheat and not use CLI)

```
version: "2.1"
services:
  filebrowser:
    image: hurlenko/filebrowser:latest
    container_name: filebrowser
    environment:
      - FB_BASEURL=/f
    volumes:
      - /:/data
      - /docker/filebrowser:/config
    ports:
      - 8081:8080
    restart: unless-stopped
```

Create the `schema.postgresql.sql` file and place it in `/docker/umami`

Paste the following into the `schema.postgresql.sql` file.

```
schema.postgresql.sql
```

```
drop table if exists event;
drop table if exists pageview;
```

```
drop table if exists session;
drop table if exists website;
drop table if exists account;

create table account (
    user_id serial primary key,
    username varchar(255) unique not null,
    password varchar(60) not null,
    is_admin bool not null default false,
    created_at timestamp with time zone default current_timestamp,
    updated_at timestamp with time zone default current_timestamp
);

create table website (
    website_id serial primary key,
    website_uuid uuid unique not null,
    user_id int not null references account(user_id) on delete cascade,
    name varchar(100) not null,
    domain varchar(500),
    share_id varchar(64) unique,
    created_at timestamp with time zone default current_timestamp
);

create table session (
    session_id serial primary key,
    session_uuid uuid unique not null,
    website_id int not null references website(website_id) on delete cascade,
    created_at timestamp with time zone default current_timestamp,
    hostname varchar(100),
    browser varchar(20),
    os varchar(20),
    device varchar(20),
    screen varchar(11),
    language varchar(35),
    country char(2)
);

create table pageview (
    view_id serial primary key,
    website_id int not null references website(website_id) on delete cascade,
```

```

    session_id int not null references session(session_id) on delete cascade,
    created_at timestamp with time zone default current_timestamp,
    url varchar(500) not null,
    referrer varchar(500)
);

create table event (
    event_id serial primary key,
    website_id int not null references website(website_id) on delete cascade,
    session_id int not null references session(session_id) on delete cascade,
    created_at timestamp with time zone default current_timestamp,
    url varchar(500) not null,
    event_type varchar(50) not null,
    event_value varchar(50) not null
);

create index website_user_id_idx on website(user_id);

create index session_created_at_idx on session(created_at);
create index session_website_id_idx on session(website_id);

create index pageview_created_at_idx on pageview(created_at);
create index pageview_website_id_idx on pageview(website_id);
create index pageview_session_id_idx on pageview(session_id);
create index pageview_website_id_created_at_idx on pageview(website_id, created_at);
create index pageview_website_id_session_id_created_at_idx on pageview(website_id, session_id,
created_at);

create index event_created_at_idx on event(created_at);
create index event_website_id_idx on event(website_id);
create index event_session_id_idx on event(session_id);

insert into account (username, password, is_admin) values ('admin',
'$2b$10$BULi0c.muyCWlErNJc3jL.vFRftFJWrT8/GcR4A.sUdCznaXiqFXa', true);

```

Run the docker stack and install

```

version: '3'

services:
    umami:

```

```
image: ghcr.io/mikecao/umami:postgresql-latest
ports:
  - "3000:3000"
environment:
  DATABASE_URL: postgresql://umami:umami@db:5432/umami
  DATABASE_TYPE: postgresql
  HASH_SALT: H6ei601tdLNxIQLRs4Mw
depends_on:
  - db
restart: always

db:
image: postgres:12-alpine
environment:
  POSTGRES_DB: umami
  POSTGRES_USER: umami
  POSTGRES_PASSWORD: umami
volumes:
  - /docker/umami/schema.postgresql.sql:/docker-entrypoint-initdb.d/schema.postgresql.sql:ro
  - /docker/umami/db:/var/lib/postgresql/data
restart: always
volumes:
  umami-db-data:
```

Connect to the web UI

Go to `your.server.ip.here:3000` and log in using `admin` as the username and `umami` as the password.

Video tutorial

https://www.youtube.com/embed/nUjDGxazkOQ?ab_channel=Geeked

Did you find this helpful? [Subscribe to me on Youtube](#) for more content!

Revision #17

Created 3 September 2021 15:26:42 by Jeremy

Updated 29 September 2021 11:39:44 by Jeremy