

Random Services

- Pterodactyl
 - docker-compose

Pterodactyl

docker-compose

This is the docker-compose full-up service from <https://pterodactyl.io/>

This requires some customization on your router to port forward 80 & 443 & 2022 to the host with this compose for full functionality, but that is beyond the scope of this testing.

Below is an annotated docker-compose that should be largely copy & pastable if you already have swag running. If not then see <https://docs.linuxserver.io/general/swag> for more details on top-down configuration steps.

```
---
version: "2.1"
services:
  swag:
    image: linuxserver/swag
    container_name: swag
    cap_add:
      - NET_ADMIN
    environment:
      - PUID=1000
      - PGID=1000
      - TZ="America/Los Angeles"
      - URL=domain.com
      - SUBDOMAINS=pterodactyl,daemon1-pterodactyl
      - VALIDATION=http
      - ONLY_SUBDOMAINS=false
      - STAGING=false
    volumes:
      - /home/tokugero/apps/swag/config:/config
    ports:
      - 443:443
      - 80:80
    restart: unless-stopped
#####
#pterodactyl
```

#####

pteropaneldb:

container_name: ppdb

image: linuxserver/mariadb

restart: unless-stopped

environment:

- PUID=1000

- PGID=1000

- TZ=America/Los Angeles

- MYSQL_DATABASE=pterodactyl

- MYSQL_USER=pterodactyluser

- MYSQL_PASSWORD=pterodactylpass

volumes:

- ./pteropanel/mariadb/config:/config

restart: unless-stopped

pteropanelcache:

container_name: ppcache

restart: unless-stopped

image: redis

pteropanel:

container_name: pteropanel

restart: unless-stopped

image: ccarney16/pterodactyl-panel

links:

- pteropaneldb

- pteropanelcache

#####

#FIRST RUN STEPS

#add below "command" to allow log into container

#command: ["sleep", "infinity"]

#

#With compose running,

docker exec -it pteropanel sh

#

#Follow instructions for environments as needed and migrate & make user here:

https://pterodactyl.io/panel/0.7/getting_started.html#environment-configuration

#

#Then remove the command: directive

#####

volumes:

- "/.pteropanel/pteropanel/var:/app/var/"

#####

Must manually create ./pteropanel/pteropanel/log/nginx & ./pteropanel/pteropanel/log/php7 folders

####

- "/.pteropanel/pteropanel/log:/var/log/"

- "/.pteropanel/pteropanel/config:/data"

environment:

- "APP_ENV=production"

- "APP_DEBUG=true"

- "APP_THEME=pterodactyl"

- "APP_CLEAR_TASKLOG=720"

- "APP_DELETE_MINUTES=10"

- "APP_ENVIRONMENT_ONLY=false"

- "QUEUE_HIGH=high"

- "QUEUE_STANDARD=standard"

- "QUEUE_LOW=low"

- "CACHE_DRIVER=redis"

- "SESSION_DRIVER=redis"

- "QUEUE_DRIVER=redis"

- "REDIS_HOST=pteropanelcache"

- "REDIS_PASSWORD=null"

- "REDIS_PORT=6379"

- "APP_URL=https://pterodactyl.domain.com"

- "APP_TIMEZONE=America/Los_Angeles" ## <http://php.net/manual/en/timezones.php>

- "APP_SERVICE_AUTHOR=tokugero@domain.com"

- "DB_HOST=pteropaneldb"

- "DB_PORT=3306"

- "DB_DATABASE=pterodactylldb"

- "DB_USERNAME=pterodactyluser"

- "DB_PASSWORD=pterodactylpass"

- "TRUSTED_PROXIES=172.16.0.0/12,192.168.0.0/16,10.0.0.0/8"

pterodaemon:

container_name: pterodaemon

privileged: true

restart: unless-stopped

image: ccarney16/pterodactyl-daemon

ports:

- "2022:2022"

volumes:

- "/var/lib/docker/containers:/var/lib/docker/containers"

- "/var/run/docker.sock:/var/run/docker.sock"
- "/srv/daemon-data:/srv/daemon-data"
- "./pterodaemon/config:/srv/daemon/config"
- "/tmp/pterodactyl:/tmp/pterodactyl"
- "/etc/timezone:/etc/timezone:ro"

Below is the config for the daemon, this is the same config as the panel but with the `server_name` updated, the expectation is that this is publicly reachable for things like health checks & automation. I'm not clear on the ramifications of hiding this from public under other circumstances and this requires more testing.

```
server {
    listen 443 ssl;
    listen [::]:443 ssl;

    server_name daemon1-pterodactyl.*;

    include /config/nginx/ssl.conf;

    client_max_body_size 0;

    # enable for ldap auth, fill in ldap details in ldap.conf
    #include /config/nginx/ldap.conf;

    # enable for Authelia
    #include /config/nginx/authelia-server.conf;

    location / {
        # enable the next two lines for http auth
        #auth_basic "Restricted";
        #auth_basic_user_file /config/nginx/.htpasswd;

        # enable the next two lines for ldap auth
        #auth_request /auth;
        #error_page 401 =200 /ldaplogin;

        # enable for Authelia
        #include /config/nginx/authelia-location.conf;

        include /config/nginx/proxy.conf;
```

```

resolver 127.0.0.11 valid=30s;
set $upstream_app pterodaemon;
set $upstream_port 443;
set $upstream_proto http;
proxy_pass $upstream_proto://$upstream_app:$upstream_port;


proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_redirect off;
proxy_buffering off;
proxy_request_buffering off;
}
}

```

Below is the configs from a combination of bazaar & pterodactyl nginx recommendations. Host & Schema forwarding were removed from this to force this to work with the swag opinions. This should be copy/pastable.

```

server {
    listen 443 ssl;
    listen [::]:443 ssl;

    server_name pterodactyl.*;

    include /config/nginx/ssl.conf;

    client_max_body_size 0;

    # enable for ldap auth, fill in ldap details in ldap.conf
    #include /config/nginx/ldap.conf;

    # enable for Authelia
    #include /config/nginx/authelia-server.conf;

    location / {
        # enable the next two lines for http auth
        #auth_basic "Restricted";
        #auth_basic_user_file /config/nginx/.htpasswd;

        # enable the next two lines for ldap auth

```

```
#auth_request /auth;
#error_page 401 =200 /ldaplogin;

# enable for Authelia
#include /config/nginx/authelia-location.conf;

include /config/nginx/proxy.conf;
resolver 127.0.0.11 valid=30s;
set $upstream_app pteropanel;
set $upstream_port 80;
set $upstream_proto http;
proxy_pass $upstream_proto://$upstream_app:$upstream_port;

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_redirect off;
proxy_buffering off;
proxy_request_buffering off;
}
}
```

Once the Panel is up and the Daemon has run at least once, browse to the Settings > Nodes > Create New Node section in the Panel. The important notes are to use the SSL connection, set "Behind Proxy" as it's behind the swag proxy, and update the Daemon Port to match the swag configuration (443).

image-1599844322774.png

Copy the newly generated "Configuration" (/admin/nodes/view/1/configuration) from the panel and make a new file in the /pterodaemon/config folder on your host.

image-1599844425700.png

Restart the Daemon and verify that the Nodes health checks are passing from your browser:

image-1599844469858.png