How to install Passwork using Docker

Docker installation

Install Docker CE (https://docs.docker.com/engine/installation/).

Get root privileges and update packages info

```
sudo –i
apt-get update
apt-get upgrade
```

Install Git

```
apt-get install git
```

Load configuration files

Create a new directory **/server** and clone configuration files from public git repository:

```
mkdir /server

git clone https://garakh@bitbucket.org/passworkteam/env.git /server
```

Download Passwork Sources

Remove file from the destination folder:

```
rm /server/sites/prod/.gitkeep
```

Clone the repository using your login and password:

```
git clone http://get.passwork.pro:81/passwork/passwork.git /server/sites/prod
```

Create Passwork config.ini file from example

```
cp /server/sites/prod/app/config/config.example.ini /server/sites/prod/app/config/config.ini
```

Run Nginx in Docker

```
docker run -d --name=nginx \
--restart unless-stopped \
-p 80:80 -p 443:443  \
-v /server/conf/nginx:/server/conf/nginx \
-v /server/conf/php:/server/conf/php \
-v /server/conf/ssl:/server/conf/ssl \
-v /server/conf/postfix:/server/conf/postfix \
-v /server/log/nginx:/server/log/nginx \
-v /server/log/php:/server/log/php \
-v /server/log/syslog:/server/log/syslog \
-v /server/sites/:/server/sites/ \
passwork/nginx \
/bin/bash -c "/server/phalcon-enable-v3; /server/run"
```

Run MongoDB in Docker

```
docker run -d --name=db \
--restart unless-stopped \
-p 27017:27017 \
-v /server/conf/mongo:/server/conf/ \
-v /server/log/mongo:/server/log/ \
-v /server/data/mongo/:/server/data \
passwork/mongo \
/server/run
```

Take into account that 27017 port is open for everybody without authentication. You may want to set up your firewall to block incoming connections to the 27017 port.
Or you can bind a local IP address:

```
docker run -d --name=db \
--restart unless-stopped \
-p 10.0.0.1:27017:27017 \
-v /server/conf/mongo:/server/conf/ \
-v /server/log/mongo:/server/log/ \
-v /server/data/mongo/:/server/data \
passwork/mongo \
/server/run
```

This says to Docker to bind 27017 port with your local IP 10.0.0.1.

Check the both containers are running:

```
docker ps
```

Restore a database from backup

Copy the initial backup to Mongo container's folder:

```
cp -r /server/sites/prod/dump/ /server/data/mongo/dump
```

Execute ***mongorestore*** tool in Mongo container:

```
docker exec -i db mongorestore /server/data/dump
```

Set up Passwork config.ini

Open config.ini file using any editor:

```
mcedit /server/sites/prod/app/config/config.ini
```

Find section *[mongo]* and specify IP of your HOST server. Usually Docker creates a new network interface and assign 172.17.0.1 to the HOST operation system.

Make sure that your HOST server allows incoming connections from 172.17.0.0/24 to 27017 port.

```
[mongo]
connectionString = mongodb://172.17.0.1:27017
dbname = pwbox
useCreds = false
username =
password =
```

Useful commands

Bash to container:

```
docker exec -it <container> bash
```

Restore correct permissions:

```
/server/docker-nginx-permissions nginx
```

Reload Nginx without stopping:

```
/server/docker-nginx-reload nginx
/server/docker-php-reload nginx
```

Docker containers run with *autostart* feature. It means that Docker automatically launches containers again if a root process is down (the roots process are nginx and mongodb).

If you need to stop container you need to disable the *autostart* feature before:

```
/server/docker-norestart <container>
```

Don't forget to enable the autostart:

```
/server/docker-autorestart <container>
```

When the containers are not autostartable you can stop the services:

```
/server/docker-nginx-stop nginx
/server/docker-mongo-stop db
```

If the containers are autostartable these commands just restart the services.

Hard way to stop any container:

```
docker stop <container>
```

Use it in emergency case because it can corrupt containers data.

Structure of files

Configuration files:

```
/server/conf/
```

Data (mongo database):

```
/server/data/
```

Logs:

```
/server/log/
```

Web-site:

```
/server/sites/
```

Example: How to edit nginx/php configuration

Edit configuration files:

```
mcedit /server/conf/nginx/prod.site
mcedit /server/conf/nginx/nginx.conf
mcedit /server/conf/php/php.ini
```

Restart nginx and php-fpm:

```
/server/docker-nginx-reload nginx
/server/docker-php-reload nginx
```

Mail Configuration

Docker container uses Postfix to send emails. Postfix configuration files are stored at

```
/server/conf/postfix/
```

Feel free to edit them to configure Postfix for your goals.
Don't forget to reload Postfix to apply changes:

```
docker exec -i nginx service postfix reload
```

Done

Now you can open your server in a browser. You should see Passwork landing page.

Click «Log in» and use default built-in account:

```
Login: admin@passwork.me
Password: DemoDemo
```