
VODAN in a Box

VODAN Team

Jul 29, 2020

ABOUT

1	About the VODAN in a Box	3
1.1	What is VODAN in a Box?	3
1.2	What is in the “box”?	3
1.3	Demo instance	3
2	Usage Scenarios	5
2.1	Data-entry only	5
2.2	Metadata publication only	6
2.3	Data-entry and metadata publication (complete package)	6
3	Components	7
4	Local Deployment	9
4.1	Requirements	9
4.2	Setup	9
4.3	Usage	9
4.4	Update	10
4.5	Notes	10
5	Production Deployment	11
5.1	Requirements	11
5.2	Setup	11
5.3	Update	13
5.4	Notes	14
6	Advanced Configuration	15
6.1	Persistence	15
6.2	CRF Data Submission	16
6.3	Changing ports	17
6.4	CRF visibility	17
6.5	CRF Wizard emails	17

Welcome to the *VODAN in a Box* documentation mainly focused on deployment.

ABOUT THE VODAN IN A BOX

1.1 What is VODAN in a Box?

VODAN in a Box (ViaB) is a toolset to facilitate the capture of data related to virus outbreaks and the publication of metadata describing these datasets.

The toolset can be deployed wherever the user wants. It can be deployed in a cloud provider, in a server or on a local machine. Naturally, the first two options can be made accessible anywhere on the Web while the third option is normally for testing and demonstration purposes only. This deployment freedom provides flexibility for users who want **ViaB** the (meta)data to be stored locally (e.g., in a given hospital), nationally or internationally.

1.2 What is in the “box”?

VODAN in a Box is composed of:

- Data Stewardship Wizard (DSW) - to capture and store data based on WHO’s COVID-19 CRF;
- FAIR Data Point (FDP) - to publish metadata about the COVID-19 CRF dataset and other pandemic-related content;
- WHO COVID-19 Rapid Version CRF Semantic Data Model - this semantic data model has been embedded in DSW to provide semantically-rich RDF export to the data entered with the DSW.

1.3 Demo instance

You can explore and try out *VODAN in a Box* using our instance intended for demonstration purposes:

- CRF Wizard dsw.vodan.fairdatapoint.org
- FAIR Data Point fdp.vodan.fairdatapoint.org
- Triple Store (AllegroGraph) sparql.vodan.fairdatapoint.org

Be aware that it is for **demonstration purposes only**. The data and metadata do not reflect real measurements or observations and should not be used for analysis of real-world phenomena.

USAGE SCENARIOS

The **VODAN in a Box (ViaB)** toolset can be used in the following scenarios:

2.1 Data-entry only

In this scenario, only the VODAN DSW and its COVID-19 semantic data model are used as a data-entry tool. Users can fill in the DSW's web form with data to report COVID-19 cases.

Normally, this scenario is indicated for the cases where metadata about the data does not need to be published.

Those use cases are require user to be logged in in VODAN DSW:

2.1.1 Create eCRF

1. Select *CRFs* from the left menu
2. Press *Create* button
3. Fill-in identifier and press *Save* button
4. Fill CRF with data you have, *Save* or *Discard* changes accordingly

2.1.2 Update eCRF

1. Select *CRFs* from the left menu
2. Find by name the CRF you want to edit
3. Fill CRF with new data you have, *Save* or *Discard* changes accordingly

2.1.3 Submit eCRF

1. Open CRF you want to submit
2. Press *Create Report*
3. Press *Create* (optionally, you can name the report, e.g. "My report - v0.1")
4. Press three dots on the right for the new report and press *Submit*
5. Select the triple store you want to use and press *Submit*

2.2 Metadata publication only

In this scenario, only the VODAN FDP is used to publish the pandemic-related content. This option is indicated for the cases where data have already been captured and only the FAIR metadata about them need to be published.

Data could have been made available from other CRF-entry tools and extracted directly from databases of information systems such as electronic health record systems.

2.3 Data-entry and metadata publication (complete package)

In this scenario, the whole **VODAN in a Box** is used, covering the data capture, semantic data generation and metadata publishing.

COMPONENTS

VODAN-in-a-Box consists of two significant services:

- **Data Stewardship Wizard (DSW)** adjusted to serve as Wizard for filling and maintaining electronic case report forms (eCRF),
- **FAIR Data Point (FDP)** to maintain metadata about eCRFs created in DSW.

To support it, there are other services included:

- AllegroGraph triple store for eCRF data and queries,
- BlazeGraph triple store for FDP,
- MongoDB used by both DSW and FDP,
- JSON server providing controlled vocabulary for filling answers,
- Submission Service that handles storing eCRFs in triple store and updating metadata in FDP,
- RabbitMQ for queueing generation of an eCRF to a RDF document using DSW document worker,
- (optionally) Nginx proxy for *Production Deployment*.

LOCAL DEPLOYMENT

Important: This deployment is intended only for testing and demonstration purposes and should not serve for real production use. If you want to provide *VODAN in a Box* as a service, visit [Production Deployment](#).

4.1 Requirements

- Docker Engine version 19.03 (or higher)
- Docker Compose version 1.25 (or higher)

4.2 Setup

1. Download or `git clone` repository <https://github.com/VODAN-Tech/vodan-deployment-basic> locally
2. Change working directory to the root folder `vodan-deployment-basic`
3. Use `docker-compose` to start *VODAN in a Box*

```
git clone https://github.com/VODAN-Tech/vodan-deployment-basic.git
cd vodan-deployment-basic
docker-compose up -d
```

For additional configuration options, see [Advanced Configuration](#).

4.3 Usage

When *VODAN in a Box* is running, you can access the following services:

- <http://localhost:8080> - CRF Wizard (DSW)
- <http://localhost:8081> - FAIR Data Point (FDP)
- <http://localhost:8085/blazegraph> - BlazeGraph
- <http://localhost:27017> - MongoDB (for MongoDB clients)
- <http://localhost:3000> - CRF Wizard API

For both CRF Wizard and FDP, you can use default admin account `albert.einstein@example.com` with password `password`. BlazeGraph and MongoDB are without any authentication.

- To start *VODAN in a Box*, use `docker-compose up -d` in the root directory.
- To stop *VODAN in a Box*, use `docker-compose down` in the root directory.
- To restart *VODAN in a Box*, use first `docker-compose down` and then `docker-compose up -d` again.
- To see running services of *VODAN in a Box* and their status, use `docker-compose ps`.
- For debugging and investigating logs, use `docker-compose logs` (or `docker-compose logs -f`).

Optionally, you can also use separate [AllegroGraph](#) for submitted CRF data. To do that, simply uncomment `agraph` section in `docker-compose.yml` and update `submission-service/config.yml`. Then, you will be able to access it on <http://localhost:10035>. Of course, you can similarly set any other triple store of your choice.

4.4 Update

1. Stop *VODAN in a Box*
2. Overwrite configurations and `docker-compose.yml` or simply `git pull`
3. Start *VODAN in a Box* again

From root directory of `vodan-deployment-basic`:

```
docker-compose down
git pull
docker-compose up -d
```

4.5 Notes

For more information about `docker-compose` and its options, visit [Docker documentation](#).

Various advanced deployment options of FAIR Data Point are well-described in [FAIR Data Point Reference Implementation Documentation](#).

The main difference with respect to the *Production Deployment* is the absence of proxy and certificates, with opened ports directly instead.

PRODUCTION DEPLOYMENT

Important: This deployment is intended for production use. If you want to just test *VODAN in a Box* locally, visit *Local Deployment*.

5.1 Requirements

- Docker Engine version 19.03 (or higher)
- Docker Compose version 1.25 (or higher)
- Domain and DNS records set for providing *VODAN in a Box*:
 - `dsw.your-domain.tld` - for CRF Wizard (DSW)
 - `api.dsw.your-domain.tld` - for CRF Wizard API (DSW API)
 - `fdp.your-domain.tld` - for FAIR Data Point
 - `sparql.your-domain.tld` - for Triple Store (CRF data)
- certbot

5.2 Setup

5.2.1 Get VODAN in a Box

Download or `git clone` repository <https://github.com/VODAN-Tech/vodan-deployment-production> locally.

The folder `vodan-deployment-production` we call *VODAN in a Box* root directory. It consists all necessary configuration files and `docker-compose.yml`.

5.2.2 Configure domains and secrets

There are several things that you need to configure before running *VODAN in a Box* for production deployment. In files, look for comments marked with (!):

1. `server_name` and `ssl_certificate` values in `proxy/nginx/agraph.conf`, `proxy/nginx/dsw.conf`, and `proxy/nginx/fdp.conf` with your domain names. Those need to have valid DNS records pointing to that server.
2. `docker-compose.yml` - `API_URL` (`dsw_client` service) to your value for `api.dsw.your-domain.tld`
3. `dsw-server/application.yml` - `clientId` to your value for `dsw.your-domain.tld`, then `secret`, `serviceToken`, and `email` section according to the comments there
4. `fdp/application.yml` - `clientId` to your value for `fdp.your-domain.tld` and then `,` `persistentUrl`, `secret`, `serviceToken`, and `secret-key` (JWT)
5. `allegrograph/agraph.cfg` - set **strong password** and optionally change username using `SuperUser` directive, the same credentials must be configured in `submission-service/config.yml`

5.2.3 Obtain SSL certificates

Before providing *VODAN in a Box* you need also to get SSL certificates to be able to use HTTPS. We recommend using Let's Encrypt but you can use any other way and change Nginx proxy configuration accordingly.

1. Comment out include lines at the end of `proxy/nginx/nginx.conf`
2. Start the proxy service

```
docker-compose up -d proxy
```

3. Get certificates for your domains:

```
sudo certbot certonly --webroot -w ./proxy/letsencrypt -d dsw.your-domain.tld
```

```
sudo certbot certonly --webroot -w ./proxy/letsencrypt -d api.dsw.your-domain.tld
```

```
sudo certbot certonly --webroot -w ./proxy/letsencrypt -d fdp.your-domain.tld
```

```
sudo certbot certonly --webroot -w ./proxy/letsencrypt -d sparql.your-domain.tld
```

4. Create certificate file for AllegroGraph (it needs to merge `cert.pem` and `privkey.pem` obtained by Let's Encrypt into a single file):

```
sudo cat /etc/letsencrypt/live/sparql.your-domain.tld/cert.pem /etc/letsencrypt/live/
↪sparql.your-domain.tld/privkey.pem > ./allegrograph/cert.pem
```

5. Stop the proxy service

```
docker-compose down
```

6. Uncomment lines at the end of `proxy/nginx/nginx.conf`

If getting certificates fail, it can be caused by incorrectly set DNS records. Optionally, verify if Nginx container is running and view its logs. You should also setup certificates renewal according to [Certbot documentation](#).

5.2.4 First start

1. Start *VODAN in a Box* (and wait a bit until all services start).

```
docker-compose up -d
```

2. Navigate to `dsw.your-domain.tld`, login using `albert.einstein@example.com` with password `password` and change default user accounts with **strong passwords**.
3. In `sparql.your-domain.tld`, create a repository `crf` in `catalog /` and create other users with permissions according to your needs (see [AllegroGraph documentation](#) for details). For example, create an *anonymous* user with only *read* permissions to `catalog /` and repository `crf`.
4. Navigate to `fdp.your-domain.tld` and login again as `albert.einstein@example.com` and change default user accounts with **strong passwords**.
5. In `fdp.your-domain.tld`, create and publish `catalog`, `dataset`, and `distribution` representing CRF data based on your use case.
6. Update `submission-service/config.yml` with **UUID** (Universally Unique Identifier) of your `distribution URL` from `FDP`, e.g. from `https://fdp.vodan.fairdatapoint.org/distribution/3335345b-ee66-4678-ab73-74a4b6ea1bee` it would be `3335345b-ee66-4678-ab73-74a4b6ea1bee`. (If you used different than `crf` repository name in triple store, change `sparql-endpoint` accordingly.)
7. Restart *VODAN in a Box* and wait a bit until all services start up (depending on your hardware, less than a minute).

```
docker-compose down
docker-compose up -d
```

8. Verify setup by creating CRF, saving it, creating a report, and submitting a report.

After this, your *VODAN in a Box* is ready to be used!

To check if everything is working, you can use `docker-compose logs` and `docker-compose ps` commands.

For additional configuration options, see [Advanced Configuration](#).

5.3 Update

1. Stop *VODAN in a Box*
2. Overwrite configurations and `docker-compose.yml` or simply `git pull`
3. Check if there are new configuration values to be changed according to your setup (marked with `(!)` comments)
4. Start *VODAN in a Box* again

From root directory of `vodan-deployment-production`:

```
docker-compose down
git pull
docker-compose up -d
```

This may need you to `git stash` your changes and then `git stash pop` them (and eventually solve `git` conflicts).

5.4 Notes

For more information about docker-compose and its options, visit [Docker documentation](#).

Various advanced deployment options of FAIR Data Point are well-described in [FAIR Data Point Reference Implementation Documentation](#). Similarly, for more details about DSW which used as CRF Wizard, see [Data Stewardship Wizard documentation](#).

The main difference with respect to the *Local Deployment* is the adding Nginx proxy, certificates, and other additional security.

ADVANCED CONFIGURATION

To work with *VODAN in a Box* you are not required to change anything in the included `docker-compose.yml` nor configuration files. For some specific use cases you might want to make some of the following changes.

6.1 Persistence

In the basic setup, persistence is assured using mounted folders (bind mounts):

- `./mongo/data` - for MongoDB (used by both FDP and CRF Wizard)
- `./blazegraph` - for BlazeGraph triple store (used both by FDP and as CRF-in-RDF data storage)

This allows you to easily work with data used by *VODAN in a Box*. For example, you can clear those folders (while it is not running) to start over. In some cases you might want to use [Docker volumes](#) instead. Using Docker volumes is recommended when using Docker for Windows due to common problems related to mounting Windows folders into Linux containers.

```
# ...
mongo:
  image: mongo:4.2.3
  restart: always
  ports:
    - 27017:27017
  environment:
    MONGO_INITDB_DATABASE: wizard
  volumes:
    - mongoData:/data/db # <- USING DOCKER VOLUME
    - ./mongo/init-mongo.js:/docker-entrypoint-initdb.d/init-mongo.js:ro
# ...

blazegraph:
  image: metaphacts/blazegraph-basic:2.2.0-20160908.003514-6
  ports:
    - 8085:8080
  volumes:
    - blazegraphData:/blazegraph-data # <- USING DOCKER VOLUME
# ...

volumes:
  mongoData:
  blazegraphData:
```

To avoid persistence totally (i.e. all data will be lost after `docker-compose down`). Just comment out or delete lines related to mounting volumes in `docker-compose.yml`:

```
# ...
mongo:
  image: mongo:4.2.3
  restart: always
  ports:
    - 27017:27017
  environment:
    MONGO_INITDB_DATABASE: wizard
  volumes:
    # - ./mongo/data:/data/db
    - ./mongo/init-mongo.js:/docker-entrypoint-initdb.d/init-mongo.js:ro
# ...

blazegraph:
  image: metaphacts/blazegraph-basic:2.2.0-20160908.003514-6
  ports:
    - 8085:8080
  #volumes:
  # - ./blazegraph:/blazegraph-data
```

Important: Data backups are your responsibility. It is recommended to backup regularly all mounted volumes and store such backups in different site(s).

6.2 CRF Data Submission

To simplify the setup, *VODAN in a Box* uses the same triple store and the same namespace for both FAIR Data Point data and data of submitted CRFs. You can easily change this behavior using a configuration file `submission-service/config.yml`. All you need to have is URL of SPARQL endpoint to be used for data submission. Additionally, if you want to maintain metadata in the FAIR Data Point you need to have a URL to distribution to be updated on submission.

```
triple-store:
  sparql-endpoint: http://my-triple.store/repository/my-crf-repo/sparql # <- change_
  ↪to your SPARQL endpoint
  auth: # <- only if triple store uses auth
    method: BASIC # <- authentication method: BASIC (default) or DIGEST
    username: usernameToMyTripleStore # <- change to your triple store username
    password: passwordToMyTripleStore # <- change to your triple store password
  graph: # !! do not change this section
    named: true
    type: http://purl.org/vodan/whocovid19crfsemadatamodel/who-covid-19-rapid-crf

fdp:
  token: a274793046e34a219fd0ea6362fcc61a001500b71724f4c973a017031653c20 # !! do_
  ↪not change this
  distribution: http://fdp_client/distribution/<distribution_uuid> # <- change UUID_
  ↪(obtained from FAIR Data Point)
```

Do not forget to restart *VODAN in a Box* after making the changes using `docker-compose down && docker-compose up -d`.

6.3 Changing ports

If you need to change ports because you already use those for other services, you just need to adjust the mappings in `docker-compose.yml` file. For example, if you want to access BlazeGraph on other port than 8085 change the mapping `8085:8080` to something else, e.g. `8885:8080`.

```
# ...  
  
blazegraph:  
  image: metaphacts/blazegraph-basic:2.2.0-20160908.003514-6  
  ports:  
    - 8885:8080 # <- USING 8885 INSTEAD OF 8085  
  volumes:  
    - ./blazegraph:/blazegraph-data
```

6.4 CRF visibility

You can easily change settings regarding CRF visibility according to your needs. In CRF Wizard (DSW), navigate as administrator to *Settings* and *CRFs*. You can allow to set visibility per single CRF upon its creation and also select the default one:

- **Public** = every user can view and edit the CRF
- **Public Read-only** = every user can view the CRF but only owner can edit it
- **Private** = only owner can view and edit the CRF

6.5 CRF Wizard emails

There is optional configuration in `dsw-server/application.yml` related to email server. You need that to enable:

- User registrations with email-based verification: upon registration a verification email is sent, otherwise administrator have to set new accounts as *Active* manually in users administration.
- Password recovery: when someone forgots password, they can ask for reset link that will be sent to their email address, otherwise it can be again changes only by administrators.

To make those emails working, fill the configuration with your SMTP server and account. We recommend using secured emails with SSL/TLS or STARTTLS. For more information, visit [DSW documentation](#).

Note: Registrations can be totally turned off using *Settings* and *Authentication*.
