visionect/visionect-server-v3 ☆

By visionect • Updated 20 days ago

Visionect Server 3 public repository

Container

Overview Tags

Running Visionect Server 3

Visionect Server 3 is a container/Docker-first software, which means that it is run in a container by default. Native Ubuntu packages are not available anymore.

You will need to install Docker, which is supported on most modern operating systems and natively integrated in latest Microsoft Windows Server 2016. More information regarding Docker installation can be found at: https://docs.docker.com/engine/installation/

Quick start

Requirements: Docker (http://www.docker.com/)

This will deploy a fresh Visionect Server 3 container, PostgreSQL container and configure data volumes for data persistence.

```
# create postgres container
docker run -d --restart=always -e POSTGRES_PASSWORD=visionect -e POSTGRES_USER=visionect -e POSTGRES_DB
# create data container  named 'vdata' (persistent volumes for things like, log files, firmware package
docker create --name vdata visionect/visionect-server-v3
# run docker container
docker run --privileged --cap-add=MKNOD --cap-add SYS_ADMIN -v /dev/shm:/dev/shm --device /dev/fuse -d
```

Quick start using Docker-compose

All the required services and the Visionect Server can also be started using Docker Compose. Create a file, named: `docker-compose.yml` :

```
version: "3"
services:
  vserver:
    image: visionect/visionect-server-v3:latest
    shm_size: 1GB
    depends_on:
      - vserver_postgres
      - vdata
    privileged: true
    cap_add:
      - MKNOD
      - SYS_ADMIN
    devices:
      - "/dev/fuse:/dev/fuse"
    restart: always
    ports:
      - 8081:8081
      - 11113:11113
      - 11112:11112
    links:
       - vserver_postgres:db2_1
    volumes_from:
      -/dev/shm:/dev/shm
      - vdata
    environment:
      - DB2_1_PORT_5432_TCP_ADDR=db2_1
  vdata:
    image: visionect/visionect-server-v3:latest
    entrypoint: /bin/true
  vserver_postgres:
    image: postgres
    restart: always
    environment:
      - POSTGRES_USER=visionect
      - POSTGRES_PASSWORD=visionect
      - POSTGRES_DB=koala
```

Start it up:

```
docker-compose up
```

Advanced installation guide for Visionect Server 3

Installation guide covers advanced installation topics for Visionect Server 3. Please read through this if you plan on using Visionect Server 3 in production.

Visionect Server 3 requires access to a PostgreSQL database (https://www.postgresql.org/) to store configuration and events. Minimum requirement a PostgreSQL server version 9.4 or higher. This does not need to be a fresh server install (we can use an existing server) and it can be run both natively or via Docker.

1. Install PostgreSQL

If you already have a PostgreSQL database version 9.4 or higher running, then just provision a new database for Visionect Server 3 to use. If not follow the instructions:

1.a. Dockerised PostgreSQL

This is the simplest way to run PostgreSQL, but it is not the recommended way for production setups. It is generally regarded that running databases in a container is not the recommended practice for high-performance and high-volume production setup.

To start using Docker: run the PostgreSQL container and setup the database using `POSTGRES_PASWORD` , `POSTGRES_USER` and `POSTGRES_DB` variables:

```
docker run -d --restart=always -e POSTGRES_PASSWORD=visionect -e POSTGRES_USER=visionect -e POSTGRES_DB
```

PostgreSQL Container Upgrades

If you need to upgrade the PostgreSQL container, you will need to take care about migrating data from old to new container. We recommend you check out the following links below.

- About migrations: https://www.postgresql.org/docs/9.0/static/migration.html
- Postgres Docker container: https://hub.docker.com/_/postgres/

1.b. Install directly on host or other machine

These instructions are tailored for Ubuntu based installs, but general configuration steps still apply to other platforms. Generally speaking Visionect Server requires a PostgreSQL database with a username and password running on a PostgreSQL v.9.4 server.

If not already installed, install PostgreSQL v.9.4 or higher:

For Ubuntu 16.04 LTS based distributions:

```
apt-get install postgresql-server
```

For other systems check: https://wiki.postgresql.org/wiki/Detailed_installation_guides

Create a new database, new user and new password for Visionect Server:

Run in the console of the PostgreSQL server:

```
$sudo -u postgres psql postgres
psql (9.6.1)
Type "help" for help.
```

```
postgres=# create user visionect;
CREATE ROLE
postgres=# \password visionect
Enter new password:
Enter it again:
postgres=# create database koala with owner=visionect;
CREATE DATABASE
postgres=# \q
```

Use `ifconfig` to find out PostgreSQL server IP address:

Run in the console of the PostgreSQL server:

```
$ifconfig
docker0   Link encap:Ethernet  HWaddr 56:84:7a:fe:97:99
inet addr:172.17.42.1  Bcast:0.0.0.0  Mask:255.255.0.0
...

eth0      Link encap:Ethernet  HWaddr 08:00:27:3b:97:3f
inet addr:192.168.101.62  Bcast:192.168.101.255  Mask:255.255.255.0
...
```

If postgres is running on the same host as server, use `docker` (e.g `POSTGRES_IP=172.17.42.1`), can be used, in opposite case use `eth[0-9]*` external interface (e.g. `POSTGRES_IP=182.168.101.62`).

Remember this IP, in last step you will need it to set `DB2_1_PORT_5432_TCP_ADDR=POSTGRES_IP` variable.

Make PostgreSQL to accept remote connections:

Use your favorite text editor (e.g. vi, nano, ..) to modify the `/etc/postgresql/PSQL.VERSION/main/postgresql.conf` file.

```
listen_addresses = '*'
```

WARNING: For security reasons you might want limit `listen_addresses` from all interfaces (i.e. `'*'`) to `POSTGRES_IP` only.

If `vserver` is running on the same host, use the IP from the previus step:

```
listen_addresses = '172.17.42.1'
```

For details, see: https://www.postgresql.org/docs/9.4/static/runtime-config-connection.html for more details.

Also modify/create /etc/postgresql/PSQL.VERSION/main/pg_hba.conf file.

```
host      koala      visionect      0.0.0.0/0      md5
```

WARNING: For security reasons, you migt want to limit access to koala database to `vserver` container only.

Here's an example where the IP of the local `vserver` container is 172.17.0.3 (we will create the `vdata` container in next step):

```
host      koala      visionect      172.17.0.3/32      md5
```

If `vdata` is runing on another machine than PostgreSQL, we have to use the IP of the external interface that can access the container:

For details see: https://www.postgresql.org/docs/9.4/static/auth-pg-hba-conf.html

2. Install the Visionect Server container

Create the `vdata` persistent storage container for storing the log files:

```
# create data container  named 'vdata' (persistent volumes for things like, log files, firmware package
docker create --name vdata visionect/visionect-server-v3
```

Run the Visionect server container and connect it to PostgreSQL - use the same database name, username and password as you did when setting up PostgreSQL.

Non-default DB parameters should be pased when creating new continer using optional `DB2_1_PORT_5432_TCP_ADDR` , `DB2_1_PORT_5432_TCP_USER` , `DB2_1_PORT_5432_TCP_PASS` and `DB2_1_PORT_5432_TCP_DB` enviorment variables.

Default parmeter values are: `ADDR: "localhost"` , `USER: "visionect"` , `PASS: "visionect"` and `DB:"koala"` .

2.a If PostgreSQL is running in Docker and on the same host, then two containers can be linked like this:

```
# Run docker container
docker run --privileged --cap-add=MKNOD --cap-add SYS_ADMIN --device /dev/fuse -v /dev/shm:/dev/shm -d
```

2.b In all other cases:

```
# Run docker container.
docker run --privileged --cap-add=MKNOD --cap-add SYS_ADMIN -v /dev/shm:/dev/shm --device /dev/fuse -d
```

Migrate from Visionect Server 2 '.deb' installation to Dockerized Visionect Server 3

This is the case where PostgreSQL is already installed and we will use the database with the new server version.

1. Please upgrade the koala2 packages to the latest version to make sure that database migrations are in place.

```
sudo apt-get update && apt-get install koala2
```

2. Remove the DEB installation:

```
sudo apt-get remove koala2
```

3. Prepare PostgreSQL for use with Docker container

Allow access to the PostgreSQL from Docker - see "Make PostgreSQL to accept remote connections" above.

4. Copy the existing `config.json` configuration file content to the environmental variable `VISIONECT_SERVER_CONFIG` :

```
VISIONECT_SERVER_CONFIG=`cat /opt/visionect/koala2/config.json`
```

5. Create the Container

Don't forget to replace `POSTGRES_IP` , with real IP, when you run following commands:

```
# create data container  named 'vdata' (persistent volumes for things like, log files, firmware package
docker create --name vdata visionect/visionect-server-v3
## run docker container
docker run --privileged --cap-add=MKNOD --cap-add SYS_ADMIN -v /dev/shm:/dev/shm --device /dev/fuse -d
```

Update Visionect Server 2 Docker container to Visionect Server 3

This section is relevant for customers that were using Visionect Server 2 Docker images in the past.

1. Remove the old container:

```
docker stop vserver
docker rm vserver
```

2. Migrate the data:

2.a If you don't care about migrating data:

```
docker rm vdata
## create new vdata container
docker create --name vdata visionect/visionect-server-v3
```

2.b If you want to migrate data from the old data container:

```
# rename old vdata container
docker rename vdata vdata_old
# create new vdata container
docker create --name vdata visionect/visionect-server-v3
```

Create a temprary Docker container to manualy migrate data.

```
docker run -ti --volumes-from vdata --volumes-from vdata_old --name vdata_migration alpine sh
```

Enter the container and migrate the logs:

```
cp -r /var/log/koala2/* /var/log/koala3/
```

If you are upgrading from version older than 2.12.0 which doesnt't yet store configuration in database, then you should also migrate the configuration, stored in `config.json` :

```
cp /opt/visionect/koala2/config/config.json /opt/visionect/koala3/config/config.json
```

Leave the temporary container:

```
exit
```

And remove it:

```
### stop and remove temporary container

docker stop vdata_migration
docker rm vdata_migration
# stop and removie old vdata container
docker rm vdata_old
```

3. Start the server

```
docker run --privileged --cap-add=MKNOD --cap-add SYS_ADMIN -v /dev/shm:/dev/shm --device /dev/fuse -d
```

Known Environmental Variables

This is a list of environmental variables that can be passed when starting the docker container.

Database-related Environmental Settings

The database settings below are mandatory and should be set if you use an external PostgreSQL server.

- `DB2_1_PORT_5432_TCP_ADDR` the TCP host address of the PostgreSQL server to use
- `DB2_1_PORT_5432_TCP_PORT` the TCP port of the PostgreSQL server
- `DB2_1_PORT_5432_TCP_USER` the PostgreSQL database owner username
- `DB2_1_PORT_5432_TCP_PASS` the PostgreSQL database owner password
- `DB2_1_PORT_5432_TCP_DB` the PostgreSQL database name to be used for this deployment

Server Configuration

The settings below are not mandatory but must be used to set up a master/slave server configuration.

- `VISIONECT_SERVER_CONFIG` if set, the content should be a JSON-formatted configuration file and wil be used on first run to store the entire server configuration. Use this if you want to migrate older `config.json` configuration to a new Visionect Server 3 setup.

- `VISIONECT_SERVER_DEPLOYMENT_KEY` if set, the value overrides any set deployment key in configuration and uses it to identify components of the deployment.
- `VISIONECT_SERVER_MASTER_HOST` if set, the value is used as the master host if setting up a master/slave configuration

Docker Pull Command

```
docker pull visionect/visionect-server-v3
```

Owner

visionect