Shinobi CCTV docker

- Instalação e configurações Shinobi
  - Shinobi API
  - Instalação Shinobi Docker
  - Instalação Shinobi Docker Desktop Windows
  - Recriação de tabela Files do Shinobi Docker

Instalação e configurações Shinobi

Shinobi API

Link: https://docs.shinobi.video/api

RESTful API

Simple and Efficient

Everything that the official dashboard does is through the Shinobi API. Meaning If you decide to build your own user interface or just using certain functions of Shinobi; The API will make quick work of it.

When you authenticate with Shinobi it will offer you an Authorization Token. This token is your Session Key as well and can be used as an API Key. This key will remain active for 15 minutes after the last acivity or while your WebSocket is connected.

Let's Begin

Create an API Key or Login to get started.

- Authentication
- Managing API Keys

After Authentication

Managing API Keys by UI
/api/managing-api-keys-ui

Managing API Keys by API
/api/managing-api-keys-api

WebSocket Connection
/api/api-connecting-websocket

Get Monitors
/api/get-monitors

Get Detection Events
/api/get-events

Get Streams
/api/get-streams

Embedding Streams

/api/embedding-streams

Get Videos
/api/get-videos

Get Timelapse
/api/get-timelapse

Get FileBin
/api/get-fileBin

Custom Settings
/api/custom-settings

Add, Edit or Delete a Monitor
/api/add-edit-or-delete-a-monitor

Modifying a Video or Deleting it
/api/modifying-a-video-or-deleting-it

Monitor Triggers
/api/monitor-triggers

Superuser
/api/superuser-only

Administrator
/api/administrator-only

Monitor Presets
/api/monitor-states-preset-configurations

Schedules for Monitor Presets
/api/scheduling-for-monitors

System
/api/system-triggers

ONVIF Management through Shinobi
/api/direct-camera-management-via-onvif

Instalação Shinobi Docker

Link: https://gitlab.com/Shinobi-Systems/ShinobiDocker git clone https://gitlab.com/Shinobi-Systems/ShinobiDocker.git

Install Shinobi with Docker

2024-05-07

https://shinobi.video https://docs.shinobi.video/installation/docker

Quick Install

1. Run this in terminal.
   bash <(curl -s https://gitlab.com/Shinobi-Systems/Shinobi-Installer/raw/master/shinobi-docker.sh)

Advanced Install

1. Download this repository and enter it.
   - If you **do not have Docker** installed run `sh INSTALL/docker.sh`.
2. Review and modify the `docker-compose.yml` file.
   - Leave it as-is for default setup.
3. Run the preparation and starter script.
   bash setup_and_run.sh

Once Installed

You will be asked if you want to use the included database, default is Yes. Once complete open port 8080 of your Docker host in a web browser.

> The following tables offer a breakdown of the configurations that control how the `shinobi` and `shinobi-sql` services are set up and interact within your Docker environment. Adjustments can be made to these values directly in the associated `docker-compose` files to modify the behavior of the deployment as needed.

`docker-compose-sql.yml` : `shinobi-sql`
Service Environment Variables

Variable	Description	Default Value
MYSQL_ROOT_PASSWORD	The password for the MySQL root user.	`rootpassword`
MYSQL_DATABASE	The name of the database to create.	`ccio`
MYSQL_USER	The username for the database.	`majesticflame`
MYSQL_PASSWORD	The password for the database user.	`1234`

`docker-compose-main.yml` : `shinobi`
Service Build Arguments and Environment Variables

Build Arguments

Argument	Description	Default Value
SHINOBI_BRANCH	The branch of the Shinobi git repository to clone during the build process.	`dev`

Environment Variables

Variable	Description	Default Value
HOME	The home directory path within the container.	`/home/Shinobi`
DB_HOST	Hostname of the MySQL database server.	`shinobi-sql`
DB_USER	Username to connect to the MySQL database.	`majesticflame`
DB_PASSWORD	Password to connect to the MySQL database.	`1234`
DB_DATABASE	Name of the MySQL database to use.	`ccio`
SHINOBI_UPDATE	Whether to pull updates from git when the container starts.	`false`

Script Failing? Run this.

```
apt install dos2unix -y && dos2unix entrypoint.sh && chmod +x entrypoint.sh && dos2unix setup_and_run.sh && chmod +x setup_and_run.sh && bash setup_and_run.sh
```

Instalação Shinobi Docker Desktop Windows

```
docker pull registry.gitlab.com/shinobi-systems/shinobi:dev
```

<Image_ID>

```
docker run -d -p 8080:8080 <Image_ID>
```

```
docker run -d -p 8080:8080
sha256:f3f6ed55e741e260c0b975f8d89f1358018b53ab28b9887ae00e363fd2e3423c
```

```
http://localhost:8080/super
```

admin@shinobi.video

admin

Recriação de tabela Files do Shinobi Docker

ChatGPT:

- Acessar banco de dados do Shinobi em container docker:

docker exec -ti shinobi_db bash
- Dentro do container docker
mysql -uroot -p blubsblawoot ccio;

- Aparecendo o promp **"MariaDB [ccio]>"** , executar os comandos abaixo:

```
DROP TABLE IF EXISTS `Files`;

CREATE TABLE `Files` (
  `ke` varchar(50) NOT NULL,
  `mid` varchar(50) NOT NULL,
  `name` varchar(255) NOT NULL,
  `size` bigint(20) DEFAULT NULL,
  `details` longtext,
  `status` varchar(20) DEFAULT NULL,
  `end` datetime DEFAULT NULL,
  `time` datetime DEFAULT NULL,
  `timeZone` varchar(64) DEFAULT NULL,
  `deleteAfterDays` int(11) DEFAULT NULL,
  `monName` varchar(255) DEFAULT NULL,
  KEY `ke` (`ke`),
  KEY `mid` (`mid`),
  KEY `time` (`time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Sair do container com CTRL+p+q