



Общество с ограниченной ответственностью
«СИБИРСКИЕ НЕЙРОСЕТИ»

Pisets

Инструкция по установке

Оглавление

1.	Аннотация	3
2.	Предварительные условия	3
2.1.	Среда исполнения	3
2.2.	Директории, пользователи и права	4
3.	Состав поставки:	5
4.	Запуск и остановка сервиса	5
5.	Получение токена доступа	5
6.	Инструкция для разработчика	6

1. Аннотация

Документ содержит описание процесса установки программного обеспечения «Pisets» (далее – ПО), в том числе описание:

- необходимых предварительных условий;
- инструкций для запуска и остановки работы сервиса;
- получения токена доступа;
- инструкций для разработчика.

2. Предварительные условия

2.1. Среда исполнения

- Оборудование:
 - CPU: ≥ 4 ядер
 - RAM: ≥ 16 ГБ
 - SSD: ≥ 240 ГБ
 - GPU: NVIDIA ≥ 8 ГБ VRAM (требует CUDA 11+)
- Окружение:
 - ОС: Debian 12+
 - Докер (Docker Engine 24+ с GPU-поддержкой)

2.2. Директории, пользователи и права

Pisets Core Deployment File System Organization since v2.10

Host Path	Container Path	Description
/		
├─ psz		deploy
│ └─ database	/docker-entrypoint-initdb.d	DB init scripts
│ └─ {dev prd}		deployment specifications
│ └─ *.cps.yml		
│ └─ example.env		ENV config example
├─ fs		filesystem storage
│ └─ models	/root/.cache	models storage
│ └─ huggingface	/root/.cache/huggingface	
│ └─ results	/usr/src/pisets/results	user data
│ └─ templates	/usr/src/pisets/images	presentation templates
│ └─ cache	/usr/src/pisets/vol/data/flow	user data cache
├─ cfg	/usr/src/pisets/vol/cfg	configs
│ └─ main	/usr/src/pisets/env	loaded static config (will be deprecated)
│ └─ pg	/usr/share/postgresql	PG config
├─ log	/log	logs
├─ db		databases storage
│ └─ pg	/var/lib/postgresql/data/pgdata	DB disk persistency path

Pisets Core Deployment File System Organization till v2.9

/		
├─ data		
│ └─ models	/root/.cache	
│ └─ huggingface	/root/.cache/huggingface	
│ └─ results	/usr/src/pisets/results	
│ └─ images	/usr/src/pisets/images	
│ └─ flow	/usr/src/pisets/vol/data/flow	
├─ cfg	/usr/src/pisets/vol/cfg	
│ └─ main	/usr/src/pisets/env	
│ └─ pg	/usr/share/postgresql	
├─ log	/log	
├─ database	/docker-entrypoint-initdb.d	
├─ db		
│ └─ pg	/var/lib/postgresql/data/pgdata	
├─ fs		
│ └─ srd	/d	

3. Состав поставки:

1. Образ контейнера в формате `docker image` в архиве `tar.gz`
2. Спецификация развертывания в формате `docker compose`:
 - сервис
 - агент платформы наблюдаемости*
 - агент сбора логов*
3. Пример конфигурации переменных окружения `.env`
4. Скрипт автоматизации запуска `script_to_run`

** опционально в зависимости от пакета поставки*

4. Запуск и остановка сервиса

Одна команда для полного развертывания.

Для запуска в контейнере необходимо ввести команду, с ним также запустится агент для перехвата логов и сервис для мониторинга логов `victoria-logs`:

```
./script_to_run start
```

Для создания папок, которые мапятся в контейнер необходима ввести команду:

```
./script_to_run initialize
```

5. Получение токена доступа

Чтобы начать работу с токенами, сначала нужно отправить POST-запрос на адрес `/auth/oauth/v1/token`, указав в заголовках `Content-Type: application/json`

В теле запроса нужно передать JSON со следующими параметрами:

- `client_id` = ваш_client_id
- `client_secret` = ваш_client_secret
- `grant_type` = `client_credentials`

После успешного запроса вернется пара токенов:

- access_token – используется для доступа к API
- refresh_token – используется для обновления доступа к API

Длительность действия access_token один час. Когда access_token истечет, нужно использовать refresh_token, чтобы получить новый. Для этого необходимо отправить POST-запрос на тот же адрес /auth/oauth/v1/token с заголовком Content-Type: application/json и телом в формате JSON, содержащим:

- client_id = ваш_client_id
- refresh_token = полученный_ранее_refresh_token
- grant_type = refresh_token

Сервер вернет новый access_token, оставив refresh_token без изменений.

При каждом обращении к защищённым методам API необходимо добавлять заголовок Authorization: Bearer полученный_access_token.

ВАЖНО: Храните refresh_token надёжно, так как его компрометация может привести к несанкционированному доступу.

6. Инструкция для разработчика

Для скачивания моделей нужно ввести команду:

```
python3 download_models.py ru
```

Если модели скачаны, то их нужно разместить в папке `vol/data/models/huggingface/hub`

Изображение с логотипом для необходимо разместить в папку `vol/data/images`

Если нужно полностью пересобрать или собрать образ контейнера с нуля:

```
./script_to_run build_all
```

Если нужно пересобрать только код сервиса:

```
./script_to_run build_edit_code
```