



VisionLabs
MACHINES CAN SEE

VisionLabs FaceStream

System requirements

v.5.2.0

Contents

1	General requirements	3
2	Processors requirements	3
2.1	CPU	3
2.2	GPU	4
3	Requirements for LP services and third party applications	4

1 General requirements

FaceStream is delivered in Docker containers. FS Docker container images are required for installation. An Internet connection is required to download the Docker images to the server, or the images must be downloaded to any other device and transferred to the server. Authorization to the VisionLabs Docker registry is required to download Docker images.

FaceStream can be deployed:

- by manually launching containers using Docker
- using the Docker Compose script

The distribution package contains examples of the Docker Compose script.

Running FaceStream has been tested on the following versions of the above software:

- Docker: 20.10.8
- Docker Compose: 1.29.2

It is not guaranteed to work with other versions of the above software.

FaceStream and LUNA Streams service can be run on any OS that supports Docker. Running Docker containers is officially tested on CentOS 7/8. All procedures in the installation manual are described for CentOS 7. Running on other operating systems may have some peculiarities not described in this documentation.

FaceStream and LUNA Streams Docker containers use CentOS 8.

FaceStream can use both CPU and GPU processors.

2 Processors requirements

Below are the requirements to launch FaceStream in a minimal configuration. System requirements for the production system are calculated based on the intended system load.

2.1 CPU

The following minimum requirements are given per FaceStream instance.

For the application to work correctly, the hardware must meet the following minimum requirements:

- 2 GHz or faster processor;
- 4 Gb RAM or higher;
- 10 Gb available hard disk space.
- Access to the Internet (for containers and additional software download).

Hardware requirements can be affected by several factors:

- Number of video streams;
- Frame frequency and resolution of video streams;
- FaceStream settings. The default settings are the most versatile. Depending on the operating conditions of the application, using their values can affect the quality, or performance.

Hardware should be selected based on the above factors.

FaceStream can also work in the computation speedup mode due to usage of video card resources or AVX2 instructions. CPU with AVX2 support is required. The system automatically detects available instructions and runs best performance.

2.2 GPU

GPU calculations are supported for FaceDetV3 only. See “defaultDetectorType” parameter in the FaceEngine configuration (“faceengine.conf”).

A minimum of 6GB or dedicated video RAM is required. 8 GB or more VRAM recommended.

Pascal, Volta, Turing architectures are supported.

Compute Capability 6.1 or higher and CUDA 11.4 are required.

The recommended NVIDIA driver is r470.

Now only one video card is supported per FaceStream instance.

3 Requirements for LP services and third party applications

FaceStream requires LUNA PLATFORM components, additional databases, and the LUNA Streams service. Basic information about this software is contained in this document.

LUNA Streams is not a component of the LUNA PLATFORM.

The following LUNA PLATFORM components are used by default with FaceStream:

- **LUNA Licenses** is used to license the LUNA Streams service.
- **LUNA Configurator** is used for quick access to the basic FaceStream settings and LUNA PLATFORM service settings.
- **PostgreSQL** is used as the default database for the LUNA Streams service. It is also possible to use an Oracle database instead of PostgreSQL.
- **InfluxDB** is used for monitoring. If necessary, monitoring can be disabled.

The following database versions are recommended for use with LUNA Streams:

- **PostgreSQL:** 16
- **Oracle:** 21c

To upload settings to the LUNA Configurator service, **Python version 2.x or 3.x** is required.

Installation and configuration of Oracle is not described in this manual. Further in the document, examples of launching using PostgreSQL will be given.

Balancers (for example, Nginx) and other software can be used when scaling the system to provide fail-safety. Their configuration is not described in this document.