



picturesafe-search Acceleration Server Installation Guide

Produkt Version 1.0.1 / 06.07.2020

Guide Version 1.0.1 / 06.07.2020

Table of contents

1. About This Manual	2
1.1. Contents of this manual	2
1.2. Tasks not described in this manual	2
1.3. Assistance from the support team	2
2. Installation and system overview	3
2.1. Version	3
2.2. System components	3
2.3. Installation requirements	3
2.3.1. Installation environment	3
2.3.2. Installation data	4
2.4. Password generation	4
2.4.1. Docker based configuration	4
2.4.2. JVM based configuration	5
3. Docker based installation	6
3.1. Provision of Docker and docker-compose	6
3.2. Service definition	6
3.2.1. picturesafe-search Acceleration Server	6
3.2.2. Elasticsearch	7
3.2.3. Cerebro	8
3.2.4. Kibana	8
3.2.5. Prometheus	8
3.2.6. Grafana	9
3.3. Initial deployment of the system	9
3.4. Access to the services	9
4. JVM based installation on a Linux machine	11
4.1. Java Runtime Environment	11
4.1.1. Installation	11
4.1.2. Check	11
4.2. Elasticsearch	11
4.2.1. Installation	11
4.2.2. Configuration of the Elasticsearch process	11
4.2.3. Configuration of the Elasticsearch Java memory	12
4.2.4. Control of the Elasticsearch node	12
4.2.5. Check	13
4.3. Installation of the <i>picturesafe-search Acceleration Server</i>	14
4.3.1. Installation	14
4.3.2. Configuration	14
4.3.3. Start the server	14
4.3.4. Check	15

1. About This Manual

Installation Guide Version 1.0.1 from 06.07.2020

Copyright © 2020 picturSAFE media/data/bank GmbH

All rights reserved.

This manual may not be copied, transmitted or archived in whole or in part, either electronically or mechanically, by recording or in any other form, without the prior written permission of picturSAFE media/data/bank GmbH. Please note that the contents of this manual are protected by copyright laws and that this also applies if the manual is not distributed with software that includes an end user license agreement. The contents of this manual are for information purposes only and are subject to change without notice.

This manual uses registered trademarks, trade names, common names, hardware and software designations. Even if these are not marked as such, the corresponding protection provisions apply.

1.1. Contents of this manual

This manual describes the installation of the product *picturSAFE-search Acceleration Server* in a Docker environment and on a Linux x86-64 system. The system also runs on the Windows and macOS operating systems. Administrative knowledge of the operating system and Java is required.

This manual covers the following topics:

- Overview of the installation
- Prerequisites and preparatory work
- Installing the third-party software
- Installation and configuration of the *picturSAFE-search Acceleration Server*

1.2. Tasks not described in this manual

This manual does not describe the following topics:

- Sizing and advanced deployment options
- Installation and configuration of
 - the operating system
 - an Elasticsearch cluster consisting of several nodes
 - an upstream proxy server

1.3. Assistance from the support team

For further support, please contact the picturSAFE media/data/bank GmbH support team at support@picturSAFE.de [mailto:support@picturSAFE.de].

2. Installation and system overview

The system can be installed in various ways, this manual describes the following two:

- the Docker based installation
- the JVM based installation

We recommend the installation in a Docker environment.

2.1. Version

The software is delivered with a version number in the form <major>.<minor>.<patch>. The names of the installation files always contain the version number as suffix.

This manual refers to version 1.0.0 of the product.

If errors have been corrected, the patch version of the delivered software may differ from the version mentioned in the manual. In this case, references and commands that contain a version number must be adapted to the version actually delivered.

2.2. System components

The *picturesafe-search Acceleration Server* requires a minimum of resources:

- a Java 11 installation as runtime environment
- an instance of the Elasticsearch server

NOTE

We recommend the use of OpenJDK 11 (LTS), which is included in the docker images used.

2.3. Installation requirements

This manual assumes that the following items are available or known to you.

2.3.1. Installation environment

You need a computer prepared with an operating system installation. This can also be a virtual machine (for example, VMWare vSphere Hypervisor).

NOTE

This manual assumes installation under CentOS 7.

The provided computer must fulfill the following minimum requirements:

- 1-2 processors with 4-12 cores, preferably fast clocked Intel processors
- 16GB main memory
- 2GB free hard disk space for installation
- additional disk space for the Elasticsearch index data
- an active network interface to provide the *picturesafe-search Acceleration Server*

- access to the Internet to download installation components

For the initial work steps, you need administrative access as operating system user *root*.

2.3.2. Installation data

You need an installation package of the *picturSAFE-search Acceleration Server*, consisting of:

- this installation manual
- the archive *picturSAFE-search-acceleration-server-1.0.1.tgz*

Unpack the archive in the installation folder with the command:

```
> cd path-to-installation-directory
> tar xzf picturSAFE-search-acceleration-server-1.0.1.tgz
```

The archive is automatically unpacked into a new directory *picturSAFE-search-acceleration-server-1.0.1*. This is referred to as *base directory* in this manual.

2.4. Password generation

To use the *picturSAFE-search Acceleration Server*, passwords must be set for the predefined users *admin*, *writer* and *reader*. These passwords must be entered in the configuration file in encrypted form using BCrypt.

To generate the encrypted passwords, you can use an Online BCrypt Service (e.g. <https://bcrypt-generator.com>) or use the supplied tool *encpasswd-1.0.1*.

2.4.1. Docker based configuration

In the Docker based installation, a Java Runtime Environment is included in the Docker image. The password tool *encpasswd-1.0.1* is therefore provided by the Docker image, too:

```
> docker run --rm \
    picturSAFE/acceleration-server:1.0.1 \
    encode my-password \ ①
    | sed 's/\$/\$\$/g' ②
    $$2a$$10$$18R5RrRrRTTzMGV.6fXQneU8Qd1t7LmiVMgX1jLr5LxySDxqMIMBa
```

① Replace *my-password* with the desired password.

② By using *sed*, in this example the escaping of \$ is automated.

Please enter the encoded passwords in the file *docker/docker-compose.yml*:

```
server:
  image: picturesafe/acceleration-server:1.0.1
  ...
  environment:
    - "AUTHENTICATION_USER_ADMIN_PASSWORD=encoded-password"    ①
    - "AUTHENTICATION_USER_WRITER_PASSWORD=encoded-password"    ②
    - "AUTHENTICATION_USER_READER_PASSWORD=encoded-password"    ③
```

- ① Replace *encoded-password* with the BCrypt-encoded password of the admin user.
- ② Replace *encoded-password* with the BCrypt-encoded password of the writer user.
- ③ Replace *encoded-password* with the BCrypt-encoded password of the reader user.

NOTE

Please note that the \$ character within the docker-compose.yml environment information must be escaped with the \$ character. So the sequence ABC\$EFG becomes ABC\$\$EFG. This is already implemented in the call examples with the tool sed.

2.4.2. JVM based configuration

Here the password tool is started from the base directory via a Java call:

```
> java -jar lib/encpasswd--1.0.1.jar my-password    ①
$2a$10$XRwCpeC9sH60ZbI/4uLX60MQaIW2qanrz8nfwEo8SRWfQa0BWrK10
```

- ① Replace *my-password* with the password of your choice.

After you have created the three passwords, please set them within the *config/application.yml* file:

```
authentication:
  [...]
  user:
    admin:
      username: username    ①
      password: encoded-password    ②
    writer:
      username: username    ①
      password: encoded-password    ②
    reader:
      username: username    ①
      password: encoded-password    ②
```

- ① Specification of the user name for *admin*, *writer* and *_reader*
- ② Specification of the password encoded with BCrypt for *admin*, *writer* and *reader*

3. Docker based installation

3.1. Provision of Docker and docker-compose

If the system is to be operated in a Docker environment, please install

- Docker, see <https://docs.docker.com/get-docker/> [Get Docker]
- docker-compose, see <https://docs.docker.com/compose/install/> [Install docker-compose]

After the installation you can check the versions:

```
> docker --version
Docker version 19.03.8, build afacb8b

> docker-compose --version
docker-compose version 1.25.5, build 8a1c60f6
```

If you are not using Linux, be sure to allocate sufficient processor and memory resources to the Docker installation.

3.2. Service definition

The *docker/docker-compose.yml* file contained in the base directory defines the following services and volumes.

3.2.1. picturSAFE-search Acceleration Server

The docker container *acceleration-server* provides the *picturSAFE-search Acceleration Server*.

It connects to the Elasticsearch node via the internal network and provides the GUI and the REST interface.

```
acceleration-server:
  image: docker.picturSAFE/acceleration-server:1.0.1 ①
  ports:
    - 8080:8080 ②
    - 8081:8081 ③
  environment:
    - "SERVICE_JAVA_OPTS=-Xms512m -Xmx512m" ④
    - "ELASTICSEARCH_HOSTS=elasticsearch:9200" ⑤
    - "AUTHENTICATION_JWT_SECRET=super-duper-secret-sauce" ⑥
    - "AUTHENTICATION_JWT_SIGNATUREALGORITHM=HS512"
    - "AUTHENTICATION_USER_ADMIN_USERNAME=admin"
    - "AUTHENTICATION_USER_ADMIN_PASSWORD=$$2a$$10$$/ ... trriy" ⑦
    - "AUTHENTICATION_USER_WRITER_USERNAME=writer"
    - "AUTHENTICATION_USER_WRITER_PASSWORD=$$2a$$10$$ ... M14C" ⑦
    - "AUTHENTICATION_USER_READER_USERNAME=reader"
    - "AUTHENTICATION_USER_READER_PASSWORD=$$2a$$10$$ ... 9eMG" ⑦
```

① The Docker Image is obtained here directly from DockerHub.

- ② The GUI is available on container port *8080*. It can be accessed locally at <http://localhost:8080>.
- ③ The Management API is available on container port *8081*, locally accessible at <http://localhost:8081/actuator>.
- ④ Adjust the memory allocation for your deployment scenario. You can also transfer additional Java command line parameters here
- ⑤ Configures the Elasticsearch URL.
- ⑥ Define a unique value for each installation.
- ⑦ Specify the passwords to be used in BCrypt encoding.

3.2.2. Elasticsearch

The Elasticsearch image is modified by a DockerFile definition, that the *ICU Analysis Plugin* is included.

The ICU Analysis plugin integrates the Lucene ICU module into Elasticsearch, adding extended Unicode support using the ICU libraries, including better analysis of Asian languages, Unicode normalization, Unicode-aware case folding, collation support, and transliteration.

Please note that the configuration with only one Elasticsearch node is not recommended for productive operation. For this we recommend operation on at least three nodes.

```
elasticsearch:
  build: elastic-7 ①
  ports:
    - 9200:9200 ②
  volumes:
    - elastic_data:/usr/share/elasticsearch/data ③
  environment:
    - cluster.name=picturesafe-search
    - node.name=node-0
    - cluster.initial_master_nodes=node-0
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m" ④
  ulimits:
    memlock:
      soft: -1
      hard: -1
```

- ① The docker image is built here from the *docker/elastic-7* directory to install the required ICU Analysis Plugin.
- ② The Elasticsearch interface is available on container port *9200*. Locally available at <http://localhost:9200>.
- ③ The persisted data is located in the specified docker volume.
- ④ Configures the Elasticsearch Java call parameters.

3.2.3. Cerebro

With the help of the open source tool <https://github.com/lmenezes/cerebro> [Cerebro], the Elasticsearch server can be controlled and administered via a graphical user interface.

```
cerebro:
  image: lmenezes/cerebro:0.9.2
  ports:
    - 9100:9000 ①
```

① The Docker Image provides the GUI on port 9000, locally accessible at <http://localhost:9100>.

3.2.4. Kibana

<https://www.elastic.co/de/kibana> [Kibana] is a component of the Elastic stack and provides further administration and browser functionalities for Elasticsearch.

```
kibana:
  image: docker.elastic.co/kibana/kibana:7.7.1
  ports:
    - 5601:5601 ①
```

① The Docker Image provides the GUI on port 5601, locally accessible at <http://localhost:5601>.

3.2.5. Prometheus

<https://prometheus.io> [Prometheus] is a monitoring tool for storing metric data. In the supplied configuration, the operating states of the picturSAFE-search Acceleration Server are recorded every 15 seconds and used for evaluations.

```
prometheus:
  image: prom/prometheus:latest
  command:
    - '--web.external-url=http://localhost:9090'
    - '--config.file=/etc/prometheus/prometheus.yml'
    - '--storage.tsdb.path=/prometheus'
    - '--storage.tsdb.retention.time=10d'
    - '--web.enable-admin-api'
    - '--web.console.libraries=/usr/share/prometheus/console_libraries'
    - '--web.console.templates=/usr/share/prometheus/consoles'
  ports:
    - 9090:9090 ①
  volumes:
    - prometheus_data:/prometheus ②
    - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
```

① The Docker Image provides the GUI on port 9090. Locally available at <http://localhost:9090>.

② The persisted data is located in the specified Docker Volume.

3.2.6. Grafana

<https://grafana.com> [Grafana] provides evaluations of metrics. Data from both Elasticsearch and Prometheus can be processed.

```
grafana:
  image: grafana/grafana:latest
  volumes:
    - grafana_data:/var/lib/grafana
    - ./grafana/grafana.ini:/etc/grafana/grafana.ini
  ports:
    - 9300:9300 ①
  environment:
    - "GF_SERVER_HTTP_ADDR=0.0.0.0"
    - "GF_SERVER_HTTP_PORT=9300"
    - "GF_SERVER_ROOT_URL=http://localhost:9300"
    - "GF_SECURITY_ADMIN_PASSWORD=admin" ②
    - "GF_USERS_ALLOW_SIGN_UP=false"
```

① The Docker Image provides the GUI on port 9300. Locally available at <http://localhost:9300>.

② Define an administration password here.

3.3. Initial deployment of the system

Start the stack from the *docker* directory with the command:

```
> cd docker
> docker-compose [-d] -f docker-compose.yml up
```

NOTE

You can find further instructions on how to use *docker-compose* at <https://docs.docker.com/compose/>.

3.4. Access to the services

After the Docker containers have been downloaded, built, created and started, you can access the individual services.

Use the following URLs for access:

- [picturesafe-search Acceleration Server](http://localhost:8080) [<http://localhost:8080>]
User *admin*, *writer* oder *reader* (unless otherwise defined by you) with the password you have configured.
- [Elasticsearch](http://localhost:9200) [<http://localhost:9200>]
- [Cerebro](http://localhost:9100) [<http://localhost:9100>]
- [Kibana](http://localhost:5601) [<http://localhost:5601>]
- [Prometheus](http://localhost:8080) [<http://localhost:8080>]

Is normally controlled via *Grafana*.

- [Grafana](http://localhost:9300) [http://localhost:9300]

User *admin* with the password you have configured (default: *admin*).

4. JVM based installation on a Linux machine

4.1. Java Runtime Environment

To run Elasticsearch and the *picturesafe-search Acceleration Server* you need a Java 11 installation.

4.1.1. Installation

Please install version 11 of the OpenJDK on your operating system. This example shows the installation on CentOS 7:

```
> yum install java-11-openjdk-devel
```

4.1.2. Check

You can check the correct installation with the `java --version` command.

The output must contain the expected Java version 11:

```
> java --version
openjdk 11.0.7 2020-04-14
OpenJDK Runtime Environment 18.9 (build 11.0.7+10)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.7+10, mixed mode)
```

4.2. Elasticsearch

The *picturesafe-search Acceleration Server* connects to an Elasticsearch server to perform searches efficiently and quickly.

4.2.1. Installation

Install Elasticsearch according to the instructions on the <https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started-install.html#run-elasticsearch-local> [installation page] from Elasticsearch.

The *picturesafe-search Acceleration Server* system uses the [ICU] plugin, which must be installed:

```
> sudo bin/elasticsearch-plugin install analysis-icu
```

4.2.2. Configuration of the Elasticsearch process

Edit the file `/etc/elasticsearch/elasticsearch.yml`.

Please pay special attention to the following settings:

Table 1. Werte in elasticsearch.yml

Property	Value	Comment
cluster.name	picturSAFE-search	Name of the cluster. All nodes of a cluster must be configured to the same name.
node.name	node-0	Name of the node. This name appears in various overviews and must be disjunctive in the cluster.
path.data	/var/lib/elasticsearch	The path where the index data is stored. An SSD is ideal here.
bootstrap.memory_lock	true	Setting this property causes the Java HEAP memory to be locked. Make sure that the process has permission to lock the requested memory (ulimit -l unlimited).
network.host	node-0.domain.com	IPv4 address or name of the interface used for Elasticsearch
discovery.seed_hosts	["node-1.domain.com"]	List of the Elasticsearch nodes that build the cluster
index.max_result_window	500000	Maximum number of hits that Elasticsearch can return in a single query.

4.2.3. Configuration of the Elasticsearch Java memory

Edit the file `/etc/elasticsearch/jvm.options` and pay special attention to the following values:

Table 2. Werte in `jvm.options`

Property	Value	Comment
-Xms	-Xs4g	Initial Java HEAP memory. Maximum half of the available RAM of the machine. Please set identical to -Xmx.
-Xmx	-Xmx4g	Maximum Java HEAP memory. Maximum half of the available RAM of the machine. Please set identical to -Xms.

4.2.4. Control of the Elasticsearch node

The following commands are available for starting, stopping and status requests:

- `systemctl stop elasticsearch`
- `systemctl start elasticsearch`
- `systemctl restart elasticsearch`
- `systemctl status elasticsearch`

4.2.5. Check

To see if the Elasticsearch process is running, the following commands can be used.

A simple access via HTTP to the bind address and the corresponding port returns an initial status:

```
> curl localhost:9200
{
  "name" : "node-0",
  "cluster_name" : "picturesafe-search",
  "cluster_uuid" : "wccxyZ-pS_28KGnYhETp7A",
  "version" : {
    "number" : "7.7.1",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "ad56dce891c901a492bb1ee393f12dfff473a423",
    "build_date" : "2020-05-28T16:30:01.040088Z",
    "build_snapshot" : false,
    "lucene_version" : "8.5.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Via the `cat` interface, status information readable by humans can be obtained.

For a list of supported commands, query them with `/_cat`:

```
> curl localhost:9200/_cat
=^.^=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/tasks
/_cat/indices
[...]
```

Example State:

```
> curl localhost:9200/_cat/health?v
epoch      timestamp cluster  status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1552655502 13:11:42  diocontent green      1          1      4  4   0   0     0           0              -             100.0%
```

Example Nodes:

```
> curl localhost:9200/_cat/nodes?v
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.16.0.120      9          90    0    0.01    0.02    0.05 mdi      *    node-0
```

TIP

With the help of the open source tool <https://github.com/lmenezes/cerebro> [Cerebro], further information can be queried and manipulations can easily be made.

4.3. Installation of the *picturesafe-search Acceleration Server*

4.3.1. Installation

Change to the directory created when unpacking the installation archive.

```
> cd picturesafe-search-acceleration-server-1.0.1
```

4.3.2. Configuration

Table 3. Values in *application.yml*

Property	Value	Comment
server.address	0.0.0.0	The network interface of the server on which the user interface is accessible. Can also be set using the <code>--server.address=0.0.0.0</code> command line option.
server.port	8080	The port of the server on which the user interface is accessible. can also be set with the command line option <code>--server.port=8080</code> .
management.server.address	0.0.0.0	The server's management interfaces
management.server.port	8080	The server's management port

4.3.3. Start the server

Start the server with the following call:

```
> export ELASTICSEARCH_HOSTS=adresse:port
> java -Xms=512m -Xmx=512m -jar picturesafe-search-acceleration-server-1.0.1.jar
```

Alternatively use:


```
> java -Xms=512m -Xmx=512m \  
  -jar picturesafe-search-acceleration-server-1.0.1.jar \  
  --server.address=localhost \  
  --server.port=8080 \  
  --elasticsearch.hosts=adresse:port
```

4.3.4. Check

After the server has been started, you can open the interface in the browser. If you have used the parameters specified in the example, the address is: <http://localhost:8080>.

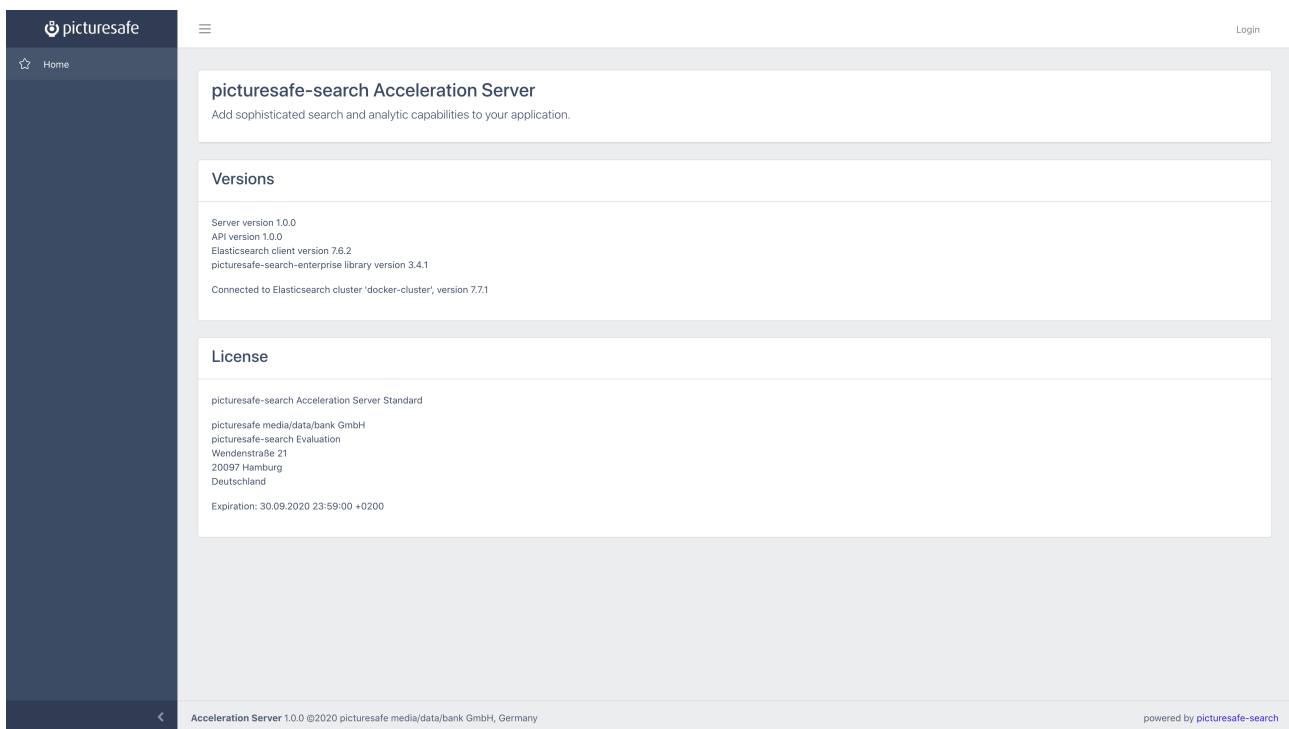


Figure 1. Startup screen