



HPSee Deployment Guide

Version 1.0.0

Tobias Gebauer, René Kraus, Frank Sonnenburg & Sophia Hönig

May 10, 2024

©2024 BioSolveIT. All rights reserved.

Contents

1	Introduction	2
2	Technical Prerequisites	3
3	Deployment Configuration	4
3.1	Installation	4
3.2	Compose Setup (Single Node)	4
3.3	Cluster Swarm Setup (Multiple Orchestrated Nodes)	5
3.4	Cleanup Configuration	5
3.5	Configuration of HTTPs Access and Certificates	6
3.5.1	Creating a Self-Signed Certificate	7
4	Admin Dashboard First Steps	7
5	Troubleshooting	8
5.1	Hosted services are not accessible from outside - similar network address for services and hosting machines	8
5.2	Connectivity Issues Between Containers in a Swarm	9
6	Further Reading, References	10

1 Introduction

All links, references, table of contents lines etc. in this pdf are clickable.

Please note that the following instructions focus on the installation of the HPSee platform.

HPSee is a computing service platform that offers high-performance computing and access to all virtual screening data in an utmost flexible manner. It is designed to be easily configurable, maintainable, and fault-tolerant, allowing convenient access to remote docking and related workflows. The foundation of the HPSee platform is Docker [1] which is free software for building, sharing, and running container applications. HPSee comes as a container, meaning all needed packages are included in an isolated environment to guarantee a smooth installation and execution. The HPSee platform consists of three main components to handle chemical computations and workloads:

- The **API** offers a REST service for accessing compute services, receiving data, and starting new computations. The API can be utilized to integrate HPSee into existing in-house workflows, to couple HPSee with data workflow tools like KNIME, or in connection with clients like SeeSAR. In the context of Docker, the compute instance or node holding the API and the database is referred to as **manager**, since it coordinates the other compute instances.
- These other compute instances, so-called **workers**, carry out the “heavy lifting”. Workers contain all needed resources to perform certain tasks independently and can be scaled throughout a cluster to maximize node usage.
- The **database** stores resulting output data as well as all information to process computations and to queue up new tasks. This yields maximum efficiency and easy access throughout a cluster. The database can be made accessible for further integrations and backup procedures.

An overview in Figure 1 shows how these components interact to enable large-scale computations from a small laptop - fast visual and easy. To achieve a straightforward deployment and scalability, HPSee is available as container images. To run these containers, the most prominent container runtime **containerd** can be utilized, which is included in Docker [1] and Docker Desktop.[2]

HPSee in connection with Docker allows two basic modes of operation:

- The **compose mode** [3] can be used to run all three main components of HPSee on a single machine. Please note that this mode does not scale up to multiple nodes. An application scenario might be a single compute resource dedicated to running chemical computations in an asynchronous manner. This resource can then easily be shared by a small workgroup and easy access to the API is provided.
- The **cluster swarm mode** [4] allows to join multiple computers or nodes to form a compute cluster. HPSee allows to scale up in case new nodes are added to a cluster, offering a very simple way to use all available compute resources. Typically, one node will host the API and database, while all other nodes host the workers for computations. This behavior can be modified and adapted to your needs, i.e. depending on use cases and a given cluster setup. This is the recommended setup for workgroups that need a simple way to bundle available resources.

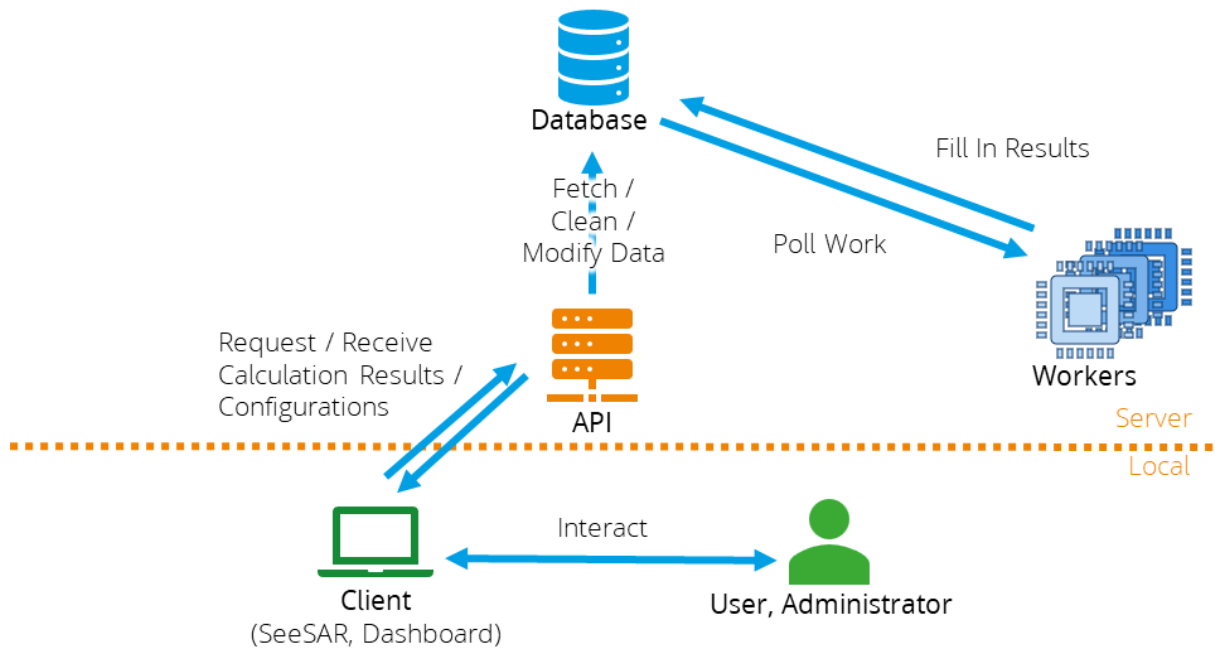


Figure 1: Overview of the HPSee components interacting with each other.

2 Technical Prerequisites

The herein-mentioned requirements focus on the installation of HPSee in compose or in swarm mode. Client software, e.g. SeeSAR, might have different requirements.

- An application supplying a **container runtime**, for example
 - Docker Desktop [2] on Windows systems
 - Docker Engine, [5] mainly on Linux systems
- **Operating system**
 - Linux is supported. The use of a pure Linux system is recommended.
 - Windows is supported for Windows 10 (version 1903, build 18362) and all later versions since Docker requires WSL2. To check your Windows build version,
 1. windows logo key + R,
 2. type `winver`
 3. click `OK`
 - MacOS is currently not supported
- **Internet access** is required for the initial setup and start of the HPSee server. This is due to the docker images pulled during the installation. After the installation is complete, running the server **does not require internet access anymore**. Only the HPSee API server node has to be internally accessible.
- A valid **license** for the complementing tools (from `license@biosolveit.com`). The license setup instructions will come with the license that we will send out — or has already been sent out to you. The license will be set in HPSee during the setup. To renew a license, a redeployment may be necessary.

The hardware requirements vary based on workload and needs. Our flexible deployment enables you to adapt your installation of HPSee to the available compute resources.

3 Deployment Configuration

3.1 Installation

For the installation of HPSee, an installer executable is provided, named either HPSee.Installer (Linux/Unix) or HPSee.Installer.exe (Windows). A shell (Linux/Unix) or a commandline environment (Windows; e.g.: cmd.exe) is needed for the initial setup to run the installer.

By running the installer, you will be guided through the installation process. The installation process offers several options like the specification of ports or the supply of license files. After the installation is complete, a **yaml** file is generated to deploy HPSee containers suitable for either compose or swarm mode, depending on the options you choose. Finishing up, the installer will provide you with Docker commands with which the HPSee server may be started. In the case of swarm mode, the deployment must be done either on the manager node or from a machine with access to the manager of the swarm. [6] The installer can be started again to overwrite the **yaml** file and thereby update or adapt the server.

In case you are facing any difficulties throughout an installation or update procedure, please get in touch with us: <mailto:support@biosolveit.com>

3.2 Compose Setup (Single Node)

1. **Install Docker** on the machine on which you intend to host the API, the database, and which will carry out the computations. Please see Section 2 for further information on technical prerequisites.
2. **Check that Docker is running** by executing the command `docker version` in a command prompt. In case you receive an error message about the Docker Daemon, the Docker container runtime has not been started. Please check your Docker installation. In case you are using Docker Desktop, check the engine status of the application for any errors.
3. **Check that compose is available** by executing the command `docker compose version` in a command prompt. In case you receive an error message, compose is not available. Please check your Docker installation and the compose documentation.[3]
4. To **install HPSee**, execute the installer executable in a command prompt. During the installation process, select the **Single Node (Compose)** option. Please see Section 3.1 for details. This will generate a **yaml** file storing all configurations from the installation process.
5. To **start HPSee**, use the **yaml** file as follows. In the folder at which the **yaml** file with your configuration is located, please execute the command `docker compose -f compose.yaml up -d` which will also be suggested by the installer. Using this command, the device will pull the needed images from the BioSolveIT registry, start the required services, and set up volumes for data storage. After completion, HPSee will be up and running. Please check this by opening the Admin Dashboard in a browser of your choice at `http[s]://<hpsee_host>:<port>`. Protocol, hostname, and port may vary based on the configuration you chose in the installation process.
6. To **shut down HPSee**, execute the command `docker compose down`.

3.3 Cluster Swarm Setup (Multiple Orchestrated Nodes)

To set up a cluster consisting of multiple nodes/compute instances, we recommend the use of Docker Swarm,[4] as it offers a simple and efficient implementation of this use case. Other container orchestrators are applicable but out of the scope of this guide. Please note that the nodes, including the manager node, which are planned to operate as a swarm must be members of the same network to allow communication.

1. **Install Docker on each node/compute instance** of the cluster. Please see Section 2 for further information on technical prerequisites.
2. **Install Docker Swarm and create a manager node.** To do this, log in to the node that you want to use as the manager node to coordinate the other nodes. Cluster configuration commands can only be executed on the manager node. Logged in to the manager node, please execute the command `docker swarm init`. From the output of this command, please note the line starting with `docker swarm join` because it will be needed to join workers in the cluster in the next step. You may regenerate the command with `docker swarm join-token worker` to join nodes at any time.
3. **Join worker nodes.** For this step, please log in to each of the nodes/compute instances you would like to use as workers. On each of these nodes, please execute the command noted from the previous step. It should have the following format:
`docker swarm join --token <SWMTKN-token> <manager-ip:port>`
4. **Check the state of the cluster** by executing the command `docker node ls` on the manager node. This command will list all nodes in the cluster and their roles. Please check that all nodes are joined in the cluster as expected.
5. To **install HPSee**, execute the installer executable in a command prompt. During the installation process, select the **Swarm** option. Please see Section 3.1 for details. This will generate a **yaml** file storing all configurations from the installation process.
6. To **start HPSee**, copy the generated **yaml** file to the manager node. On the manager node, please execute the command `docker stack deploy -c compose.yaml HPSee` which will also be suggested by the installer. After completion, HPSee will be up and running. Please check this by opening the Admin Dashboard in a browser of your choice at `http[s]://<hpsee_host>:<port>`. Protocol, hostname, and port may vary based on the configuration you chose in the installation process. The service status may also be checked with `docker stack ps HPSee` from command prompt.
7. To **shut down HPSee**, execute the command `docker stack down HPSee`.

3.4 Cleanup Configuration

With the default settings of the API, calculations and their results are deleted 90 days after their initialization. This is done automatically by the system to keep the database in a clean state. The number of days to keep calculations and their results before deletion can be altered through the endpoint `/System/CleanupConfiguration` in the Swagger User Interface at the following URL: `http[s]://<hpsee_host>:<port>/swagger/`

3.5 Configuration of HTTPs Access and Certificates

Currently, HPSee offers HTTP communication only. The usage of HTTPs requires a valid server certificate and manual editing of the yaml file. A public certificate provided by your IT is recommended. For testing and/or internal use, a self-signed certificate may be considered a viable workaround. A brief explanation of how to create a self-signed certificate can be found in Section 3.5.1.

To configure HPSee suitable for HTTPs access, two lines in the `yaml` file must be added. Figure 2 shows the structure of the section to be altered. Below the tag `hpsee_api` (1) and nested below the tag `environment` (2), add the following lines (3):

```
Kestrel:Certificates:Default:Path: "/path/to/your/certificate.pfx"
```

```
Kestrel:Certificates:Default>Password: "< certificate password >"
```

Additionally, the folder containing the certificate file must be available on the machines that will host the API. For this, the folder containing the certificate must be mounted in the container (4).

The `certificate_volume` can either be a path to the folder that will be mapped to the `/httpcert` or you can create a separate Docker volume [7] containing the certificate:

```
volumes:
```

- "certificate_volume:/httpcert"

Finally, the edited section in the file should look similar to Figure 2.

```
1 hpsee_api:
  image: image_tag
  ports:
    - ${API_PORT:-8080}:443
 2 environment:
    Db:Host: "hpsee_db"
    Db:Port: 5432
 3 Kestrel:Certificates:Default:Path: "/httpcert/certificate.pfx"
    Kestrel:Certificates:Default>Password: "MyCertificatePassword"
    ASPNETCORE_URLS: "https://+;http://+"
 4 volumes:
    - "certificate_volume:/httpcert"
  depends_on:
    - hpsee_db
```

Figure 2: Structure of the `yaml` file section to be altered for HTTPs communication. The red-marked lines have to be added to the file.

3.5.1 Creating a Self-Signed Certificate

This guide offers only a limited introduction to the creation of certificates. Please refer to the official OpenSSL documentation [8] for further reference. The execution of the commands suggested below will lead to the creation of the corresponding self-signed certificate files in the current directory of the shell.

Create the private key and public certificate using the following command:

```
openssl req -x509 -newkey rsa:4096 -sha256 -keyout my.key
           -out my.crt -subj "/CN=<HOST_MACHINE>"
           -days 600
```

Please edit the command by replacing the `<HOST_MACHINE>` by the name or address of your API host. Using the flag `--days 600`, the generated files will be valid for 600 days. You will be asked to enter a PEM passphrase for the private key. This will create two files, the private key `my.key` and a public certificate `my.crt`.

To combine those two files into a single PKCS#12 (P12) bundle, please use the following command:

```
openssl pkcs12 -export -in my.crt -inkey my.key -out cert.pfx
```

You will be asked for the PEM passphrase you previously set for the private key. Next, you will be asked to enter an export password. After the command was executed, the PKCS#12 bundle will contain both your private key and your public certificate with the name `cert.pfx`.

For editing the `yaml` file, you will need the location of the `cert.pfx` file and the export password.

4 Admin Dashboard First Steps

The Admin Dashboard provides simplified access to the most routine tasks of HPSee for an administrator. Also, the Dashboard gives a quick overview of HPSee and allows certain maintenance and cleanup tasks. Having completed the HPSee installation successfully, you can access the Admin Dashboard in a browser of your choice at `http[s]://<hpsee_host>:<port>`. Protocol, hostname, and port may vary based on the configuration you chose in the installation process.

For a first login, you can use the username **admin** and the password **admin**. As a first step, please use the menu bar to navigate to the **Users** page. A picture of this page with a few hints is shown in Figure 3. Use the **+** button to create a new user. Please fill in all fields of the dialog to create a new user for yourself. In the field **Role**, please select **Admin**, since only admins can use the Admin Dashboard. As soon as you have finished, please click **CREATE USER** to complete the dialog. Now, for a secure use of HPSee, please delete the default user **admin**. To do this, please log out and log in with your newly created username and password. Navigate to the **Users** page again. Now, the **bin icon** in the table row of the default user **admin** should be enabled. Click the icon and delete the user **admin** to prevent others from using the default login. Create as many users as needed.

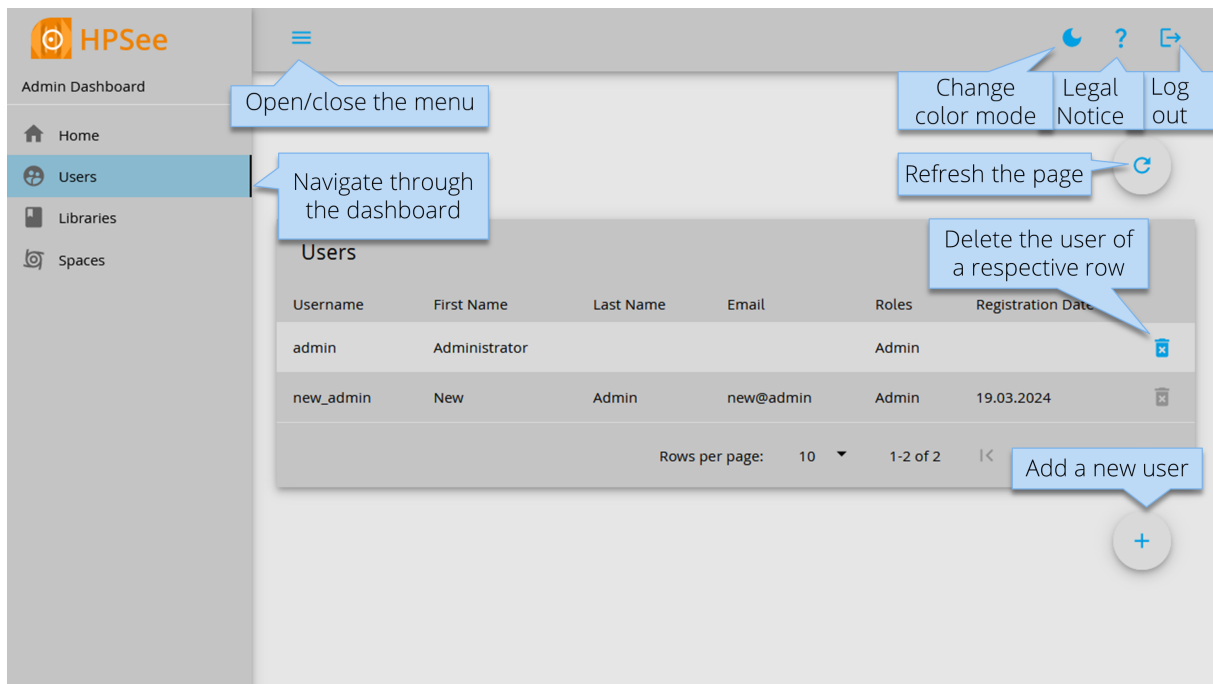


Figure 3: Users page of the Admin dashboard with a few hints to get started.

5 Troubleshooting

5.1 Hosted services are not accessible from outside - similar network address for services and hosting machines

This error occurs if the ingress network (default: 10.0.0.0/24) and the network address are similar. The ingress network is meant to enable communication between containers. The network address is utilized to link the host machines. To fix this, the established ingress network must be replaced by a network with a different network address. To do so, please follow these steps for customizing the default ingress network:

```
docker network rm ingress
```

```
docker network create --driver overlay --ingress
--subnet=<desired-subnet> --gateway=<gateway>
--opt com.docker.network.driver.mtu=1200 ingress
```

After completion, please redeploy HPSee. Please see <https://docs.docker.com/network/overlay/> for further information.

In case you are facing any difficulties that are not part of this guide, please get in touch with us: <mailto:support@biosolveit.com>; and please mention any errors or warnings that you see.

5.2 Connectivity Issues Between Containers in a Swarm

A known issue is a possible ambiguity between a network of nodes and the internal network created by Docker to enable communication between containers. The default network created by Docker is `127.17.0.0/16`. To check whether your host machines are using a similar network address, please execute `ifconfig` on Linux or `ipconfig` on Windows.

If a network device is using a similar network address, you are facing the problem addressed in this section. In case you are using Docker Swarm, none of the nodes must have the network address of the Docker network. A solution to this problem is to change the default network address pools that are created by Docker:

1. Access the `daemon.json` file. On Linux, the file location is `/etc/docker/daemon.json`. On Windows, you may alter the file in the settings of Docker Desktop at **Settings** → **Docker Engine**
2. Add the following lines to the file:

```
"default-address-pools": [  
  { "base" : "11.12.0.0/16" , "size" : 24 }  
]
```

Please ensure the given address is not already taken by another network device on any given Docker host. The above changes will configure the following settings:

- `"base" : "11.12.0.0/16"` defines the base IP range managed by Docker
- `"size" : 24` configures the subnet of all networks created by Docker in the given managed IP range. In this case, the first 3 octets define the network and the last byte defines the corresponding host addresses. With the above configuration Docker can create the network addresses `11.12.[0-255].0`.

3. Save the changes to the `daemon.json` file. For this purpose, on Linux, execute the command `sudo systemctl restart docker`. On Windows with Docker Desktop, click **Apply & Restart**.

For more detailed information, please read the corresponding Docker documentation [9].

6 Further Reading, References

Additional information about Docker is available at <https://docs.docker.com/engine/reference/commandline/cli>.

At this stage, a complementary version of **SeeSAR** as a client can be obtained on demand from Bio-SolveIT.

References

- [1] <https://docs.docker.com/>.
- [2] <https://www.docker.com/products/docker-desktop/>.
- [3] <https://docs.docker.com/compose/>.
- [4] <https://docs.docker.com/engine/swarm/>.
- [5] <https://docs.docker.com/engine/install/>.
- [6] <https://docs.docker.com/engine/context/working-with-contexts/>.
- [7] <https://docs.docker.com/storage/volumes>.
- [8] <https://www.openssl.org/docs/>.
- [9] <https://docs.docker.com/engine/swarm/swarm-mode/#configuring-default-address-pools>.

We wish you great success and much joy with HPSee!