HPSee Deployment Guide
Version 2.1

Sophia Hönig, Priya Kempanna, Tobias Gebauer, René Kraus, Frank Sonnenburg

February 7, 2025

Contents

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.

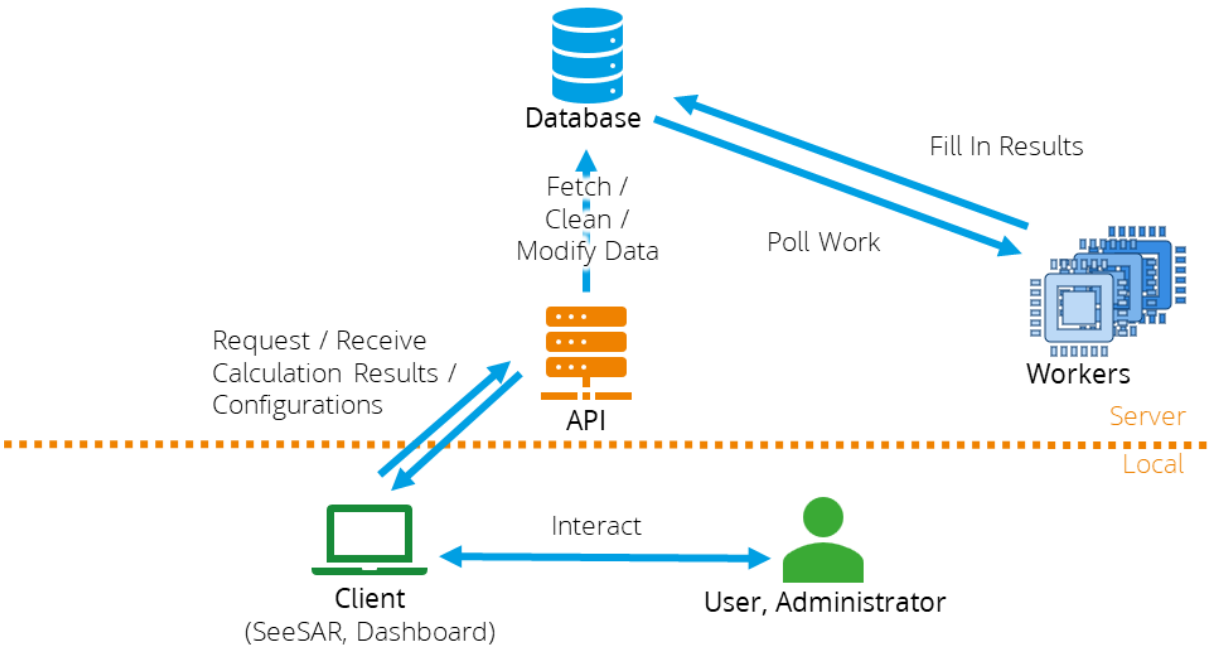

Figure 1: *Overview of the HPSee components interacting with each other.*

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
  - Podman, [6] only for Single Node (Compose) mode, see Section 3.2 for details
- **Operating system**
  - Linux is supported according to the availability of Docker Engine. 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`
  - For the use with other operating systems or hardware platforms, please contact us
- **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 additional tools (from license@biosolveit.com). The path to the license file is requested during the installation of HPSee. A new installation may be required to renew a license.

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. A shell (Linux/Unix) or a command-line interface (Windows; e.g.: cmd.exe) is needed for the initialization of an HPSee server. Please refer to Section 3.2 for single node setup or Section 3.3 for a swarm setup of multiple machines.

In case you are facing any difficulties throughout an installation or update procedure, please get in touch with us: 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. See Figure 2 for detailed instructions. This will generate a `yaml` file storing all configurations from the installation process.
5. To **start HPSee**, 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. See Figure 2 for detailed instructions. 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` as shown in Figure 4.

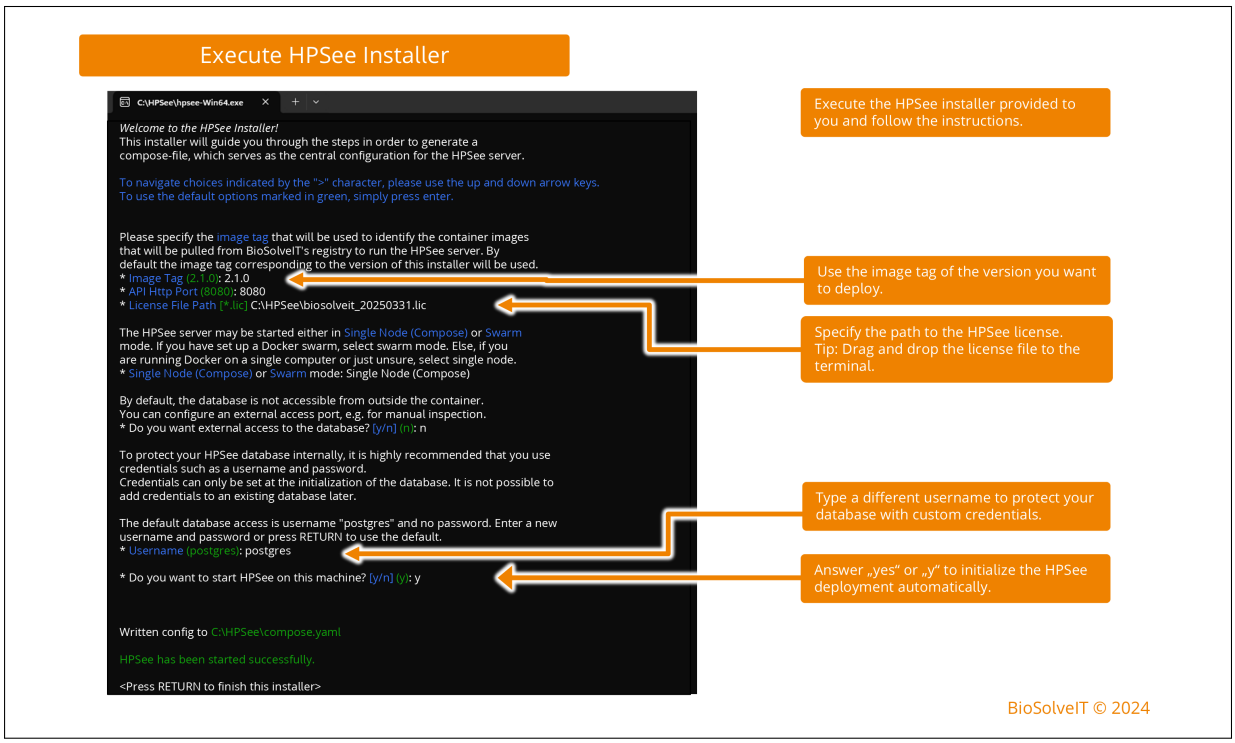

Figure 2: Initialization of deployment `yaml` file and server setup with the HPSee installer.

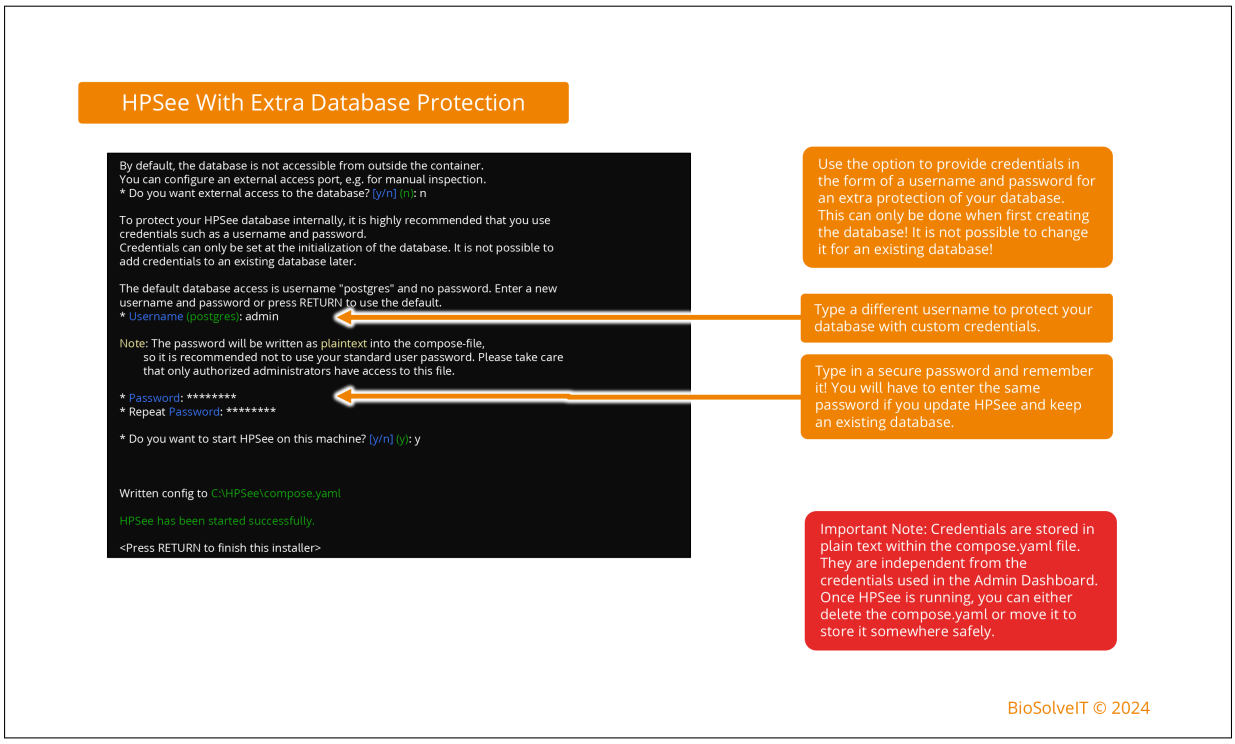

Figure 3: Server setup with the HPSee installer using custom credentials. For more information, please see Section 3.4.

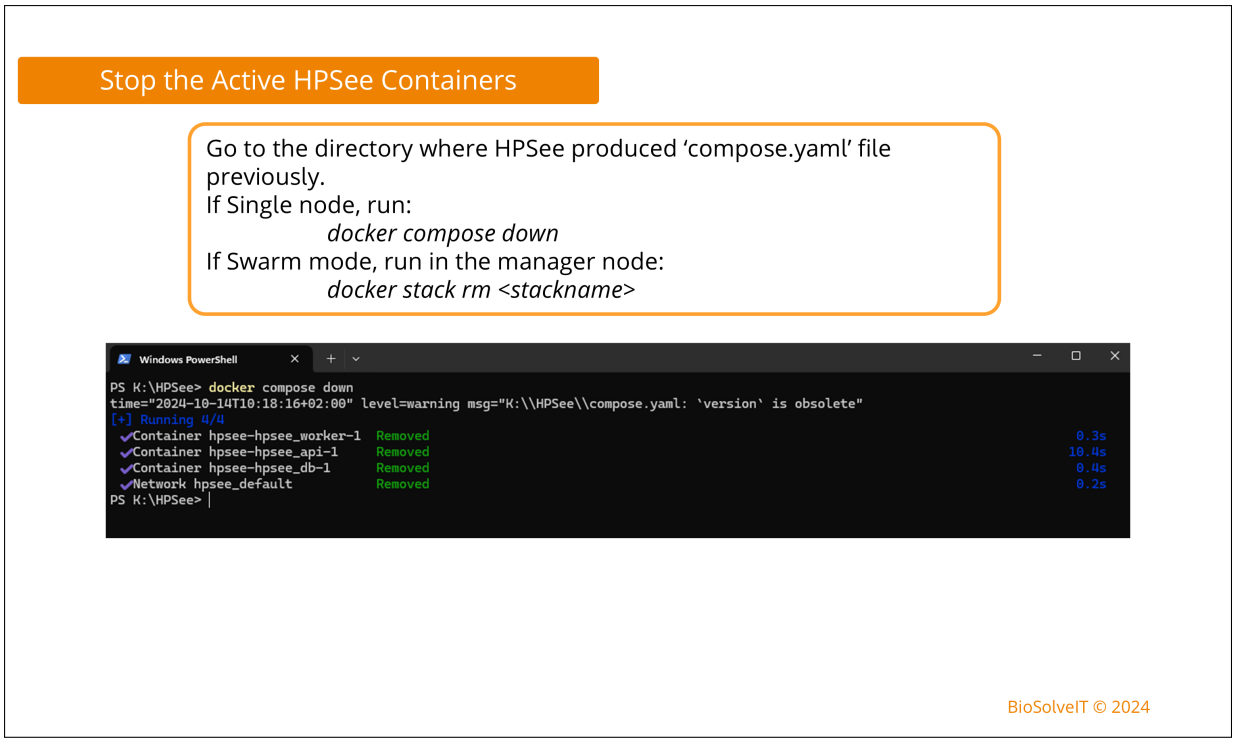

Figure 4: Shut down of a running HPSee server. The default for the `<stackname>` is set to HPSee.

3.3 Cluster Swarm Setup (Multiple Orchestrated Nodes)

To set up a cluster consisting of multiple nodes/compute instances, Docker Swarm [4] offers a simple and efficient implementation. Please note that all nodes 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. See Figure 2 for guidance, but instead of using the Single Node option, select the `Swarm` option. 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. See Figure 2 for swarm mode. 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 rm <stackname>` as shown in Figure 4. By default, the `<stackname>` is set to HPSee and may be altered individually.

3.4 Providing Credentials to Protect the Database

Login credentials in the form of a username and password to access the HPSee database are highly recommended to protect your data. Please note that these credentials differ from the credentials supplied via the API and the Admin Dashboard (Section 4) as the latter only regulates the authentication to HPSee. If no credentials are generated and external access to the database is configured, public access is possible with user `postgres` on the specified port and without the need for a password. Credentials can only be set at the initialization of the database. It is not possible to add credentials to an existing database later unless you remove the volume and start from scratch. Also, using credentials is only possible for HPSee version 2.1 or later. To generate and use credentials upon setup of HPSee, please follow the instructions shown in Figure 3. HPSee stores the username and password to access the database in plaintext within the `yaml` file and propagates this information as environment variables. Therefore, after installing HPSee successfully, please delete the `yaml` file or store it somewhere safe. The username and password must be specified upon every update of HPSee if you keep your database.

The credentials can be accessed in plain text from within the Docker containers as this is predefined via Docker. Therefore, please make sure that only designated administrators who are allowed to access these have the right to access Docker containers on your system. Especially when using a network in Swarm mode, if a more secure setup is needed, please refer to Docker Secrets [7]. You may also read Section 3.4.1 for further details.

Please store your credentials somewhere safe as there is no way to recover your credentials. Once lost, there will be no way to access an existing database set up with credentials. Please also note, that there is no way to add new credentials or change credentials for an existing database. If you try to overwrite credentials for an existing database, if you do not provide the correct credentials, or if you try to add credentials for an old database, the HPSee server will not be able to communicate with the database and your HPSee deployment will fail. Thus, please treat your credentials with extra care!

3.4.1 Using Docker Secrets Within HPSee

If you wish to use Docker Secrets for the propagation of your database credentials, please ensure your secrets are prepared for your deployment. For a deployment in Single node (Compose) mode, please refer to Reference [8]. To use secrets in Swarm mode, Reference [7] gives further reference.

Please follow these steps to set up secrets for HPSee:

1. Prepare the secrets `hpsee_db_username` for your username and `hpsee_db_pw` for your password according to the Docker documentation
2. Adapt your `yaml` file to allow all services to access these secrets in the default Docker Secret location.
3. Add the following environment variables to the PostgreSQL service to use your custom credentials on first database setup. They specify file locations as sources for the root credentials of the database. See the PostgreSQL image documentation [9] for further information on how to provide credentials.

- Username environment variable key: `POSTGRES_USER_FILE`
- Username value: `/run/secrets/hpsee_db_username`
- Password environment variable key: `POSTGRES_PASSWORD_FILE`
- Password value: `/run/secrets/hpsee_db_pw`

4. Start HPSee according to the point **start HPSee** in Section 3.2 for Single node (Compose) setup or Section 3.3 for Swarm mode, respectively.

3.5 Configuration of the Database Volume

To configure an alternate location for your HPSee database, you may configure the `compose.yaml` in one of the two ways:

1. Use different volumes: This is recommended if there is enough storage available at the location of the docker installation. Please see Figure 7 for details.
2. Use a different location: This is recommended if you need to change the data storage location to a disk with available space. Please see Figure 8 for further instructions.

3.6 Upgrade From an Older HPSee Version

To upgrade your existing HPSee deployment, please make sure there are no calculations running anymore. If your previous version was set up with credentials to protect your database, please make sure you have your username and password for your database at hand to redeploy HPSee before you begin the update. For more information about the protection of your database with custom credentials, please refer to Section 3.4.

If you upgrade to a newer version and keep your database, a data migration may be triggered automatically. Migration here means adapting the data in your database to allow for compatibility with the version you are upgrading to. While this process is running, the API will not be accessible. This means HPSee including the Admin Dashboard and all services will not be available until the migration process has finished. Please make sure to delete all data you do not necessarily need in advance because the migration time depends on the amount of data to be processed and it might highly increase the needed storage for your database by adding further information for existing data. If you are unsure about the upgrade, please contact us at support@biosolveit.com. In any case, it will be safest to make a fresh start, as described below. To upgrade HPSee, please follow these general steps:

1. **Shut down your running HPSee stack** as shown in Figure 4.
2. **Optional: To make a fresh start**, you can remove your existing database and replace it with a new one. **Caution! This means your data on the server will be erased!** Before moving forward, please save your libraries locally if you wish to keep them. See Figure 6 for further instructions.
3. **Overwrite your `compose.yaml`** by executing the HPSee installer again, requesting a new version number. See Figure 5 for details.
4. **Optional: Configure the database** to use a separate location for the new version and preserve the old version in case you need it (see Section 3.5 for details).
5. **Re-deploy** your HPSee stack as shown in Figure 2.

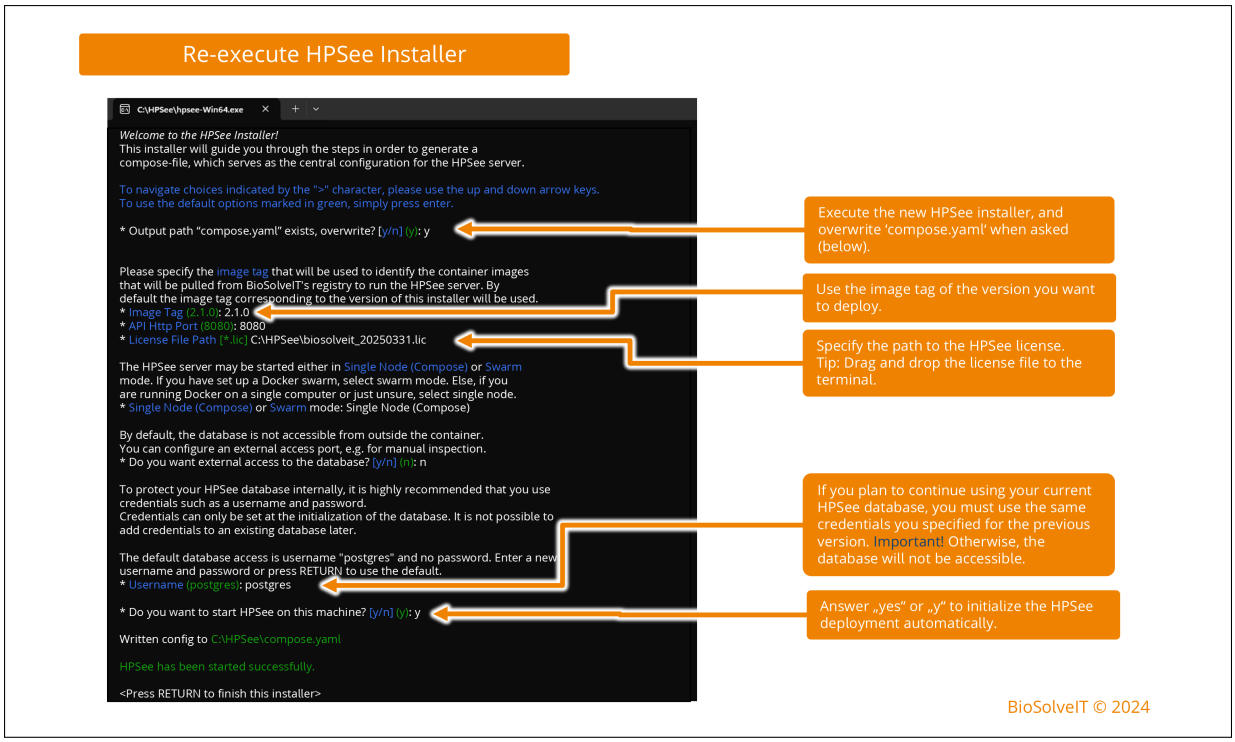

Figure 5: Overwrite an existing `yaml` file to update the HPSee version.

3.7 Cleanup Configuration

With the default settings of the API, calculations and their results are never deleted automatically. However, the automated deletion of files older than a specified number of days can be defined within the system to keep the database clean. 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/`

A specification of 0 days will lead to the automated cleanup daily at 3 am local time. A specification of -1 days prevents automated deletions like in the default setting.

3.8 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 3.8.1.

To configure HPSee suitable for HTTPs access, two lines in the `yaml` file must be added. Figure 9 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 [10] containing the certificate:

```
volumes:
  - "certificate_volume:/httpcert"
```

Stop the Active HPSee Containers and Remove Volume

Go to the directory where HPSee produced compose.yaml file previously.

(a) If Single node, run:

docker compose down

If Swarm mode, run in the manager node:

docker stack rm <stackname>

```
PS K:\HPSee> docker compose down
time="2024-10-14T10:18:16+02:00" level=warning msg="K:\\HPS
[+] Running 4/4
 ✔Container hpsee-hpsee_worker-1  Removed
 ✔Container hpsee-hpsee_api-1     Removed
 ✔Container hpsee-hpsee_db-1      Removed
 ✔Network hpsee_default           Removed
PS K:\HPSee>
```

(b) Get the volume name using:

docker volume ls

(c) Remove volume:

docker volume rm hpsee_database_volume

```
Windows PowerShell
PS K:\HPSee> docker volume ls
DRIVER    VOLUME NAME
local     hpsee_database_volume
PS K:\HPSee> docker volume rm hpsee_database_volume
hpsee_database_volume
PS K:\HPSee>
```

BioSolveIT © 2024

Figure 6: Removal of an existing database volume.

Switch to a Different Volume

Get the current volume name by running the command: *docker volume ls*. The name is 'hpsee_database_volume' in most cases.

```
PS K:\HPSee> docker volume ls
DRIVER    VOLUME NAME
local     hpsee_database_volume
```

Find `name: hpsee_database_volume` in 'compose.yaml' that you generated and rename it to anything other than itself. For example, to 'hpsee_database_volume_new' and save the changes.

```
volumes:
  database_volume:
    name: hpsee_database_volume
```

```
volumes:
  database_volume:
    name: hpsee_database_volume_new
```

Deploy the updated file as done previously.

BioSolveIT © 2024

Figure 7: Usage of different database volumes.

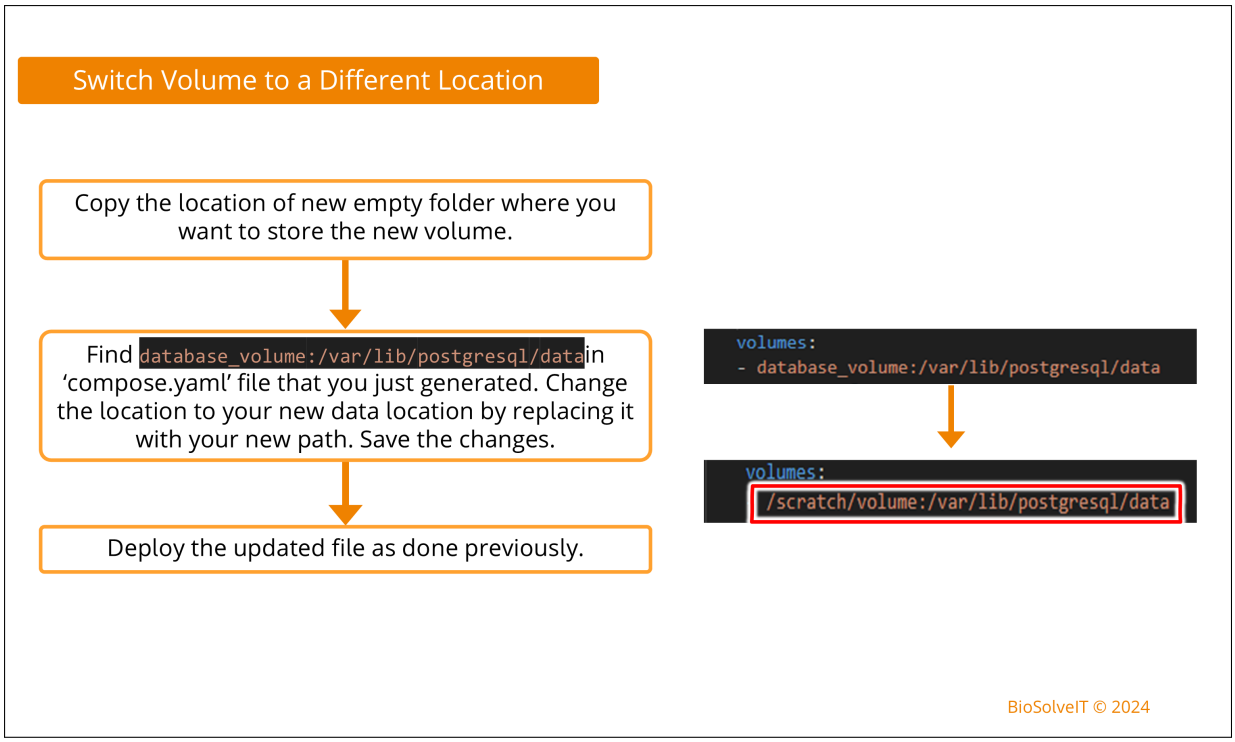

Figure 8: Usage of a different location.

Finally, the edited section in the file should look similar to Figure 9.

3.8.1 Creating a Self-Signed Certificate

This guide offers only a limited introduction to the creation of certificates. Please refer to the official OpenSSL documentation [11] 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 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

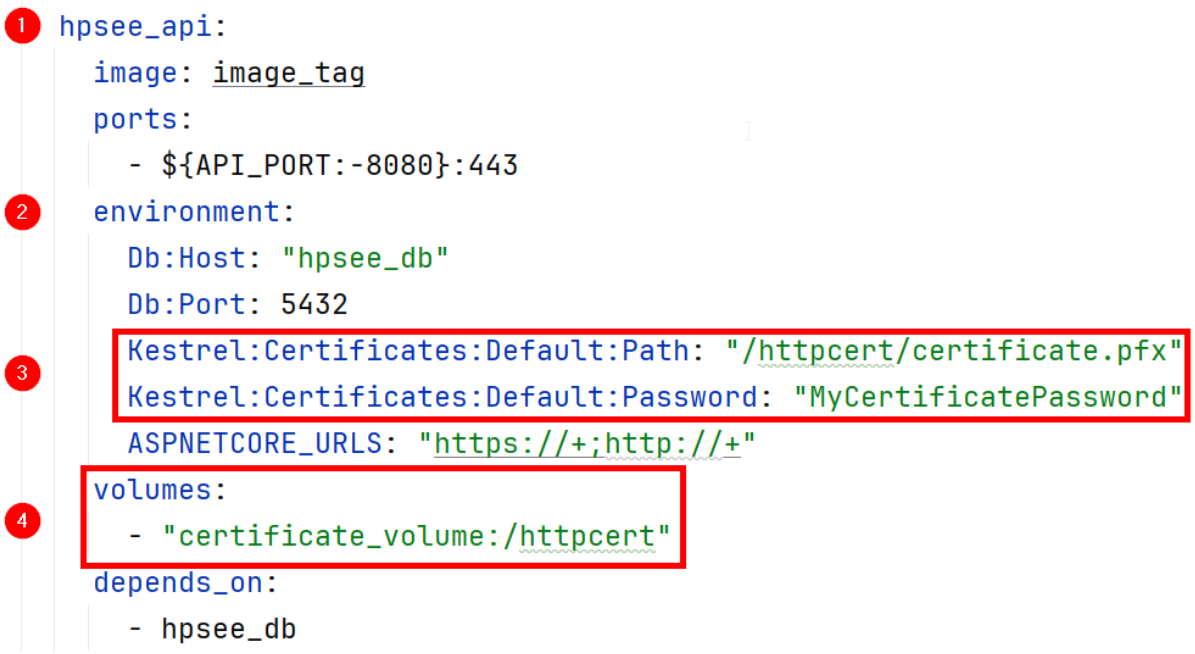

Figure 9: Structure of the `yaml` file section to be altered for HTTPs communication. The red-marked lines have to be added to the file.

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>`. Please see also Figure 10. Protocol, hostname, and port may vary based on the configuration you chose in the installation process. The port is specified during installation. To find the hostname or IP of your system, please refer to Section 6.1.

For a first login, you can use the username **admin** and the password **admin**. A comprehensive overview of the options provided within the Dashboard after login is shown in Figure 11. 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 12. Use the **+** button to create a new user as shown in Figure 13. 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. A success message will appear as shown in Figure 14.

Now, for a secure use of HPSee, please consider deleting the default user **admin**. For this purpose, please refer to Figure 15 for further details. 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.

To upload a chemical library for remote docking workflows, please follow the instructions provided in Figures 16, 17 and 18 The upload of Chemical Spaces works analogously and is shown in Figures 19 and 20. Note on the **Libraries** and **Spaces** page: The times provided in the column **Created At** are provided in UTC and may therefore differ from your local system time.

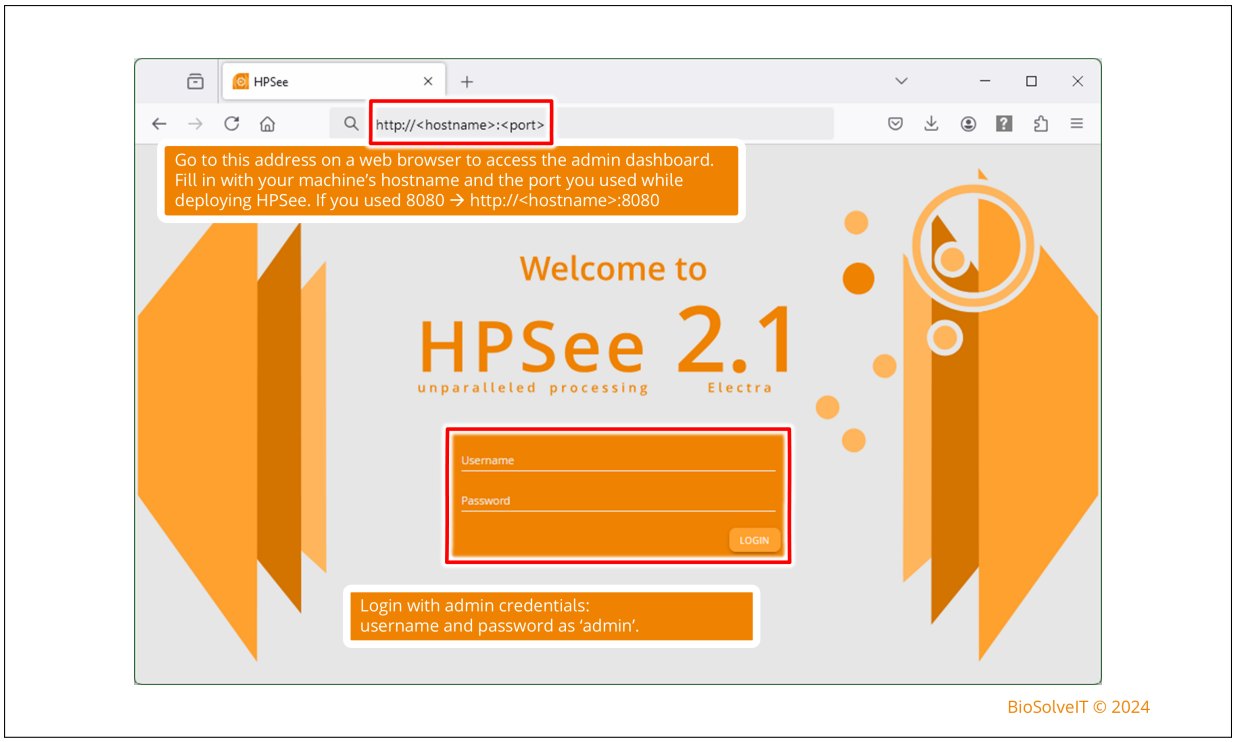

Figure 10: Access to the Admin Dashboard.

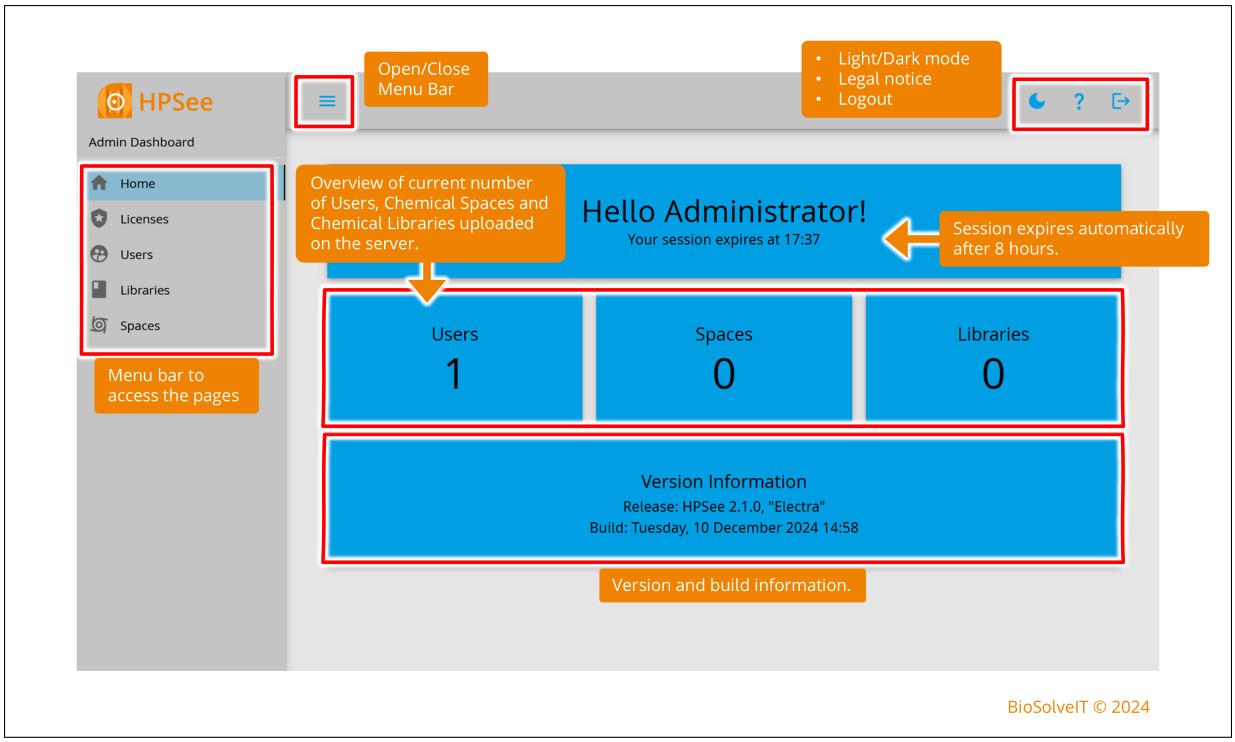

Figure 11: First steps in the Admin Dashboard.

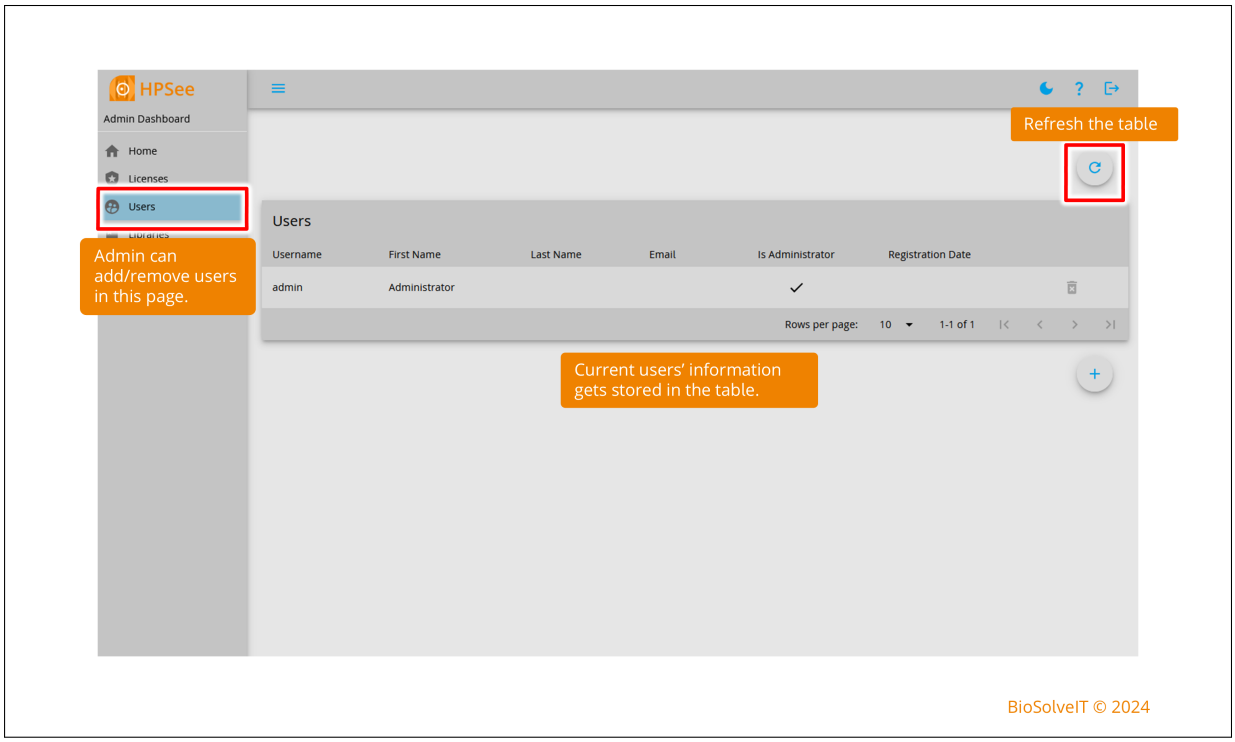

Figure 12: Users page of the Admin Dashboard.

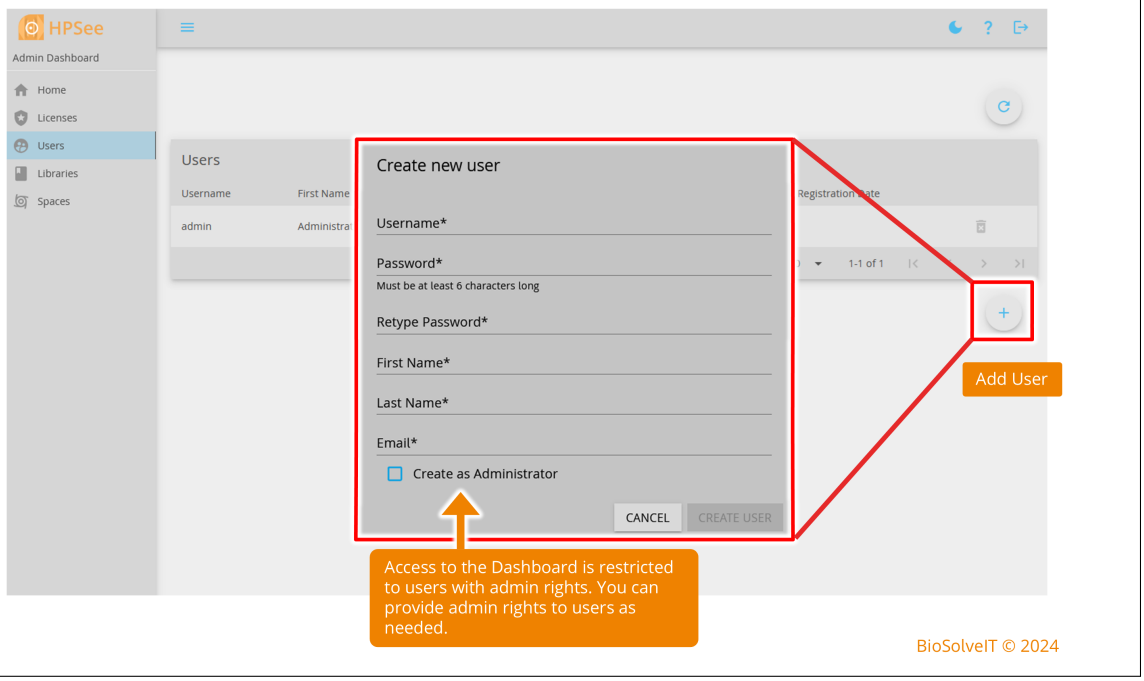

Figure 13: Dialogue to create a new user or administrator from the Admin Dashboard.

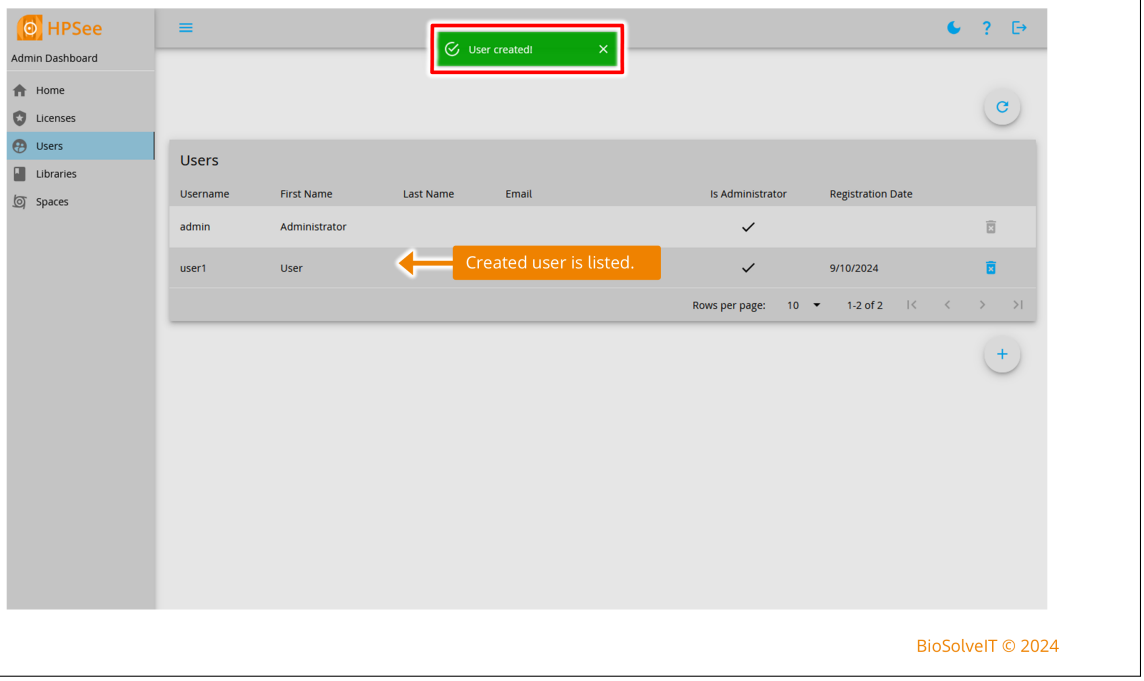

Figure 14: Success message after creating a new user.

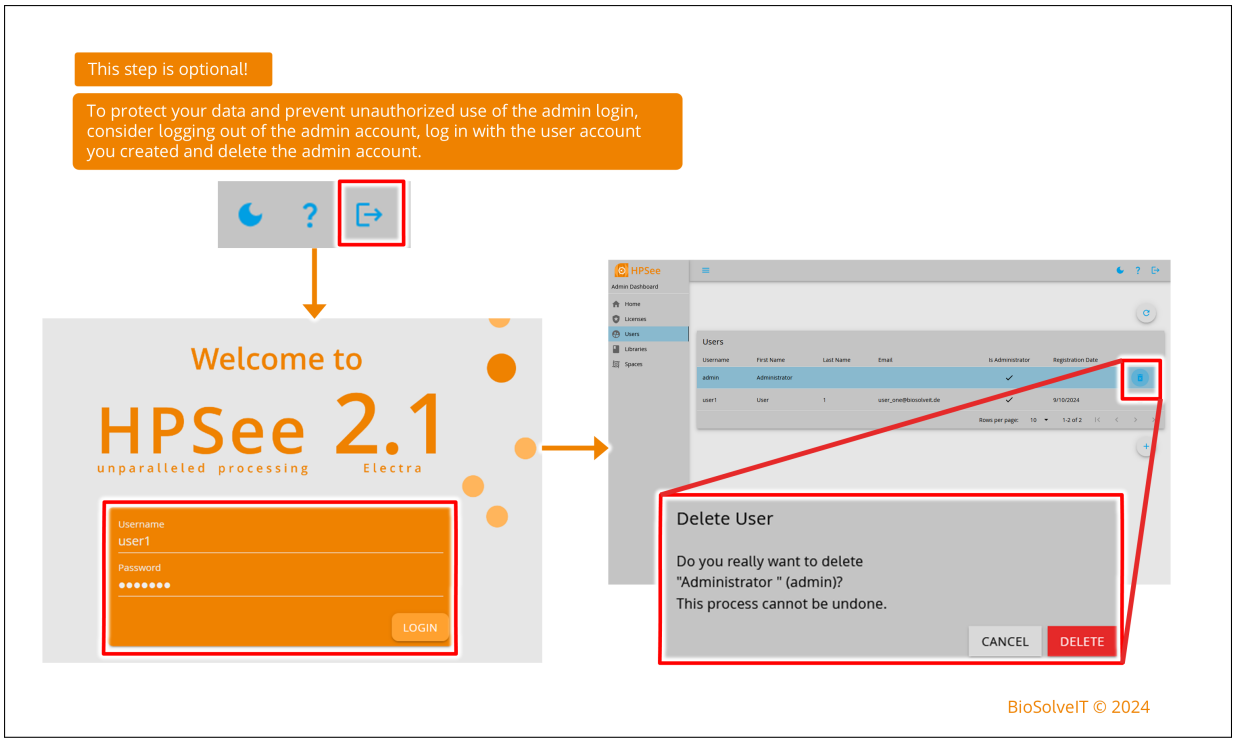

Figure 15: Deletion of the initial admin user.

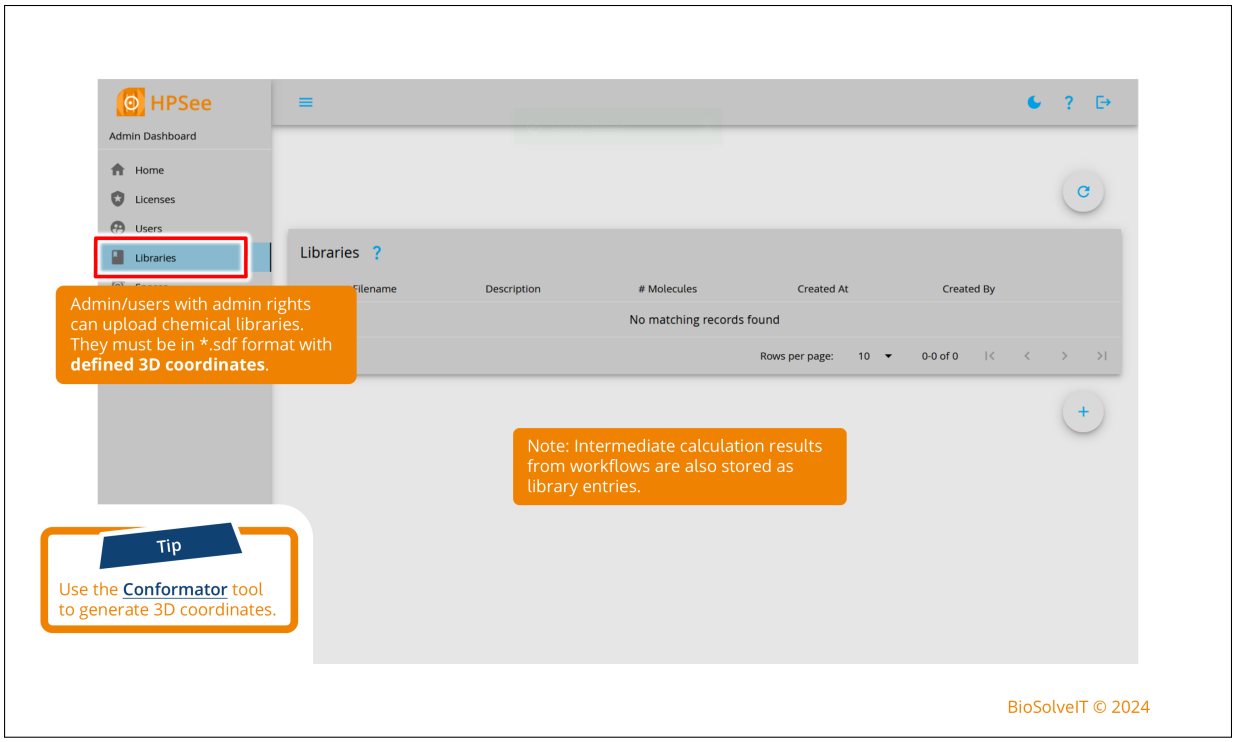

Figure 16: Overview of the libraries page.

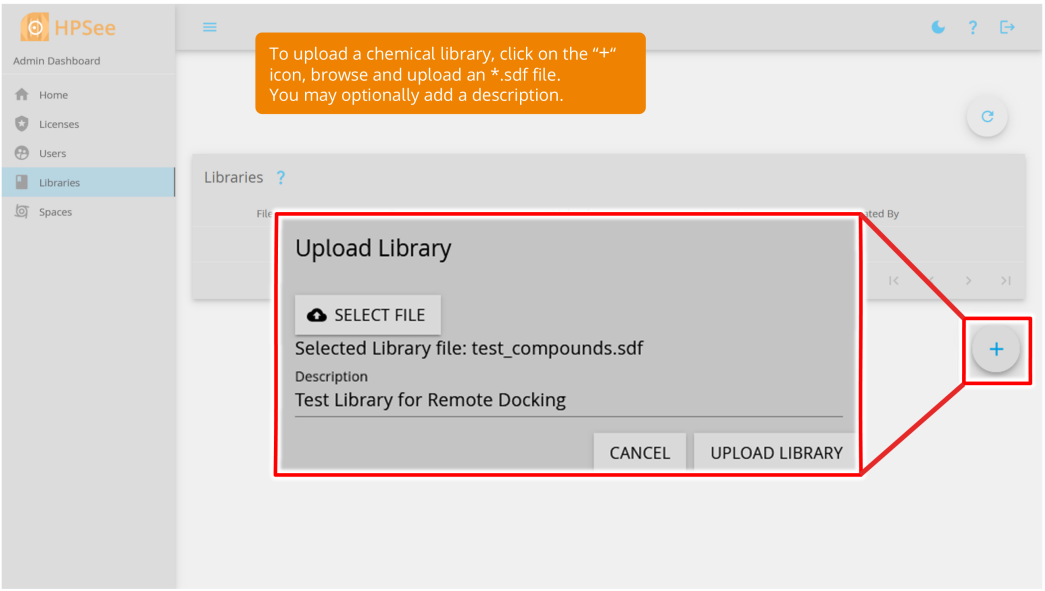

Figure 17: Upload dialogue for a chemical library.

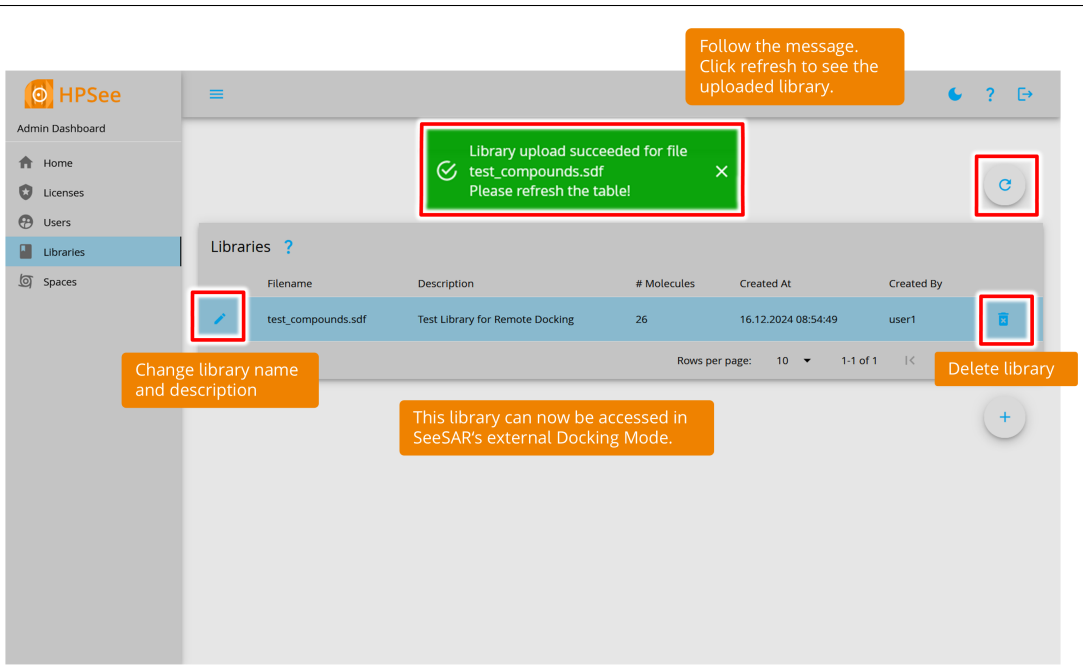

Figure 18: Successful upload of a chemical library.

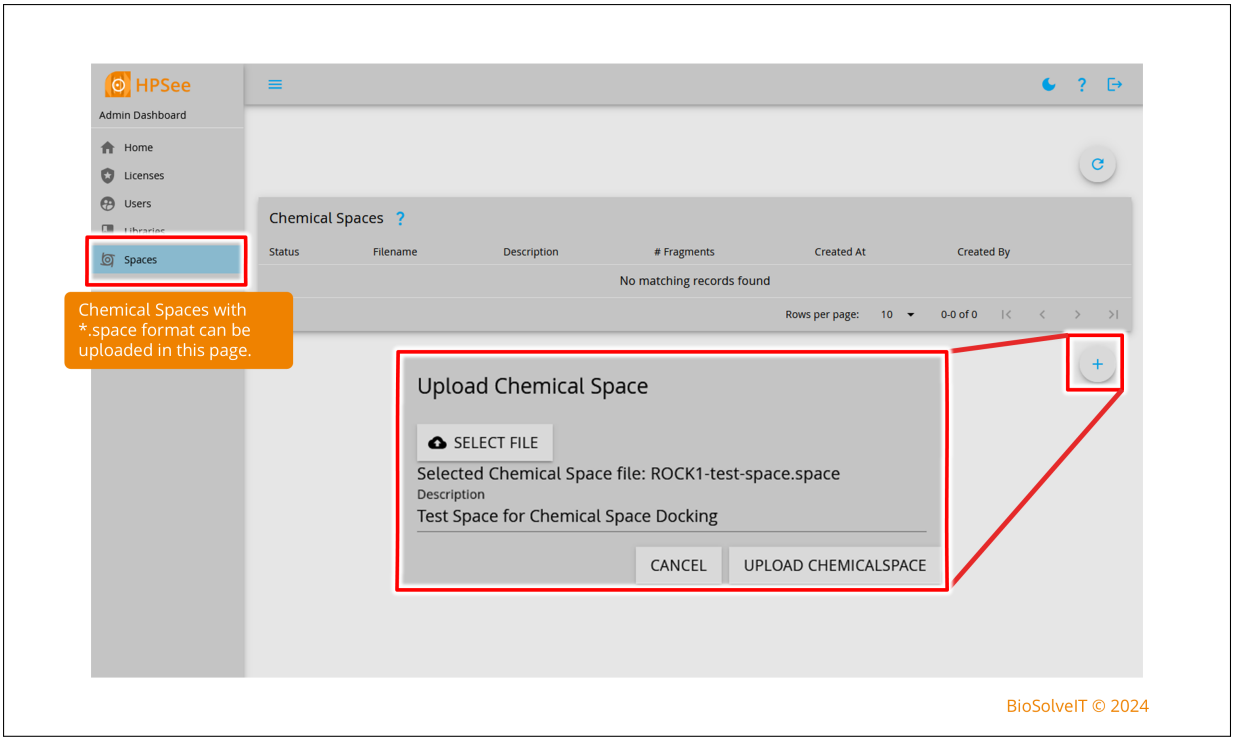

Figure 19: Upload dialogue for a Chemical Space.

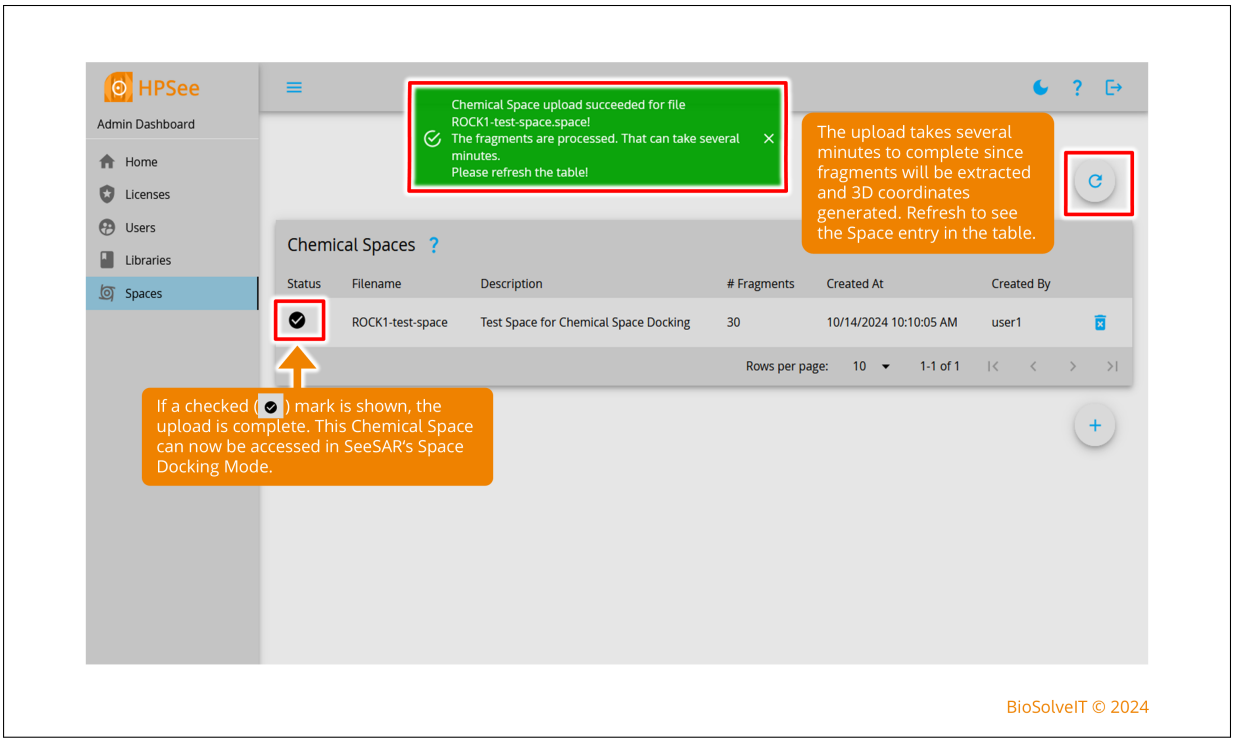

Figure 20: Successful upload of a Chemical Space.

4.1 License Status of Workflows

The license status of each workflow provided by HPSee can be checked via the **Licenses** page of the Admin Dashboard. A checkmark in the **Status** column symbolizes a valid license. Thus, the workflow named by the **Workflow Specifier** can be started from the endpoint of the HPSee API or a client like SeeSAR. If a status is marked by an exclamation mark, the license of the respective workflow is invalid, for example, because it expired or was not purchased. It is not possible to execute a workflow without a valid license. To buy a license, please get in touch with license@biosolveit.com. An example of a valid and an invalid license status is given in Figure 21.

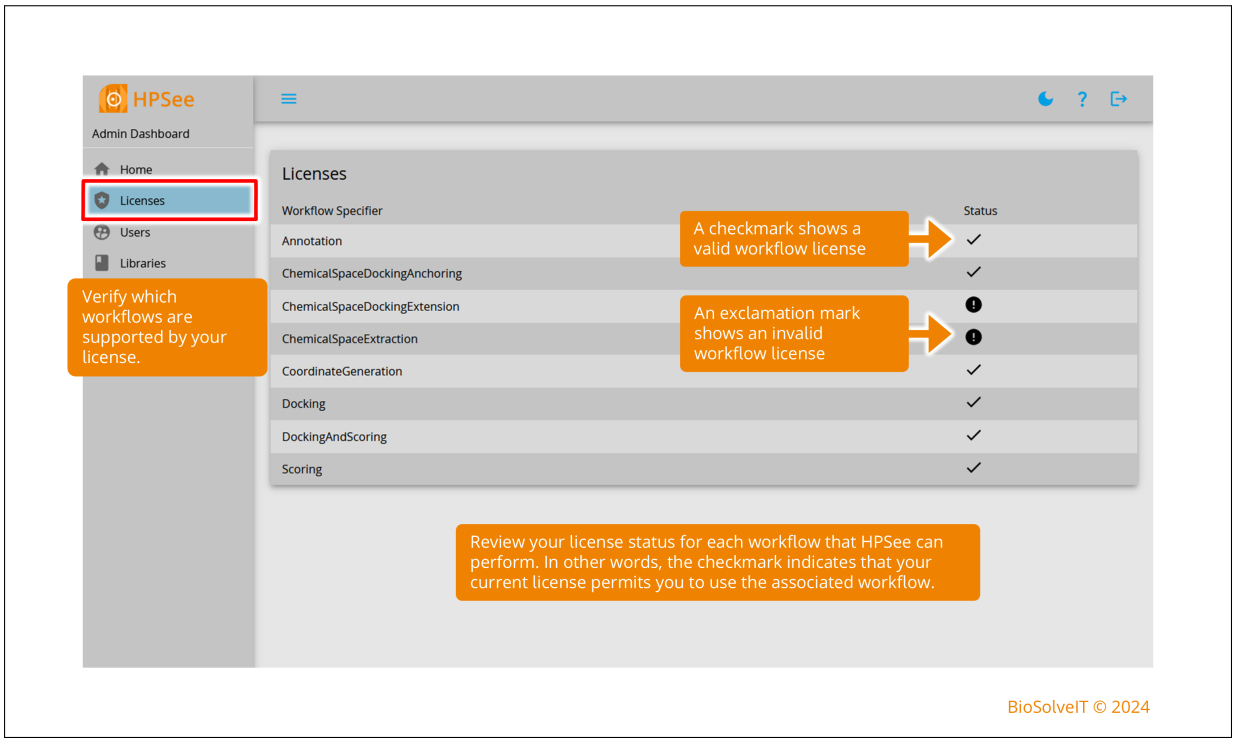

Figure 21: Licenses page of the Admin Dashboard with hints about license status symbols.

5 Advanced Usage: The Swagger API Documentation

This section is only relevant for professional computational chemists or administrators intrigued by the behind-the-scenes view. It will require you to become familiar with Swagger. Figure 22 shows how to access the Swagger interface. The authentication and authorization are shown in Figures 23 to 25 and 26, respectively. In Figure 27, the selection of an endpoint is shown with the example of the workflows. After execution, the documentation will also provide you with the request for the use with `cURL`. It can also be used as a template for an implementation of requests via Python. An examplary server response is displayed in Figure 28.

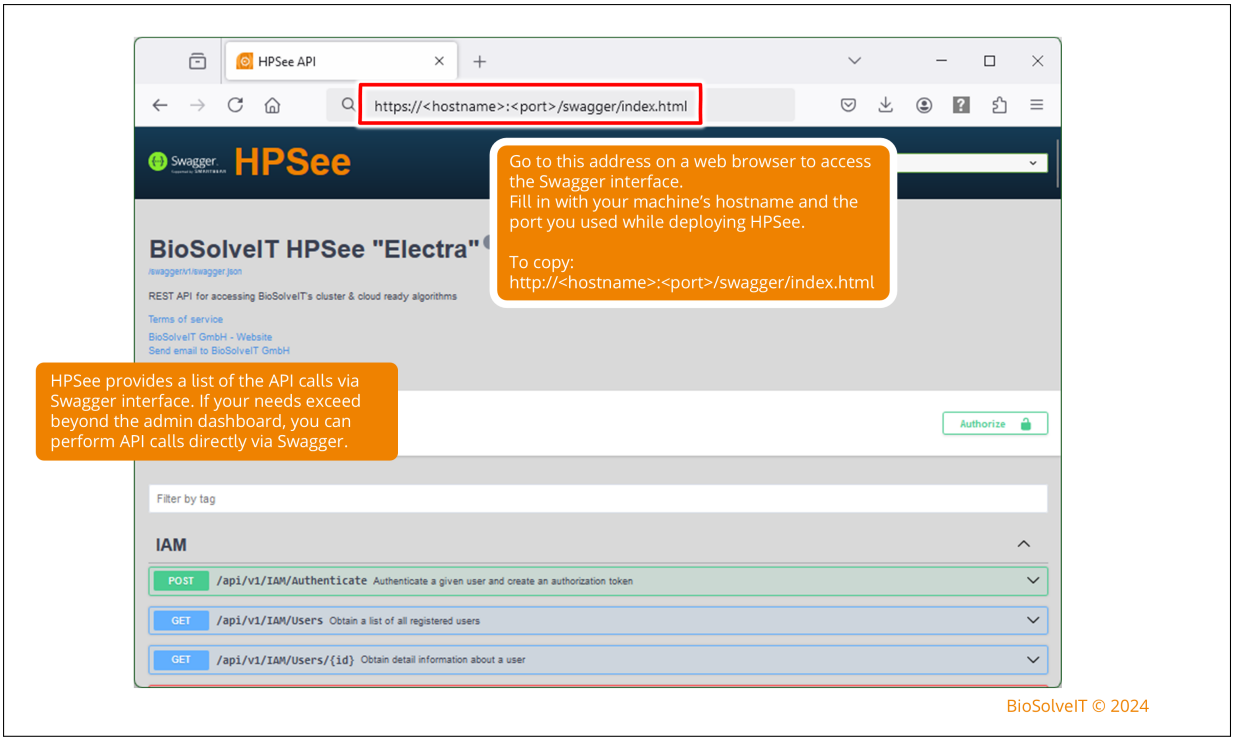

Figure 22: Access to the Swagger API documentation interface.

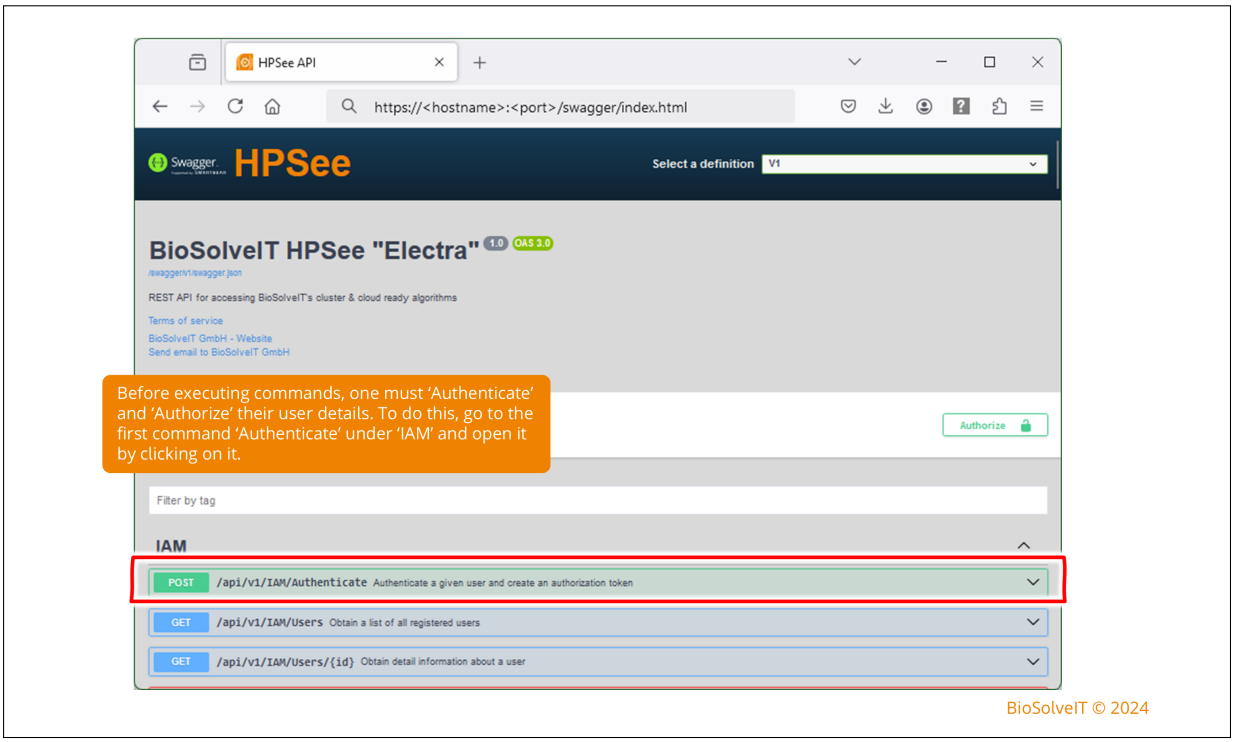

Figure 23: Authentication via the Swagger interface.

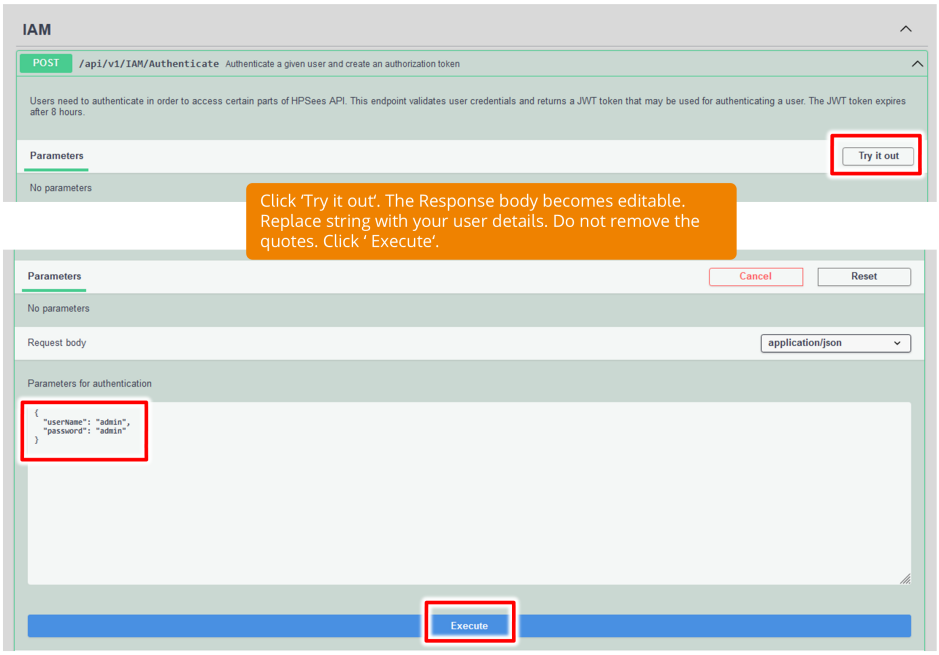

Figure 24: Login to receive a token.

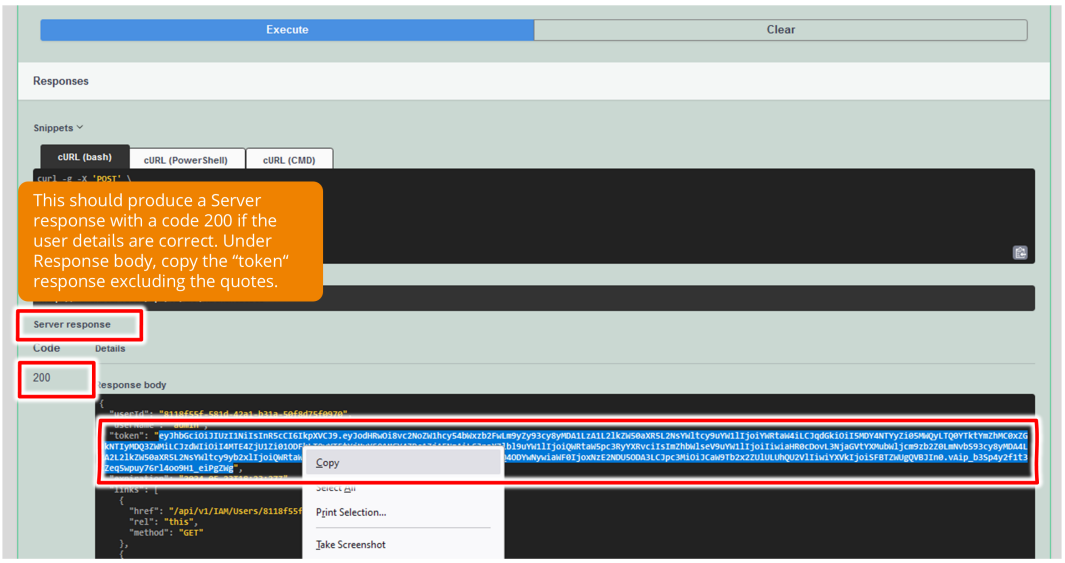

Figure 25: Token in API response.

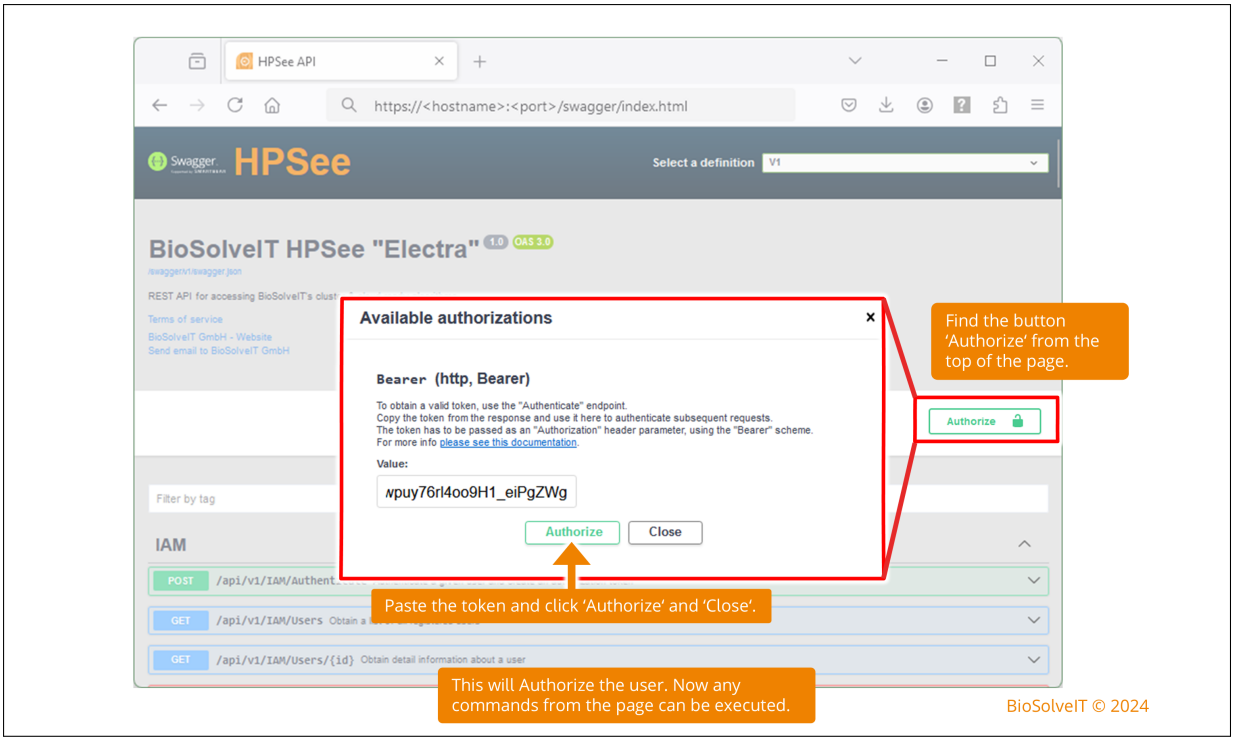

Figure 26: Authorization via the Swager interface.

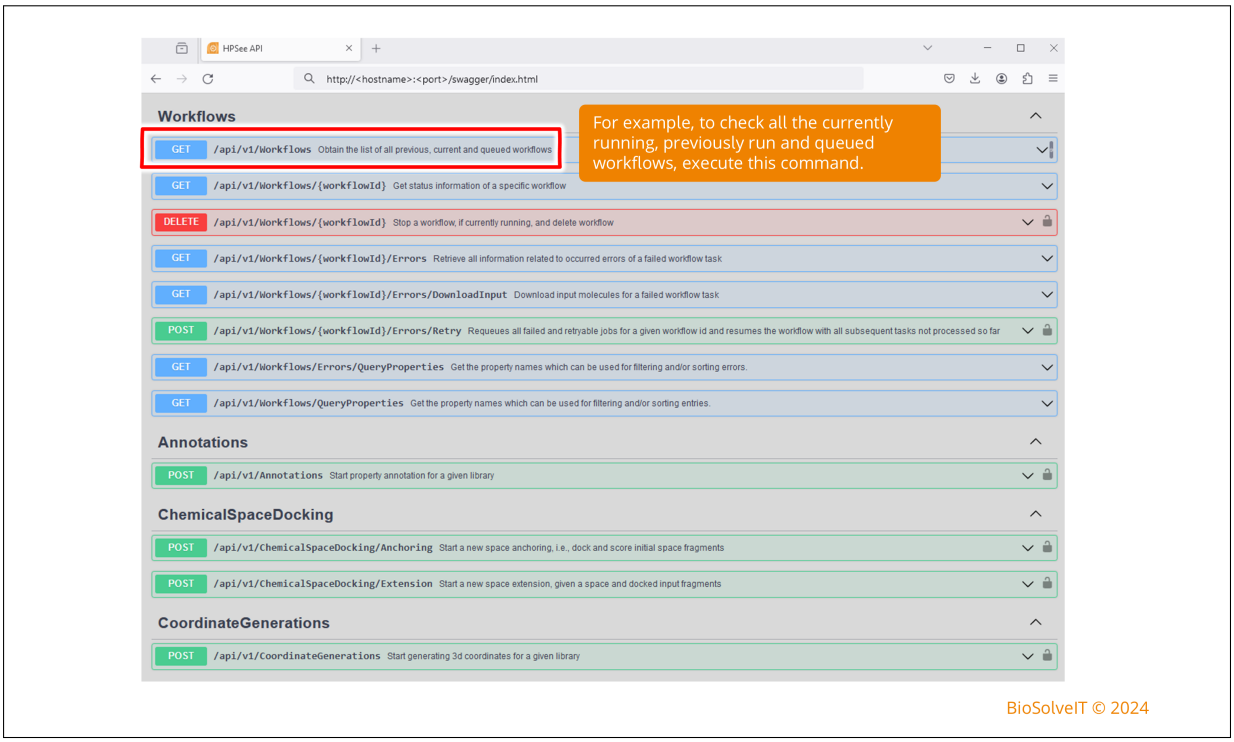

Figure 27: Selection of an endpoint via the Swagger interface.

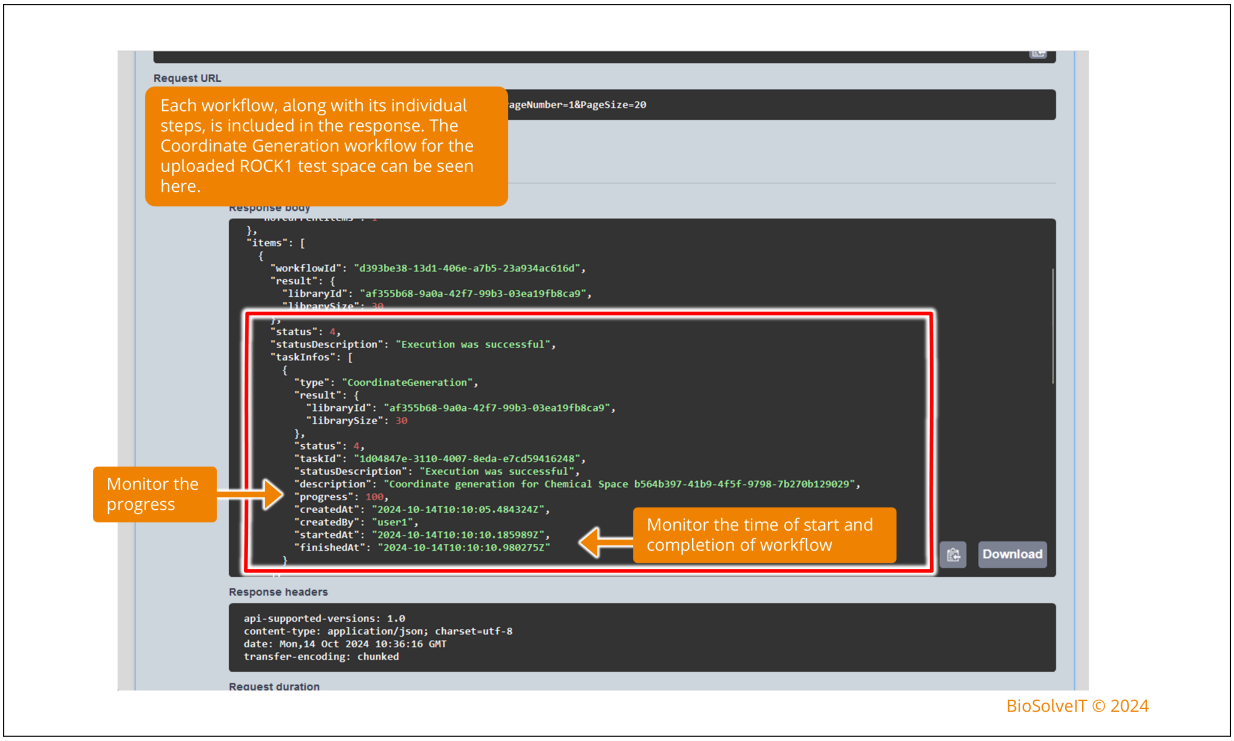

Figure 28: Response of the workflows endpoint via the Swagger interface.

6 Troubleshooting

6.1 Identify Hostname and IP

To identify the hostname of your system, on the commandline execute `hostname`. In case your system has troubles to resolve the hostname in another context, you may as well want to determine the IP. For this purpose, given the hostname, execute `ping <hostname>`. The IP will be part of the answer of the system.

6.2 Docker Images Cannot Be Pulled From the Registry

In some cases, images cannot be pulled from the registry. This may be due to a firewall or network configuration. Please contact your internal IT department. Images might as well be installed from archives that can be provided through BioSolveIT. For images as archives, please mail to support@biosolveit.de

6.3 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: support@biosolveit.com; and please mention any errors or warnings that you see.

6.4 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 [12].

7 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 BioSolveIT.

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://podman.io`.

[7] `https://docs.docker.com/engine/swarm/secrets`.

[8] `https://docs.docker.com/engine/swarm/secrets/#use-secrets-in-compose`.

[9] `https://hub.docker.com/_/postgres`.

[10] `https://docs.docker.com/storage/volumes`.

[11] `https://www.openssl.org/docs/`.

[12] `https://docs.docker.com/engine/swarm/swarm-mode/#configuring-default-address-pools`.

We wish you great success and much joy with HPSee!