

# Updating API Management

If you have a running **Scheer PAS API Management** installation and want to update, please do the following steps. If you experience problems during the update, refer to [Troubleshooting the API Management Installation](#).



## Migration Notes

Please also have a look at the API Management migration notes before starting with the update procedure.

- [API Management Migration Notes 7.1.0 -> 7.2.0](#)
- [API Management Migration Notes 7.2.0 -> 7.3.0](#)
- [API Management Migration Notes 7.4.0 -> 7.5.0](#)
- [API Management Migration Notes 7.5.0 -> 7.6.0](#)
- [API Management Migration Notes 7.6.0 -> 7.7.0](#)

## On this Page:

- [Step 1: Backup Your Current Installation](#)
- [Step 2: Download and Extract the Software](#)
- [Step 3: Configure the Installation Settings](#)
- [Step 4: Update the Software](#)
- [Step 5: Start all Services and Finalize the Update](#)

## Related Pages:

- [Troubleshooting the API Management Installation](#)

## Related Documentation:

- [API Management Administration > Export /Import Data](#)
- [API Management Release Notes](#)

## Step 1: Backup Your Current Installation

1. **Backup** your **API Management data** as described on [API Management Backup and Restore](#), section Data Export.
2. **Shut down** your API Management.  
To do this, run the following command:

```
docker-compose down
```



The **docker-compose.yml** can be found in the following folder:

- **api-mgmt\single-host-setup\**
- **api-mgmt\**

3. **Backup** the **MySQL** database and **Elasticsearch**. To do this, use the following commands:

```
docker-compose run --no-deps --rm -v $(pwd):/backup --entrypoint 'tar cvf /backup/mysql_data.tar /var/lib/mysql' mysql
```

```
docker-compose run -uroot --no-deps --rm -v $(pwd):/backup --entrypoint 'tar cvf /backup/es_data.tar /usr/share/elasticsearch /data' elasticsearch
```

Move the backup files to a safe location.

4. **Rename** folder **api-mgmt** your API Management installation resides in to e.g. **api-mgmt\_backup**. This backs up your current installation that will otherwise be overwritten by the update.

## Step 2: Download and Extract the Software

1. **Download** the new API Management software.
2. **Unzip** the downloaded software from **api-mgmt-<version>.zip**. This creates a new folder **api-mgmt** containing the new version of API Management.

## Step 3: Configure the Installation Settings

**Rework** the configuration files.

1. **.env File**  
Copy the **.env file** from your API Management backup **api-mgmt\_backup/single-host-setup** (see step 1/4) to your updated API Management installation.  
If there are new configuration variables coming with the new software release (see [API Management Release Notes](#)), you need to merge the previous **.env file** with the new one from the update.



As you have already an administrator account for your API Management installation, you can comment out the following lines in your **.env** file:

```
# KEYCLOAK_ADMIN_USERNAME=...  
# KEYCLOAK_ADMIN_PASSWORD=...
```

## 2. **docker-compose.yml**

If you did not stick to the standard installation and have made changes to the docker-compose.yml, you need to merge those changes into the new docker-compose.yml.

## 3. **Certificates and Keystores**

Copy all certificates and keystores from the backup folder of your old installation (**api-mgmt\_backup/configs**) to your updated installation (**api-mgmt/configs**). Overwrite the existing files from the target folder - they are example files only.

# Step 4: Update the Software

## 1. Load the new software with

```
docker image load -i api-mgmt-gateway-<VERSION>.tar
```

```
docker image load -i api-mgmt-ui-<VERSION>.tar
```

```
docker image load -i api-mgmt-keycloak-<VERSION>.tar
```

```
docker image load -i api-mgmt-devportal-<VERSION>.tar
```

## 2. Upgrade the MySQL database using the following commands:

```
docker-compose up -d mysql
```

```
docker exec -it mysql bash
```

```
mysql_upgrade --password
```



If you are asked for a password, use the password you specified in the **.env** file for **MYSQL\_ROOT\_PASSWORD**.

```
exit
```

```
docker-compose stop
```

# Step 5: Start all Services and Finalize the Update

## 1. Go to folder **api-mgmt/single-host-setup** (folder that contains the file **docker-compose.yml**).

**Start all containers** by running the following command:

```
docker-compose up
```

To run the containers in the background, use:

```
docker-compose up -d
```

2. **Upload** new configurations.

If the new API Management software comes with new configurations (e.g. new policies, see [API Management Release Notes](#)), you need to import them to your installation. Import the file(s) as described on [Restoring API Management Data](#).

3. **After** a successful update you may want to remove the old images to free up some disk space. In this case, just run

```
docker image rm api-mgmt/NAME-VERSION
```

For more clean up instructions refer to the [docker documentation](#).



Also consider removing the backup files from step 3 as they consume a considerable amount of disk space.