Adabas for Linux and Cloud

Adabas in a Docker Environment

Version 7.4.0

October 2025

ADABAS & NATURAL

This document applies to Adabas for Linux and Cloud Version 7.4.0 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ADAOS-DOCKER-740-20251030

Table of Contents

Preface

This document provides information about Adabas in a Docker environment. It is organized as follows:

Using the Official Adabas Container Images	Describes the location and basic usage of the prebuilt official Adabas Container Images.
Creating Container Images	Describes how to build a docker image.
Starting the Docker Container	Provides information on the steps and tasks that can be executed when the docker container starts up.
Adabas Concurrency Check	Adabas implements a locking mechanism designed to safeguard an Adabas database against multiple usage in different database processes. For further information, see *Adabas Basics*.

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://techcommunity.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Using the Official Adabas Container Images

Since the official release of Adabas 7.3, Software GmbH has been providing prebuilt Adabas container images for quick and easy use. The following images are available:

Image name	Description
`softwareagcr.azurecr.io/adabas:`*`version`*	Container image of Adabas without trace binaries and RESTful Administration Service
`softwareagcr.azurecr.io/adabas:`*`version`*`-trace`	Container image of Adabas with trace binaries and RESTful Administration Service

The images are updated with every Adabas update.

Note: The adabas-rest container images are deprecated and will no longer be developed further. However, the new Adabas 7.4 release now includes the RESTful Administration Service by default in the standard prebuilt Adabas container image.

To access the images, you have to:

1. Visit the official Container Registry at **https://containers.softwareag.com/products**.
2. Authenticate yourself using your Empower credentials.
3. Visit your **profile** and generate the required Container Registry credentials to pull container images from the official Container Registry.
4. To use these credentials on your local machine, enter `docker login -u username -p token softwareagcr.azurecr.io`.
5. Go to the product page of the desired container image and get the `docker pull` command.
6. Use the `docker pull` command on your local machine to download the desired container image.

For more information on how to use the images, visit the **Adabas** product page on the official Container Registry.

3 Creating Container Images

This section describes how to create your own container images if you do not want to use the official pre-built container images.

Adabas Docker Creation Scripts

The `installation directory`*/Adabas/docker/Adabas* directory contains scripts to create Docker-compatible packages from the Adabas installation. A prerequisite to use the scripts is to use the Software AG Installer for the installation of Adabas.

Use the scripts to generate a package for:

- a single Adabas instance without add-ons.
- a single Adabas instance with Adabas RESTful Administration. The installation must contain the Adabas RESTful Administration component.

The Adabas RESTful Administration allows you to administer and monitor an Adabas Docker container through a REST server port instead of using the `docker attach` command. Software GmbH provides a batch tool to connect to the container through a REST server port. For more information about administering Adabas through REST, see *Adabas REST Administration*.

Building a Docker Image

To create a Docker image, you need temporary and prepared files from an Adabas installation for your local instance. These temporary files can be removed after the desired container images are created.

You can use the `createAdabasSAGtar.sh` script inside the corresponding product directory (Adabas or Adabas-RestAdministration) to perform this specific task. Run the script for your desired use case (Adabas or Adabas with RESTful administration service) to create the temporary files in the same directory.

For example, to create the necessary files for a single Adabas instance without add-ons:

1. In a terminal window, go to `installation directory`*/Adabas/docker/Adabas*.
2. Execute the following command:

```
./createAdabasSAGtar.sh installation directory
```

where *installation directory* is the directory containing your Adabas installation.

The files now created under *installation directory/Adabas/docker/Adabas/tmp* contain the configuration of the Adabas installation and any custom scripts required for your container environment. You must generate new temporary files every time a configuration change occurs.

Before you build the Docker image, you can specify the values for user ID and group ID in the Dockerfile for your desired `sag` user inside the container. Docker creates the `sag` user and group during the image creation. This user requires a unique user ID and a group ID to create files in the persistent volume, located at */data*.

To build the container image with the default user ID and group ID of 1724, enter:

```
docker build --tag name:tag .
```

If you want to change the default values, add the `--build-arg` parameter to your build command:

```
docker build --tag name:tag --build-arg sag_user_id=XXXX --build-arg group_id=XXXX .
```

4 Starting the Docker Container

When you run the Docker container, an entry point shell script executes the following tasks:

1. Checks if the EULA is accepted, `ACCEPT_EULA=Y`.
2. Prepares the environment for Adabas.
3. Prepares the environment for the Adabas RESTful Administration service if you have Adabas REST Administration included.
4. Initializes the mounted Adabas license files if they are available.
5. Creates a new Adabas database. This step is optional.

   If you want to automatically generate a new Adabas database the first time you start the container, you need to set the following environment variables in your `docker run` command:

   a. Use the `ADABAS_DB_CREATION` environment variable to generate a database. Either use the `demodb` parameter to create the Adabas example database or define the `ADAFRM` Adabas parameters to create specific database container sizes.

      ```
      ADABAS_DB_CREATION=name=ABC assosize=(100M,100M)
      assoblock=(4k,16k) datasize=100M
      datablock=16k worksize=200M
      ```

   b. Specify the database ID using the `ADABAS_DBID` environment variable.

      ```
      ADABAS_DBID=177
      ```

      If you do not specify `ADABAS_DBID`, the ID of your Adabas database is equal to the deafault value of `001`.

   c. Specify the exposed access ports with the `docker run -p` parameter.

6. Restores the Adabas backup. This step is optional.

If you want to restore an Adabas backup, you must first edit the script. Use the ADABAS_RESTORE_BCK environment variable to restore files. For example:

```
ADABAS_RESTORE_BCK=/data/data.bck 4-10
```

7. Enables Adabas Analytics. This step is optional.

   If you want to enable Adabas Analytics, you must first edit the script. Use the ADABAS_ELA environment variable to enable the Adabas Analytics server.

8. Checks if another instance on a remote host is already started.

   Adabas allows to have multiple Adabas database nuclei work on the same database container. In that case, the adanuc process ensures that only one adanuc process is active at a time on this database container. The adanuc process creates a file called _DB_LOCK with which it synchronizes the different Docker container instances.

9. Starts the Adabas nucleus.

10. Starts the Adabas RESTful Administration service if Adabas REST Administration is included.

11. Checks the health of the database periodically.