ALTAIR

Altair Panopticon™ **v2025.0**
DOCKER COMPOSE PACKAGE
INSTALLATION AND REFERENCE GUIDE

TABLE OF CONTENTS

INTRODUCTION

Altair Panopticon is available as a Docker Compose package consisting of four Docker containers:

Docker Container	Image Names
Panopticon container	altairengineering.jfrog.io/panopticon-docker/com/altair/panopticon/panopticonviz
Python container	altairengineering.jfrog.io/panopticon-docker/com/altair/panopticon/pyserve
R container	altairengineering.jfrog.io/panopticon-docker/com/altair/panopticon/rserve
MonetDB (Data Store)	monetdb/monetdb

IMPORTANT	The Panopticon Docker Compose package is meant for convenient evaluation and testing of the product. It is not intended for production use. When mounting the Docker Compose package on a Windows host system, there will be performance penalties compared to a Linux host system due to folders mounted in the Windows file system. For any production grade usage of Panopticon, a Linux operating system is required. With this Docker Compose package, Panopticon is accessed on HTTP, not HTTPS. Web browsers will block some operations on HTTP connections to remote computers. This means that you will only be able to use Panopticon from **localhost** (i.e., from the same computer where Docker is running).

To use the Docker Compose package, you will need to complete the following steps:

1. Download and install Docker or some other equivalent container engine software, following the procedure valid for your operating system (Linux, maxOS, Windows)

2. Download the Panopticon Docker Compose package from Altair One Marketplace.

3. Configure Altair Units License settings in the file `Panopticon.properties.`

4. Create user credentials in the file `users.xml` folder.

5. Log on to Altair's docker image repository, and launch the Docker Compose package. The container images will be downloaded, and the containers will start.

    Each step is discussed in this document.

PANOPTICON DOCKER COMPOSE PACKAGE OVERVIEW

The Docker Compose package is downloaded from [Altair One Marketplace](#) in the form of a .ZIP file, which is around 25 MB in size.

Files and Folders in the Docker Compose Package

The downloaded .ZIP file is composed of the **pano-mount** folder and the following files:

Package Contents	Description
pano-mount	This folder contains three subfolders: config, data, and logs. This is the default docker bind mount path. The three subfolders contain files and folders used by the Docker containers for Panopticon, Python, R, and MonetDB. The files are for configuration settings, application data, and logs.
.env	The Docker Compose configuration file which specifies a Docker Compose environment variable named **PANO_MOUNT_DIR** which is the file path containing the folders for config, data, and logs. It also specifies the variable **PANOVIZ_IMAGE** which is the image version number of the Panopticon Docker image that will be used. These environment variables are used in the `docker-compose.yaml` file.
docker-compose.yaml	The Docker Compose configuration file which specifies the download URLs of the container images that will be used. The images are downloaded from [altairengineering.jfrog.io](#) when launching the Docker Compose package the first time. This file also contains configuration settings for the port numbers that will be used by each Docker container. The file also specifies several files and folders that will be used by the containers. File paths contain references to the Docker Compose environment variable **PANO_MOUNT_DIR** which is set in the `.env` configuration file.
altair-repo-login.bat	The Windows specific batch script file used for setting environment variables **DOCKER_REGISTRY**, **DOCKER_USERNAME**, and **DOCKER_PASSWORD** and effectuate a login to the Docker registry from where the docker container image files will be downloaded.
altair-repo-login.sh	The Linux/Unix/Mac specific shell script file used for setting environment variables **DOCKER_REGISTRY**, **DOCKER_USERNAME,** and **DOCKER_PASSWORD** and effectuate a login to the Docker registry from where the docker container image files will be downloaded.
Examples.exz	The Panopticon bundle file which contains example workbooks and dashboards. These examples can be imported into the Panopticon server after launching the Docker Compose package. This file makes up almost the entire file size of the Docker Compose package Zip file. **Examples.exz** is an optional file and can be removed from the Docker Compose package if you need to reduce the file size of the Docker Compose package.
README.md	Contains some brief instructions which are also covered by this document.

Step 1. Downloading and Installing Docker

Docker Desktop requires a paid license for business usage in companies above a certain size. As an alternative, you can use Rancher Desktop instead.

Docker Desktop

The easiest and most recommended way to get Docker and Docker Compose is to install Docker Desktop. Docker Desktop includes Docker Compose along with Docker Engine and Docker CLI which are Compose prerequisites. The exception is when you are using a Linux server from command line. In that case, Docker Desktop for Linux is not useful.

Instructions for Linux Computers

Docker supports several Linux distributions (e.g., Ubuntu, RHEL, etc.). Please refer to the documentation on docs.docker.com and follow the instructions relevant for your Linux distribution. Also ensure that you are using a Linux release version supported by Docker. For example, this page provides instructions for installing Docker Engine on Ubuntu Linux. Find the corresponding documentation for the Linux version you are using here: https://docs.docker.com/engine/install/ubuntu/.

While Docker Desktop is available for Linux in the form of a Virtual Machine, you should not involve Docker Desktop for Linux for the purpose of running the Panopticon Docker Compose package. Instead, you should install only the Docker Engine, which is managed by a command line interface (CLI).

Instructions for Apple Computers

On macOS, Docker is available as Docker Desktop. Please follow the instructions provided by Docker to install: https://docs.docker.com/desktop/install/mac-install/.

NOTE	In case you are running macOS on Apple Silicon hardware, such as M1 (not Intel), you will need to enable Rosetta emulation in Docker Desktop.
	You may need to upgrade macOS if you are running < Ventura 13.5.

Instructions for Windows Computers

On Windows, Docker is available as Docker Desktop. Please follow the instructions provided by Docker:

https://docs.docker.com/desktop/install/windows-install/.

NOTE	On Windows, one possibility is also to open a WSL prompt and follow the installation instructions for Linux to get Docker on your system in a way that avoids the Windows installation of Docker Desktop. You should not take this approach if your Windows system also has Docker Desktop installed, since it may give confusing results when starting and stopping containers.
	If you do choose to install Docker Engine on your WSL command line, note that you may have to change the name "**credsStore**" into "**credStore**" in the ~/.docker/config.json file.
	You may also find that you need to modify the WSL user's group membership / permissions:
	`sudo usermod -aG docker $USER` `sudo chmod 666 /var/run/docker.sock`
	Warning: By installing the Docker engine in your WSL Linux environment, you will not be able to use the Data Store feature in

Panopticon, because of a connection error between the Panopticon container and the MonetDB container. You will see an error logged in Panopticon saying:

```
SEVERE: [ImportDataService] Unable to connect
(Data Store): Connection refused.
```

The MonetDB log (merovingian.log) will show the following:

```
ERR monetdb[43]: #main thread:
createExceptionInternal: ERROR:
IOException:mal_mapi.listen:operation failed:
binding to UNIX socket file
/var/monetdb5/dbfarm/monetdb/.mapi.sock failed:
Operation not supported

MSG merovingian[10]: database 'monetdb' (43)
has exited with exit status 1
```

The issues are further discussed here:
https://github.com/MonetDB/MonetDB/issues/7069

Rancher Desktop by SUSE

Rancher Desktop by SUSE is an open-source application that provides all the essentials to work with containers and Kubernetes on the desktop.

Instructions for Linux Computers

Linux users can install via popular package managers. Read more about requirements and installation steps on https://docs.rancherdesktop.io/getting-started/installation/#linux

Instructions for Windows Computers

Steps:

1. Download the MSI file for installation on Windows, from https://rancherdesktop.io/.
2. Run the MSI file to install.
3. Start Rancher Desktop.
4. In *Preferences*, for the *Container Engine*, select **dockerd (moby)** to be able to use the same commands as with Docker.

Read more about requirements on https://docs.rancherdesktop.io/getting-started/installation/#windows.

Instructions for Apple Computers

Steps:

1. Download the suitable DMG installer for either Apple Silicon or Intel from https://rancherdesktop.io/.
2. Double-click the DMG file.
3. In the *Finder* window that opens, drag the Rancher Desktop icon to the *Applications* folder.
4. Navigate to the *Applications* folder and double-click the Rancher Desktop to launch it.
5. In *Preferences*, for the *Container Engine*, select **dockerd (moby)** to be able to use the same commands as with Docker.

Read more about requirements on https://docs.rancherdesktop.io/getting-started/installation/#macos**.**

Step 2. Downloading the Docker Compose Package

The Panopticon Docker Compose package is downloaded from Altair One Marketplace.

Steps:

1. Go to https://altairone.com and create a user account if you don't have one and log on.
2. Go to *Marketplace* then search **Panopticon**.

   Panopticon is displayed in the results pane.
3. Click the **Panopticon** tile. The application information and download options are displayed in the right pane.
4. Click the **Download** tab.
5. Select the latest version available and download the Docker Compose package.
6. Save the file anywhere on your system.
7. Extract the contents of the file to a location of your choice.

Step 3. Configuration of Altair Units License

Altair Panopticon requires a valid license. There are two options:

❑ Managed Altair Units License, where units are provided from Altair One license servers
❑ On-premises Altair License Manager and a valid license file added to the license manager

The license settings are entered in a configuration file which is part of the Panopticon Docker Compose package.

Managed Altair Units License

Steps:

1. Use a text editor like Notepad or Notepad++ and open the `pano-mount/data/panoviz/appdata/Panopticon.properties` file.
2. Find the properties with names starting with "license" and set the values as follows:

```
license.hwu.hosted=true
license.hwu.hosted.authorization.password=
license.hwu.hosted.authorization.token=<YOUR_TOKEN_HERE>
license.hwu.hosted.authorization.username=
license.hwu.uri=
license.hwu.use_client_timezone=true
license.mode=HWU
```

3. To generate the token, log on to Altair One Marketplace and go to https://admin.altairone.com/dashboard, click **User Profil**e and go to the tab **Authorized Machines** and click **Generate Auth Code**.

   For additional details, see Panopticon documentation.
4. Save and close the `Panopticon.properties` file.

NOTE	From the point where a new Altair One Managed License Token is created, the token must be used within 15 minutes, or otherwise it will become invalid and deleted. When the token gets used for the first time, it gets tied to the hostname and MAC-address of the machine using it. When the Panopticon Docker Compose package is mounted for the very first time, the four container images must be downloaded. In case you find that more than 15 minutes are required for downloading and mounting the containers and then log in at the Panopticon web interface, you will need to stop the containers, generate a new license token, and re-start the containers.

On-premises Altair License Manager

Steps:

1. Use a text editor like Notepad or Notepad++ and open the `pano-mount/data/panoviz/appdata/Panopticon.properties` file.

2. Find the properties with names starting with "license" and set the values as follows:

```
license.hwu.hosted=false
license.hwu.hosted.authorization.password=
license.hwu.hosted.authorization.token=
license.hwu.hosted.authorization.username=
license.hwu.uri=6200@host1:6200@host2
license.hwu.use_client_timezone=true
license.mode=HWU
```

3. For the **license.hwu.uri** property**,** if you are running more than one Altair License Manager, you can enter them with colon sign as a separator. Colon sign is the required separator character in this context when running on a Linux operating system, and the Docker container is a Linux based container.

   For additional details, see Panopticon [documentation](#).

4. Save and close the `Panopticon.properties` file.

Best Practices When Referencing the Physical Host

When using Panopticon in a Docker Container, and you want to use an Altair License Manager installed on the same physical host that runs Docker, you must specify **host.docker.internal** in the settings, instead of **localhost**.

Example:

```
license.hwu.uri=6200@host.docker.internal
```

Likewise, if you want to connect to a JDBC database running on the physical host, from Panopticon in a Docker container, you can do so by using a JDBC URL that specifies **host.docker.internal** instead of **localhost**.

Example, for PostgreSQL database:

```
jdbc:postgresql://host.docker.internal:5432/postgres
```

Step 4. - Configuration of User Login Credentials

User Login Configuration in users.xml

Steps:

1. For testing and development purposes, you can configure users, roles, and passwords in the `pano-mount/data/panoviz/appdata/users.xml` file.

2. Save and close the `users.xml` file. It is possible to edit the file while the Docker containers are running, and the changes will be effective without restarting the containers.

Associate User Roles with Panopticon Functional Roles

All authentication-related settings should be made in the `pano-mount/data/panoviz/appdata/security.yml` file. The server creates a template for this file when it starts if it's not already there.

Steps:

1. A Panopticon user is either a Viewer, a Designer, an Administrator, or a combination of Designer and Administrator. In the `pano-mount/data/panoviz/appdata/security.yml` file, usernames and user role names can be associated with these functional roles by setting values on these properties:

```
access:
  administrator.groups: pano-administrator-group, admin, pano-administrator
  designer.groups: pano-designer-group, designer, pano-designer
  viewer.groups: pano-viewer-group, user, pano-viewer
  default.roles: VIEWER
```

2. Save and close the `security.yml` file.

Step 5. Downloading Container Images and Starting the Docker Containers

Docker Compose will be starting up four containers, and for the three of them, the connection port is set to an automatically selected random high port number. However, the Panopticon container must use a known, static port since you will use it through a browser window. If your host system already has any process running which binds to port 8080 (the Panopticon container), then the container startup will fail.

Container Name	Container Port	Host Port
panoviz-1	8080	Static, set to 8080
monetdb-1	50000	Automatic, random
rserve-1	6311	Automatic, random
pyserve-1	9090	Automatic, random

If you depend on having some other application running on the same host, exposing port 8080, you have the option of changing the port mapping of the Panopticon container. You do this while the containers are stopped (command docker compose down, or press **Stop** in Docker Desktop).

Open the `docker-compose.yaml` file in the PanopticonDocker Compose package and change the value. For example, "**8080:8080**" to "**8081:8080**", or some other suitable port number. Likewise, you can choose to set a static port number to any of the other three containers, instead of relying on an automatically selected, random port, for example by changing from "**:9090**" into "**9090:9090**" or something else.

For Linux

1. On the command line, go to the folder where you have the files `.env` and `altair-repo-login.sh`.

   Add execute permissions to the folder pano-mount including all files and folders in it:

   ```
   chmod -R 777 pano-mount
   ```

2. Run this command:

```
./altair-repo-login.sh
```

| NOTE | You may also find that you need to use `sudo` to run `altair-repo-login.sh`:

`sudo./altair-repo-login.sh` |

This will establish an authenticated session with the Altair Docker image repository from where you will be downloading or pulling the images for the four containers included in the Docker Compose package.

3. Run this command:

```
docker compose up
```

| NOTE | You may also find that you need to use `sudo` to run docker compose up:

`sudo docker compose up` |

This will start the download of the docker container images, and the containers will be mounted. The download is needed only at the first start.

The containers will be up when you see a line like the following:

```
panoviz-1  | dd-MMM-YYYY HH:mm:ss.SSS INFO [main]
org.apache.catalina.startup.Catalina.start Server startup in [xyz]
milliseconds
```

4. Verify that you can reach Panopticon by opening a web browser and going to the URL:

```
http://localhost:8080/
```

For MacOS

Steps:

1. Start Docker Desktop by clicking the app icon.

| NOTE | You must open **Settings > Resources > File** sharing in Docker Desktop and add the *pano-mount* folder before starting up the containers, otherwise the bind will fail when mounting the containers. |

2. On the command line, go to the folder where you have the files `.env` and `altair-repo-login.sh.`

3. Run this command:

```
./altair-repo-login.sh
```

This will establish an authenticated session with the Altair Docker image repository from where you will be downloading or pulling the images for the four containers included in the Docker Compose package.

4. Run this command:

```
docker compose up
```

This will start the download of the docker container images, and the containers will be mounted. The download is needed only at this first start.

The containers will be up when you see a line like the following:

```
panoviz-1  | dd-MMM-YYYY HH:mm:ss.SSS INFO [main]
org.apache.catalina.startup.Catalina.start Server startup in [xyz]
milliseconds
```

5. Verify that you can reach Panopticon by opening a web browser and go to the URL:

   http://localhost:8080/

NOTE	Following this initial download and mount of the Docker Compose containers, you no longer need to use the command line to start and stop the containers. You can do that in the GUI of Docker Desktop for macOS.

For Windows

Steps:

1. Start Docker Desktop by clicking the app icon.

NOTE	On Windows, you don't have the option of starting the Docker daemon from command line, since that usage is supported only for Windows containers, not Linux containers.

2. Open a command line window and go to the folder where you have the files `.env` and `altair-repo-login.bat`.

3. Run this command:

   ```
   altair-repo-login.bat
   ```

   This will establish an authenticated session with the Altair Docker image repository from where you will be downloading or pulling the images for the four containers included in the Docker Compose package.

4. Run this command:

   ```
   docker compose up
   ```

   This will start the download of the docker container images, and the containers will be mounted. The download is needed only at this first start.

   The containers will be up when you see a line like the following:

   ```
   panoviz-1  | dd-MMM-YYYY HH:mm:ss.SSS INFO [main]
   org.apache.catalina.startup.Catalina.start Server startup in [xyz]
   milliseconds
   ```

   You will also see the containers on the Containers tab in Docker Desktop, where you can also monitor Container CPU usage and memory usage.

5. Verify that you can reach Panopticon by opening a web browser and go to the URL:

   http://localhost:8080/

NOTE	Following this initial download and mount of the Docker Compose containers, you no longer need to use the command line to start and stop the containers. You can do that in the GUI of Docker Desktop for Windows.

STOPPING THE DOCKER COMPOSE CONTAINERS

For Linux

Stop the Docker Compose containers by running the following command:

```
docker compose down
```

NOTE	After running the command `docker compose up,` you may find that the command line window stays in docker compose, without returning to an operating system prompt. You can either: • Press **Ctrl+C** to stop the containers, or • Pause the docker process by pressing **Ctrl+Z**, then executing `bg` to move it to the background, and then executing `sudo docker compose down.`

For MacOS

To stop the Docker Compose contains, you can either:

❑ Stop the Docker Compose container in Docker Desktop

❑ Run the following command:

```
docker compose down
```

For Windows

To stop the Docker Compose contains, you can either:

❑ In the command line window where you previously entered the command `docker compose up,` press **Ctrl+C** and wait for the four containers to be stopped, or

❑ Press the stop-button of the Panopticon container in Docker Desktop

ADDING THE EXAMPLES COLLECTION TO PANOPTICON

Examples are added to the Panopticon server by importing the bundle file named `Examples.exz` that is included with the Panopticon Docker Compose package.

A recommended approach is that you either:

❑ Select the **Keep folder structure** checkbox when doing the import to get the examples placed inside a new folder named **Examples**, or

❑ Import the bundle into a new folder under the Organization section

On the *Organization* section, right-click and select **New Folder** and enter a folder name. Then right-click on the folder and select **Import Bundle** in the context menu.

NOTE	You will need to be logged on to Panopticon as a user with Administrator capabilities to be able to create a new folder in the *Organization* section.
	Refresh the browser if:
	• The *Import Bundle* dialog displays with a spinning wheel
	• The example files are missing their thumbnail images

OTHER CONFIGURATION SETTINGS IN THE DOCKER COMPOSE PACKAGE

To change any configuration, you must first stop the Docker Compose containers. Make the settings changes, and then start the containers again.

Accessing the Managed License Server Through a Proxy Server

When using Managed Altair Units License, where the license manager server is hosted by Altair in the cloud, you have the option of configuring Panopticon to access the managed license server through a proxy. Part of the steps for making this configuration are the same regardless of whether you're running Panopticon directly on the host operating system or running Panopticon in a Docker container. That part of the instructions can be found in the document *Panopticon Real Time Installation and Reference Guide*. The remaining steps, that are specific to running Panopticon in Docker, are discussed below:

1. Stop the Panopticon Docker Compose package if it is running.

2. Edit the `panoviz` container settings in the `docker-compose.yaml` file as follows:

   2.1. Modify the `environment` section into:

```
environment:
    - JAVA_TOOL_OPTIONS="--add-opens=java.base/java.nio=ALL-UNNAMED"
    - ALM_HHWU_PROXY_HOST="<proxy host>"
    - ALM_HHWU_PROXY_PORT="<proxy port>"
```

   2.2. Modify the `volumes` section into:

```
volumes:
    - type: bind
      source: ${PANO_MOUNT_DIR}/data/panoviz/appdata
      target: /etc/panopticon/appdata
    - type: bind
      source: ${PANO_MOUNT_DIR}/config/panoviz/server.xml
      target: /usr/local/tomcat/conf/server.xml
    - type: bind
      source: ${PANO_MOUNT_DIR}/logs/panoviz
      target: /usr/local/tomcat/logs
    - type: bind
      source: <ALMutil config files folder path>
```

```
                    target: /usr/local/altair
```

3. Modify the `license` properties in the `/pano-mount/data/panoviz/appdata/Panopticon.properties` file:

```
license.hwu.hosted.authorization.password=
license.hwu.hosted.authorization.token=<YOUR_TOKEN_HERE>
license.hwu.hosted.authorization.username=
license.hwu.hosted=true
license.hwu.uri=<proxy port>@<proxy host>
license.hwu.use_client_timezone=true
license.mode=HWU
```

4. Restart the Docker Compose package.

Configuring the Sender Account for Email Reports and Email Alerts

By setting email account details for a sender account, it will be possible to send reports, data exports, and alert messages via email. In the `pano-mount/data/panoviz/appdata/Panopticon.properties` file, configure the following properties:

```
email.address=
email.host=
email.password=
email.port=
email.security.mode=NONE
email.username=
```

You will likely not be allowed to use your workplace email account as a sender account configured in Panopticon. For testing purposes, you can optionally use a third-party email service. There are several email services to choose from, which have entry level plans free of charge. Below is a brief instruction for how you can use Sendgrid:

1. Create a free account on Sendgrid.com.

2. Do a Single Sender Authentication by specifying an email address that you own.

    You will get an email with a link to click to prove that the email address is yours. The purpose of this is for Sendgrid to know that you will not expose someone else's email address as the sender.

3. Under **Email API > Integration Guide**, choose **SMTP Relay**.

4. Enter a descriptive name for a new secret API key and press **Create Key**.

5. Copy the secret API key. This will be used as the email account settings password in Panopticon.

6. Stop the Docker Compose containers. In the `Panopticon.properties` file, enter the values given by Sendgrid:

    -   For the port, enter **465.**

    -   For the security mode, enter **SSL.**

7. Start the Docker Compose containers.

    You will now be able to email PDFs, data exports, dashboard images, graph images, and alert notifications from Panopticon. The email server that is used for sending is provided by Sendgrid, and the Single Sender email address that you provided will be presented as the sending address.

NOTE	Since you are not *actually* sending email from the Single Sender email address that you authenticated, but only *presenting* it as the sending email address, this can be detected by the recipients' email server. The receiving email server could therefore consider the message to be suspicious and treat it like spam or malicious email. The way to solve this is to add certain CNAME records to the DNS service of your domain, as specified by Sendgrid, and possibly also an SPF record.

Enabling REST API Swagger and Client JavaScript API Documentation

Panopticon has a collection of REST endpoints. The documentation of the REST endpoints is disabled by default, but can be enabled by setting the property `documentation.enabled=true` in the `pano-mount/data/panoviz/appdata/Panopticon.properties` file.

This will give you access to the following:

❑ REST endpoint documentation - http://localhost:8080/swagger-ui.html

❑ Client JavaScript API documentation - http://localhost:8080/panopticon-api-doc/index.html

Adding Custom Map Tile Providers

Panopticon has a visualization part called Map Plot, which uses map tiles. You can add map tile services by editing the `pano-mount/data/panoviz/appdata/default-settings/tileproviders.json` file.

For additional details, see Panopticon documentation.

Adding Packages to R

The Rserve container of the Panopticon Docker Compose package comes with several installed packages. Additional packages are also installed based on what has been added to the `pano-mount/config/rserve/init_R_env_override.R` file.

You can also source your own functions by placing .R files inside this folder:

`pano-mount/config/rserve/rserve_extensions`

Then add a line like this in the `init_R_env_override.R` file.

`source("/etc/panopticon/rserve_extensions/yourfilename.R")`

Alternatively, you can place the source statement at the beginning of your R code in each Panopticon data table, and thereby source different files with different tables. In either case, after sourcing your own function definitions, you can call your functions directly into your Panopticon data tables without any further importing or attaching.

Adding Packages to Python

The Python container of the Panopticon Docker Compose package comes with several installed modules. Additional modules are also installed based on what has been added to the `pano-mount/config/python/requirements.txt` file.

You can import any module into your Python scripts used in Panopticon by specifying the file name without the .py extension, for example, with a file "my_python_functions.py", you do:

`import my_python_functions`

Adding JDBC Drivers

Steps:

1. Stop the containers then place the JDBC driver jars in the `pano-mount/data/panoviz/appdata/jar_extensions` folder.

2. Restart the containers.

   The Panopticon container log will include lines like these:

   ```
   /etc/panopticon/appdata/jar_extensions exists.
   Final JAR_EXT_FOLDER=/etc/panopticon/appdata/jar_extensions
   Copying all jars from extensions folder done
   ```

Configuring SSL

The Docker Compose package exposes the Tomcat `server.xml` file in the path `/pano-mount/config/panoviz/server.xml`.

Steps:

1. Configuration of SSL is done in this file, by adding a block like the one below:

   ```
   <Connector
     protocol="org.apache.coyote.http11.Http11NioProtocol"
     port="8443"
     maxThreads="150"
     SSLEnabled="true">
     <SSLHostConfig>
       <Certificate
       certificateKeystoreFile="conf/keystore.jks"
       certificateKeystorePassword="kspass"
       type="RSA"/>
     </SSLHostConfig>
   </Connector>
   ```

2. In addition, the following configuration changes must be made in the `docker-compose.yml` file:

Configuration Property	Sample Values
ports	• **8080:8080** • **8443:8443**
type:bind	• **source: ${PANO_MOUNT_DIR}/config/panoviz/keystore.jks** • **target: /usr/local/tomcat/conf/keystore.jks**

3. Generate a `keystore.jks` file (instructions and tools exist on the web).

4. Place the `keystore.jks` file in the config folder path shown above.

5. Run the `docker compose up` command to build your container.

CONTAINER MONITORING AND LOGGING

In Docker Desktop, you can monitor what is happening inside a container by clicking on the container name and looking at the **Logs** tab. You also have the tabs **Inspect**, **Bind Mounts**, **Exec**, **Files**, and **Stats**.

Logs are persisted to file in the following paths:

❑ pano-mount/logs/panoviz

  • Catalina log

- Host-manager log
- Localhost log
- Localhost access log
- Manager log
- Panopticon log
- Panopticon audit log
- pano-mount/data/monet
  - merovingian.log

UPGRADING THE DOCKER COMPOSE PACKAGE

When you have already used the Panopticon Docker Compose package, you can upgrade it to a later version. Select one from the following options.

Option 1

Add the new Docker Compose package as a separate installation by placing it in a new, separate folder, while keeping your current Docker Compose package as it is. When using this approach, follow the instructions in this section.

Option 2

Upgrade your current Docker Compose installation and continue using the same configurations and Panopticon content such as data tables and workbooks. When using this approach, follow the procedure described below.

Files to Keep

The following files and folders that can contain configurations, customizations, and server content that you need to keep:

- pano-mount/config/*
- pano-mount/data/*
- pano-mount/logs/*

Files That May Have Changed

Files that may have changed in the newer version of the Docker Compose package zip file, which means that you must replace the old files with new ones, are the following:

- .env
- altair-repo-login.bat
- altair-repo-login.sh
- docker-compose.yaml
- Examples.exz

NOTE	In case you have modified the **PANO_MOUNT_DIR** value in the .env file to a path other than `/pano-mount`, then you must edit the new version of the .env file the same way.
	In case you have modified the `docker-compose.yaml` file (e.g., port mappings), you must make the same changes in the new version of the `docker-compose.yaml` file.
	There is a possibility that the files in `/pano-mount/config/python/` and `pano-mount/config/rserve` have been modified in the new Docker Compose package. If you have made no customizations to those files, you could replace them with the files from the newer package. In case you have made modifications, you must manually review and compare the old and the new files and manually consolidate your customizations and the new file versions.

Detailed Upgrade Instructions

1. Stop and delete the containers, and then delete the images.

   This is done in Docker Desktop if you are on macOS or Windows. On Linux without a desktop gui, use the command line as described by Docker documentation.

NOTE	On Windows and Mac, use the command `docker images` to check that the images are deleted. If they still are listed, then use the command `docker rmi` to delete the images.

2. Download the latest Panopticon Docker Compose zip file from Altair One Marketplace.

3. Extract the contents of the Panopticon Docker Compose zip file to a path different from the location of your current Docker Compose package to avoid overwriting your current setting.

4. You can opt to compare old and new files in the `/pano-mount/config/python/` and `/pano-mount/config/rserve` folders for reasons described in the note above.

5. From the folder where you unzipped the new Docker Compose package, copy the files listed in this [section](#) and paste them in the folder where the old versions are located.

6. In case you did not keep the settings from your `Panopticon.properties` file and you are using Managed Altair Units License; you may discover that you need to create a **new machine authorization token** ("Auth Code") on Altairone.com as described in this [section](#).

7. To download or pull new container images and mount the containers, follow the procedures in this [section](#).

ABOUT PANOPTICON

For more information on Panopticon and other resources, go to https://www.altair.com/panopticon.