PROXEDO

API SECURITY

Proxedo API Security in Kubernetes

Migration manual from 4.7.0 to 4.8.0

2023-12-07

1. Introduction

This guide describes the necessary steps and commands to upgrade PAS instance from version 4.7.0 to 4.8.0.

There are three main stages to describe the different phases and scenarios of the upgrade.

1. Creating a backup for safety
2. Upgrading PAS in Kubernetes
3. Restoring the pre-upgrade state

2. Creating a backup for safety

Before starting the upgrade process, make sure there is a backup to which the current state of the actual PAS setup can be restored.

2.1. Bootstrap configuration

As the whole bootstrap configuration is provided at the time of installation, the directory, in which the installation was carried out, needs to be saved, so that the installation procedure can be repeated.

2.2. Creating a backup of the running configuration

1. Log in to the Web UI as admin user and navigate to the *Configuration Backup* page from the top bar.
2. On the *Configuration Backup* page select *Running* from the *Export configuration* dropdown menu.

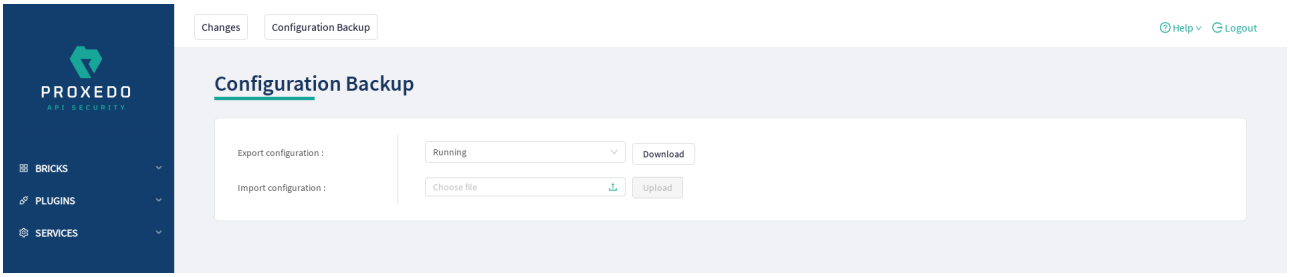

Figure 1. Backup and restore running or user configuration for Proxedo API Security

3. To export the running configuration, press the *Download* button. This will save the running configuration to a file named `running-config-backup.zip` in the working directory.

For more details, see section *Backup and restore running or user configuration for Proxedo API Security* in *Proxedo API Security in Kubernetes: Administration Guide*.

3. Upgrading PAS in Kubernetes

This section describes how to upgrade PAS in a single node setup. In case any problem occurs during the upgrade and the version 4.7.0 needs to be restored, follow the instructions in section Restoring the pre-upgrade state.

The upgrade process needs to be carried out on a Linux machine.

3.1. Prerequisites

The following requirements need to be met before carrying out the upgrade process:

- The management and storage components of version 4.7.0 are running and healthy.
- The password for the administrator user of the management component is available.
- The new *Helm chart* is downloaded: `/tmp/proxedo-api-security-4.8.0.tgz`.
- The same resources are available that have been available at the time of the installation of the old version of PAS. For more details, see section *Prerequisites for installing PAS* in *Proxedo API Security in Kubernetes: Administration Guide*.
- The user who carries out the upgrade, needs to be logged in to docker.balasys.hu via the docker login command line tool, and needs to have access to the PAS docker images.

3.2. Upgrade steps

During this process, the new version of PAS will replace the old version. The commands below will use `proxedo-api-security` for namespace.

If Windows Subsystem for Linux (WSL) is used, substitute `host.docker.internal` for `localhost` in the `--frontend-baseurl` flag.

1. Set up *port forward* for the old frontend by running the following command. This will make the Web UI available on the localhost.

```
kubectl --namespace proxedo-api-security port-forward services/proxedo-api-security-
frontend 8080:80
```

2. Run the following command in a new shell. After the pre-upgrade process is run, a `pas-config-4.7.0.zip`. can be found in the working directory.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.8.0 --frontend-baseurl
http://localhost:8080 pre-upgrade
```

3. Run the following command in the same shell and working directory. It attempts to convert the running configuration to make it compatible with the new PAS version. If the conversion is successful, the new configuration is saved as `pas-config-4.7.0.upgraded-to-4.8.0.zip`. However, if the automated conversion fails, the converter prompts the user for input. In case of validation failure, you will be required to specify the correct type of the current *Generic* file brick. The converter will provide appropriate validation errors specific to the selected file brick type. Afterwards, the user needs to manually correct the configuration through the web user interface.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.8.0 convert-config
```

The new file types must meet the following validation criteria to convert from the generic type.

Table 1. Requirements for the new file types

File type	Requirements
Diffie-Hellman Parameters	• Must be in PEM format. • Must be a parameters file, such as one generated by the `openssl dhparam` utility.

TLS Key	• Must be in PEM format. • Must be a private key file. • Could be encrypted or unencrypted. If the file is encrypted, the passphrase must be provided in the Passphrase field.
Client Certificate	• Must be in PEM format. • Must be a certificate file. • Must have a Common Name attribute, and have the CLIENT_AUTH ExtendedKeyUsage.
Server Certificate	• Must be in PEM format. • Must be a certificate file. • Must have a Common Name attribute, and have the SERVER_AUTH ExtendedKeyUsage.

If a validation error occurred during the previous step and manual corrections were made through the web user interface, re-run the convert-config command to check the configuration again. This check will also verify if the correct file brick types are used in the appropriate references.

4. Stop the *port forward* process in the first shell by pressing CTRL + C.
5. Install the new version of PAS by using the same input files as used for the old one. Follow the instructions in section *Providing the necessary files for Helm installation* in *Proxedo API Security in Kubernetes: Administration Guide*.
6. Set up *port forward* now for the new frontend by running the following command. This will make the Web UI available on the localhost.

```
kubectl --namespace proxedo-api-security port-forward services/proxedo-api-security-
frontend 8081:80
```

7. Run the following command in a new shell, in the same directory where the pre-upgrade and configuration conversion steps were performed. Follow the instructions of the script to complete. A timeout value for the application of the new configuration needs to be specified in the **timeout** field. Select a value (number of seconds) that will probably be enough for the whole PAS deployment to be rolled out. It can depend on the number of core pod instances and the bandwidth the cluster has to docker.balasys.hu.

```
docker run --user $(id -u):$(id -g) --rm -it --network=host --volume $(pwd):/upgrade
docker.balasys.hu/api-security/config-upgrader:4.8.0 --frontend-baseurl
http://localhost:8081 post-upgrade --config-apply-timeout <<timeout>>
```

8. Close the *port forwarding* in the other shell by pressing CTRL + C.

4. Restoring the pre-upgrade state

In order to restore the state prior to the upgrade, complete the following stages:

1. Complete a factory reset. In case a factory reset is necessary, the simplest solution is to delete the namespace, PAS is installed in. If that is not feasible, an alternative is to explicitly delete Kubernetes objects related to PAS. To do so, two main steps are required:

a. Uninstall the PAS *Helm* chart using the `helm uninstall proxedo-api-security` command. This will

remove all kubernetes objects managed by the *Helm* charts, including the *Persistent Volume Claim* associated with the storage components.

b. Delete the core configuration objects. These objects are not managed by the *Helm* chart but by the management component. To complete this, run the following commands:

   - `kubectl delete configmap proxedo-api-security-core-config`
   - `kubectl delete secrets proxedo-api-security-core-config proxedo-api-security-registry-credentials`

Following these steps, PAS shall be installed from scratch. For more details, see section *Installation of Proxedo API Security in Kubernetes environment* in *Proxedo API Security in Kubernetes: Administration Guide*.

2. Repeat the installation with the files backed up from the 4.7.0.

3. Import the configuration with the help of the Web UI, see steps in section *Restoring the running configuration*.

4.1. Restoring the running configuration

1. Log in to the Web UI as the administrator user and navigate to the *Configuration Backup* page from the top bar.
2. To import the running configuration, on the *Configuration Backup* page choose a configuration file from the computer and press *Upload* to upload the configuration.

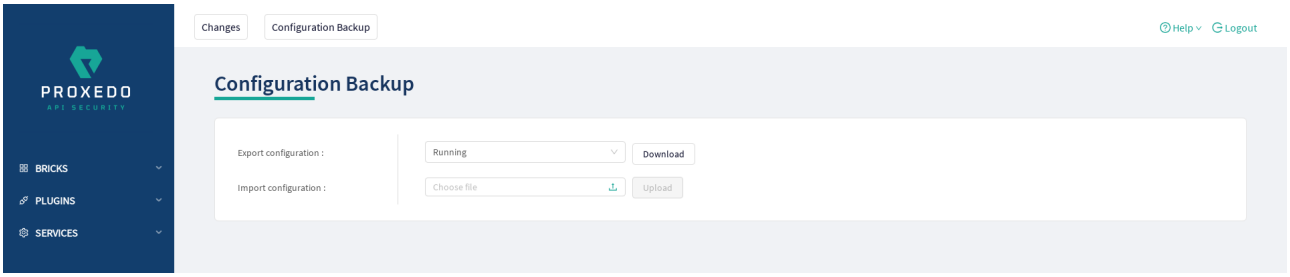

Figure 2. Backup and restore running or user configuration for Proxedo API Security

3. Apply the configuration.