Engineering
Simplicity

Upgrading Control Center to 4.2

Published
2023-09-27

RELEASE
4.2

Table of Contents

General

IN THIS SECTION

Upgrade Paths

This document describes how to upgrade from Paragon Active Assurance version 3.3 ... 4.1 to version 4.2. This involves an upgrade of Ubuntu from version 18.04 to 22.04.

To contact Juniper technical support, file a ticket at support.juniper.net/support/requesting-support.

Please also contact technical support whenever you want to upgrade from a version or between versions that are EoS.

Release Notes

Before starting the upgrade, please always read the Release Notes for the version you are upgrading to. These notes describe new features and also inform you of important under-the-hood changes such as new configuration files.

If you are upgrading across multiple versions, please read the Release Notes for all intermediate versions.

Finding Out Your Paragon Active Assurance Software Version

To find out what version of Paragon Active Assurance you currently have installed, you can use this command:

```
dpkg -l | grep paa
```

Upgrade Procedure

IN THIS SECTION

Below are instructions for upgrading Control Center. This is the procedure that should normally be followed. An alternative procedure is described in the next chapter.

Be sure to refer to the current *Installation Guide*.

- **Disable** the `apache2` and `netrounds-callexecuter` services completely:

```
sudo systemctl disable apache2
sudo systemctl disable netrounds-callexecuter
```

- **Stop** all Paragon Active Assurance services:

```
sudo systemctl stop "netrounds-*" apache2 openvpn@netrounds
```

- **Make backups** according to the Operations Guide, chapter Backing Up Product Data, starting with the section "Backing Up the PostgreSQL Database".

  In addition, back up the file `/etc/netrounds/secret_key` and keep it with the backup archives:

  ```
  cp /etc/netrounds/secret_key /etc/netrounds/secret_key-41
  ```

- **Make backups** of the disks or disk partitions used by Control Center (or take a VM snapshot).

- **Verify that all services are stopped:**

  ```
  sudo systemctl status "netrounds-*" apache2 openvpn@netrounds
  ```

  If some service is not in status stopped, run the following command once more:

  ```
  sudo systemctl stop "netrounds-*" apache2 openvpn@netrounds
  ```

  If the problem persists, please contact Juniper technical support.

- **Upgrade Ubuntu** from version 18.04 to version 22.04. This can be done in two steps as described on the pages linked below:

  - Upgrade from 18.04 to 20.04 (Click the "Clear message" link to bring up the page with instructions.)

  - Upgrade from 20.04 to 22.04 (Here, version 21.10 is taken as starting point, but it is equally possible to start from 20.04.)

- **Verify the integrity** of the tarball containing the new Control Center version:

  ```
  # Compute the checksum for the tar file and verify that it is equal to the SHA256
  # checksum provided on the download page
  export CC_VERSION=4.2.0.54
  sha256sum paa-control-center_${CC_VERSION}.tar.gz
  ```

- **Unpack the Control Center tarball:**

  ```
  tar -xzf paa-control-center_${CC_VERSION}.tar.gz
  ```

- **Install** new **Control Center** packages.

Note the following:

- In the file /etc/netrounds/netrounds.conf you need to replace the setting PASSWORD_RESET_TIMEOUT_DAYS with PASSWORD_RESET_TIMEOUT. The former gives the timeout in days, whereas the latter has seconds as unit.
- In the file /etc/netrounds/netrounds.conf you can also optionally configure the SPEEDTEST_ADDRESS setting (if you are going to use Speedtest). This can either point to the same IP address that SITE_URL resolves to, or it can have a hostname of its own.

WARNING: You will be prompted about overwriting existing configuration files. Before proceeding, please read all the information about settings below.

NOTE:

- We highly recommend that you first inspect the difference between your old configuration and the new one using the "D" choice. In most cases you will then want to keep your old settings by pressing "N" (do not overwrite).
- New optional and updated settings may be available in the example configuration files provided in the packages. We recommend that you review these and add new options as appropriate for your installation.

WARNING: For the Apache configuration files found in

```
/etc/apache2/sites-available/
```

you need to press "Y", which is the "package maintainer's version".

If you have installed proper SSL certificates (as recommended) instead of the default snakeoil ones, you will have to modify the file again to point to the correct path in the SSLCertificateFile and SSLCertificateKeyFile settings after the Debian package installation has completed. See the Installation Guide, chapter Service Configuration, section "SSL Certificate Configuration".

```
sudo apt-get update
sudo apt-get install ./paa-control-center_${CC_VERSION}/*.deb
```

- Run the database migration:

WARNING: If you have changed the database password from the default, make sure you also change this in the `db-password` setting in the `/etc/netrounds/plugin.yaml` file before running `ncc migrate`. Otherwise, the command will fail.

NOTE:

- This is a sensitive command, and care should be taken when executing it on a remote machine. In such a scenario it is strongly recommended that you use a program like `screen` (generally installed by default on popular Linux distributions) or `tmux` (run `sudo apt-get install tmux` to install) so that the migrate command will continue running even if the ssh session breaks.
- This command takes considerable time to execute.

```
sudo ncc migrate
```

- Change the `/var/lib/netrounds/openvpn` owner to `netrounds`:

```
sudo chown -R netrounds: /var/lib/netrounds/openvpn
```

- **Restart** all Paragon Active Assurance services:

```
sudo ncc services restart
```

- **Install the new Test Agent repository and plugins.**

  The plugins are used by Test Agent Applications.

```
TA_APPLIANCE_BUILD=4.2.0.34
TA_APPLICATION_BUILD=4.2.0.20
PLUGIN_BUILD=4.2.0.29

# Compute checksums for the repositories and verify that they match the
# SHA256 checksums provided on the download page
sha256sum paa-test-agent_${TA_APPLIANCE_BUILD}_all.deb
sha256sum paa-test-agent-application_${TA_APPLICATION_BUILD}_all.deb
```

```
sha256sum paa-test-agent-plugins_${PLUGIN_BUILD}_all.deb

# Start the installation
sudo apt-get install ./paa-test-agent_${TA_APPLIANCE_BUILD}_all.deb
sudo apt-get install ./paa-test-agent-application_${TA_APPLICATION_BUILD}_all.deb
sudo apt-get install ./paa-test-agent-plugins_${PLUGIN_BUILD}_all.deb
```

- **Enable** services as follows:

```
sudo ncc services enable apache2
sudo ncc services enable kafka
sudo ncc services enable callexecuter
```

NOTE: If the TimescaleDB and Metrics services are expected to be running after the upgrade, you need to enable these as well since they are not enabled by default:

```
sudo ncc services enable timescaledb
sudo ncc services enable metrics
```

- **Restart** all Paragon Active Assurance services:

NOTE: You must do this to get the services up and running again after the upgrade.

```
sudo ncc services restart
```

- To activate the new configuration, you also need to run:

```
sudo systemctl reload apache2
```

- Check that the system is up and running with the commands

```
ncc status
sudo systemctl status "netrounds-*"
```

- Do the following to enable the latest version of all plugins in all accounts:

  ```
  ncc plugins edit enabled-version --all-plugins --latest-version --all-accounts
  ```

  For more information on how to manage plugins using the Control Center CLI, see the in-app help under "Plugins".

- Log in to the Control Center GUI and go to the **Test Agents** view. Next to each Test Agent for which an upgrade is available, an up-arrow icon appears. Click that icon to go ahead with the upgrade.

Upgrading Postgres

In this section we will upgrade the `netrounds` and `paa-plugins` databases to use Postgres 14 instead of Postgres 10.

Work through the following steps:

- Verify that Postgres 10 is currently used:

  ```
  sudo -u postgres psql netrounds -c "SELECT version();"
  ```

  The expected output is:

  ```

  version
  ----------------------------------------------------------------------------------------------
  ----------------------------------------
   PostgreSQL 10.23 (Ubuntu 10.23-0ubuntu0.18.04.2) on x86_64-pc-linux-gnu, compiled by gcc
  (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0, 64-bit
  (1 row)
  ```

  If the above command reports Postgres 14, don't take any further actions.

- Stop Control Center servies:

  ```
  sudo ncc services stop
  ```

- Stop both Postgres clusters:

```
sudo -u postgres /usr/lib/postgresql/10/bin/pg_ctl -D /var/lib/postgresql/10/main stop
sudo -u postgres /usr/lib/postgresql/14/bin/pg_ctl -D /var/lib/postgresql/14/main stop

sudo systemctl stop postgresql
```

- Before doing the actual upgrade, run the following check:

```
cd /tmp

sudo -u postgres \
/usr/lib/postgresql/14/bin/pg_upgrade \
-b /usr/lib/postgresql/10/bin \
-B /usr/lib/postgresql/14/bin \
-d /etc/postgresql/10/main \
-D /etc/postgresql/14/main \
-p 5432 \
-P 5434 \
--check
```

  If the `pg_upgrade` check reports errors that are not easily fixable, contact Juniper technical support.

  If there are no errors and `pg_upgrade` reported that the clusters are compatible, you can proceed with the upgrade.

- Upgrade Postgres:

```
sudo -u postgres \
/usr/lib/postgresql/14/bin/pg_upgrade \
-b /usr/lib/postgresql/10/bin \
-B /usr/lib/postgresql/14/bin \
-d /etc/postgresql/10/main \
-D /etc/postgresql/14/main \
-p 5432 \
-P 5434
```

- Change the port assignment so that Postgres 10 uses port 5434 and Postgres 14 uses port 5432.

Update the value of port in `/etc/postgresql/14/main/postgresql.conf` to 5432:

```
sudo vim /etc/postgresql/14/main/postgresql.conf
```

Update the value of port in `/etc/postgresql/10/main/postgresql.conf` to 5434:

```
sudo vim /etc/postgresql/10/main/postgresql.conf
```

- Start the `postgresql` service:

```
sudo systemctl start postgresql
```

- Alter the password for the `netrounds` user:

```
sudo -u postgres psql -c "alter user netrounds with encrypted password 'netrounds';"
```

- Run the statistics optimizer:

```
sudo -u postgres /usr/lib/postgresql/14/bin/vacuumdb --all --analyze-in-stages
```

- Start Control Center services:

```
sudo ncc services start
```

- Verify that Postgres 14 is now being used for `netrounds` database:

```
sudo -u postgres psql netrounds -c "SELECT version();"
```

Expected output:

```

version
-----------------------------------------------------------------------------------------------
------------------------------------------
 PostgreSQL 14.8 (Ubuntu 14.8-0ubuntu0.22.04.1) on x86_64-pc-linux-gnu, compiled by gcc
```

```
(Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0, 64-bit
(1 row)
```

```
sudo DJANGO_SETTINGS_MODULE=netrounds.settings.local django-admin shell -c "from django.db
import connection; print(connection.cursor().connection.server_version)"
```

Expected output:

```
140008
```

- Verify that Control Center is functional and remove old cluster data:

```
sudo -u postgres sh delete_old_cluster.sh
```

- Optionally remove postgresql-10 and postgresql-12 client/server packages:

```
sudo apt purge postgresql-client-10 postgresql-10 postgresql-client-12 postgresql-12
```

Upgrading ConfD

If the ConfD service is installed in Control Center, do the following:

- Remove the existing version of `paa-netconf-yang` by running:

```
sudo apt-get remove paa-netconf-yang python3-pycryptodomex python3-pysnmp
```

- Follow the instructions in the Install ConfD section of the Installation Guide for version 4.2 (chapter Installing Control Center and Related Tasks

Troubleshooting

Password Authentication Failed For User

If the ncc migrate command fails with an error message

```
Failed to connect to database error="pq: password authentication failed for user
\"netrounds\"" db-host=localhost db-name=paa-plugins db-port=5432 ...
```

you must update the variable db-password in the /etc/netrounds/plugin.yaml file as explained in the warning above. Edit this file and then rerun ncc migrate.

Target WSGI Script Not Found

If you accidentally selected "N" for the Apache configuration files (see this step above) and got an error message like the one below

```
[wsgi:error] [pid 29401:tid 140567451211520] [client 127.0.0.1:37172] Target WSGI script not
found or unable to stat: /usr/lib/python2.7/dist-packages/netrounds/wsgi.py
```

run the following commands to get back on track:

```
export CC_VERSION=4.2.0.54
dpkg-deb --fsys-tarfile paa-webapp_${CC_VERSION}_all.deb | tar -x --wildcards ./etc/apache2/
sites-available/*.conf --strip-components 4
sudo mv netrounds*.conf /etc/apache2/sites-available/
sudo chown -R root:root /etc/apache2/sites-available/
sudo systemctl reload apache2
```

This overwrites the old configuration with the new one in the updated package.

Again, if you have installed proper SSL certificates (as recommended) instead of the default snakeoil ones, you will have to modify the file again to point to the correct path in the SSLCertificateFile and SSLCertificateKeyFile settings after the Debian package installation has completed. See the Installation Guide, chapter Service Configuration, section "SSL Certificate Configuration".

Same Origin Policy Disallows Reading the Remote Resource

This or some similar error may occur if you have set SITE_URL and SPEEDTEST_ADDRESS to different values in /etc/netrounds/netrounds.conf. You then need to change ALLOWED_ORIGINS in /etc/netrounds/restol.conf to allow both of these values in the `restol.conf` file. The simplest way to achieve this is to delete any value previously assigned to `ALLOWED_ORIGINS`. That setting will then get a default value which allows `SITE_URL` and `SPEEDTEST_ADDRESS` as found in `/etc/netrounds/netrounds.conf`.

openvpn@netrounds Service Fails to Find dh.pem File

If this happens, output like that below will be given:

```
Sep 04 09:56:47 ip-10-0-0-11 systemd[1]: Stopped OpenVPN connection to netrounds.
Sep 04 09:56:47 ip-10-0-0-11 systemd[1]: Starting OpenVPN connection to netrounds...
Sep 04 09:56:47 ip-10-0-0-11 ovpn-netrounds[1741772]: --cipher is not set. Previous OpenVPN
version defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need
this fallback please add '--data-ciphers-fallback BF-CBC' to your configuration and/or add BF-
CBC to --data-ciphers.
Sep 04 09:56:47 ip-10-0-0-11 ovpn-netrounds[1741772]: Options error: --dh fails with '/var/lib/
netrounds/openvpn/dh.pem': No such file or directory (errno=2)
Sep 04 09:56:47 ip-10-0-0-11 ovpn-netrounds[1741772]: Options error: Please correct these errors.
Sep 04 09:56:47 ip-10-0-0-11 ovpn-netrounds[1741772]: Use --help for more information.
Sep 04 09:56:47 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Main process exited,
code=exited, status=1/FAILURE
Sep 04 09:56:47 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Failed with result 'exit-
code'.
Sep 04 09:56:47 ip-10-0-0-11 systemd[1]: Failed to start OpenVPN connection to netrounds.
Sep 04 09:56:52 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Scheduled restart job,
restart counter is at 9.
Sep 04 09:56:52 ip-10-0-0-11 systemd[1]: Stopped OpenVPN connection to netrounds.
```

To resolve, run the following commands:

```
sudo mv /var/lib/netrounds/openvpn/certs_by_serial/dh1024.pem /var/lib/netrounds/openvpn/
sudo chown -R netrounds: /var/lib/netrounds/openvpn
sudo ncc services restart openvpn
```

openvpn@netrounds Service Reports Weak CA Key

In this case, the output will look like this:

```
Aug 30 12:49:34 ip-10-0-0-11 systemd[1]: Starting OpenVPN connection to netrounds...
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: --cipher is not set. Previous OpenVPN
version defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need
```

```
this fallback please add '--data-ciphers-fallback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: WARNING: file '/var/lib/netrounds/openvpn/private/server.key' is group or others accessible
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: OpenVPN 2.5.5 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jul 14 2022
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: NOTE: the current --script-security setting may allow this configuration to call user-defined scripts
Aug 30 12:49:34 ip-10-0-0-11 systemd[1]: Started OpenVPN connection to netrounds.
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: OpenSSL: error:0A00018E:SSL routines::ca md too weak
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: Cannot load certificate file /var/lib/netrounds/openvpn/issued/server.crt
Aug 30 12:49:34 ip-10-0-0-11 ovpn-netrounds[1848602]: Exiting due to fatal error
Aug 30 12:49:34 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Main process exited, code=exited, status=1/FAILURE
Aug 30 12:49:34 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Failed with result 'exit-code'.
Aug 30 12:49:39 ip-10-0-0-11 systemd[1]: openvpn@netrounds.service: Scheduled restart job, restart counter is at 14918.
Aug 30 12:49:39 ip-10-0-0-11 systemd[1]: Stopped OpenVPN connection to netrounds.
```

What you need to do is to modify the `/etc/openvpn/netrounds.conf` file by appending the line below at the end of the file:

```
tls-cipher "DEFAULT:@SECLEVEL=0"
```

Then restart the openpvn service:

```
sudo ncc services restart openvpn
```

Alternative Upgrade Procedure

IN THIS SECTION

This chapter describes a Control Center upgrade from 4.1 to 4.2 using a separate instance with Ubuntu 22.04 installed, as an alternative of the in-place upgrade (requiring multiple OS upgrades) described in the Upgrade Procedure chapter.

Packages Installed for 4.2

```
paa-control-center_4.2.0.54.tar.gz
paa-netconf-yang_4.2.0.54.tar.gz
paa-test-agent-application_4.2.0.9_all.deb
paa-test-agent-plugins_4.2.0.16_all.deb
paa-test-agent_4.2.0.11_all.deb
```

Steps on 4.1 Instance

Backing Up Product Data

```
sudo ncc services stop --without-timescaledb --with-zookeeper --with-kafka
sudo ncc backup
sudo ncc services start --without-timescaledb
```

The backup file is generated in the current working directory and named as follows:

```
paa-cc-backup-<PAA CC version>-yyyy-dd-mm_hh-MM-ss.tar.gz
```

Example:

```
paa-cc-backup-4.1.3.33-2023-16-05_10-21-13.tar.gz
```

was generated on May, 16th 2023 at 10:21:13.

Backing Up Plugin Service Data

Run this command:

```
sudo -u postgres pg_dump -t account \
-t user -t test_agent -t task \
-t measurement --data-only paa-plugins > paa_plugins_data.sql
```

Backing Up the SSL Certificates and Keys

NOTE: This procedure creates a TAR archive that might contain duplicates as the same cert/key file may be mentioned in multiple configuration files.

1. Create a TAR archive `paa_cc_certs.tar`:

   ```
   export TAR_ARCHIVE="paa_cc_certs.tar"
   sudo tar -cf "$TAR_ARCHIVE" --files-from /dev/null
   ```

2. Collect the secret key path:

   - Open the file `etc/netrounds/netrounds.conf`.
   - Copy the value for the `SECRET_KEY_FILE` setting without single/double quotes (`<SECRET_KEY_FILE value>`)
   - Paste the value you just obtained at the end of this command:

     ```
     sudo tar --append --file="$TAR_ARCHIVE" -C / <SECRET_KEY_FILE value>
     ```

For example, if the value of SECRET_KEY_FILE is /etc/netrounds/secret_key, the command will be:

```
sudo tar --append --file="$TAR_ARCHIVE" -C / /etc/netrounds/secret_key
```

NOTE: The output

```
tar: Removing leading `/' from member names
```

is for information only and can be ignored. The command will still add the file as expected.

3. Collect the certificate from the services configuration files:

- Open the file /etc/netrounds/consolidated.yaml.
- Copy the value <cert path> for both ssl-key and ssl-cert directives, but only if they are uncommented and have a name different from ssl-cert-snakeoil.

  Example:

  - Copy this one: /etc/certs/fullchain.pem
  - Don't copy this one: /etc/ssl/certs/ssl-cert-snakeoil.pem
- Paste the value you just copied at the end of this command:

```
sudo tar --append --file="$TAR_ARCHIVE" -C / <cert path>
```

NOTE: The output

```
tar: Removing leading `/' from member names
```

is for information only and can be ignored. The command will still add the file as expected.

- Repeat the steps above for the files /etc/netrounds/plugin.yaml and /etc/netrounds/test-agent-gateway.yaml.

4. Collect the certificates used by the Apache web server:

- Open the file /etc/apache2/sites-available/netrounds-restol-standalone.conf.

- Copy the value `<cert path>` for both `SSLCertificateFile` and `SSLCertificateKeyFile` directives, but only if they are uncommented and have a name different from `ssl-cert-snakeoil`.
- Paste the value you just copied at the end of this command:

```
sudo tar --append --file="$TAR_ARCHIVE" -C / <cert path>
```

NOTE: The output

```
tar: Removing leading `/' from member names
```

is for information only and can be ignored. The command will still add the file as expected.

- Repeat the steps above for the file `/etc/apache2/sites-available/netrounds-ssl.conf`.

5. Copy the backup files to the Ubuntu 22.04 instance:

```
paa-cc-backup-<PAA CC version>-yyyy-dd-mm_hh-MM-ss.tar.gz
paa_cc_certs.tar
paa_plugins_data.sql
```

Steps on 4.2 Instance

Installing Required OS and Software

Follow the Paragon Active Assurance 4.2 Installation Guide, chapter "Installing Required OS and Software".

Restoring Product Backup from Data

1. Drop the main PostgreSQL database and plugin database:

```
sudo -u postgres psql -c "DROP DATABASE IF EXISTS netrounds;"
```

```
sudo -u postgres psql -c 'DROP DATABASE IF EXISTS "paa-plugins";'
```

2. Recreate the main PostgreSQL database and plugin database as described in the 4.2 Operations Guide:

```
sudo -u postgres psql -c "CREATE DATABASE netrounds OWNER netrounds ENCODING 'UTF8' TEMPLATE
'template0';"
sudo -u postgres psql -c 'CREATE DATABASE "paa-plugins" OWNER netrounds ENCODING "UTF8"
TEMPLATE "template0";'
```

3. Uncompress the backup archive and enter into the backup archive directory:

```
tar -xzf paa-cc-backup-<PAA CC version/>-yyyy-dd-mm_hh-MM-ss.tar.gz
cd ./paa-cc-backup-<PAA CC version>-yyyy-dd-mm_hh-MM-ss
```

4. Restore the main PostgreSQL database and plugin database:

```
sudo -u postgres psql --set ON_ERROR_STOP=on netrounds < paa_cc_postgres.sql
sudo -u postgres psql --set ON_ERROR_STOP=on paa-plugins < paa_cc_plugins.sql
```

5. Restore plugin signing keys:

```
sudo tar -xzf paa_cc_plugin_keys.tar.gz -C /
```

6. Restore Control Center configurations:

```
sudo tar -xzf paa_cc_configs.tar.gz -C /
```

7. Restore Control Center SSL certificates and secret key:

```
cd ..
sudo tar -xf paa_cc_certs.tar -C /
```

Downloading Control Center and Test Agent Repositories

Follow the 4.2 Installation Guide.

Installing Control Center and Related Tasks

1. Install Control Center

   Follow the procedure in the 4.2 Installation Guide. Below are some specifics of updating configuration files:

   - For `/etc/netrounds/restol.conf`, press `N`.
   - For `/etc/netrounds/netrounds.conf`, press `D`, then `Z`.
     - Back up the current version:

       ```
       cp /etc/netrounds/netrounds.conf /etc/netrounds/netrounds-41.conf
       ```

     - Edit `/etc/netrounds/netrounds.conf` as follows:
       - Replace `AXES_LOGIN_FAILURE_LIMIT` with `AXES_FAILURE_LIMIT`.
       - Replace `X_FRAME_OPTIONS_HEADER` with `X_FRAME_OPTIONS`
       - Replace this

         ```
         # Number of days a password reset link is valid.
         PASSWORD_RESET_TIMEOUT_DAYS = x
         ```

         with this:

         ```
         # Number of seconds a password reset link is valid.
         # This is equal to x * 24 * 60 * 60, where x is the number of days. Enter the
         calculated value.
         PASSWORD_RESET_TIMEOUT =
         ```

         where `x` is the value of `PASSWORD_RESET_TIMEOUT_DAYS`.
     - Compare the two versions of the file:

       ```
       diff /etc/netrounds/netrounds-41.conf /etc/netrounds/netrounds.conf
       ```

     - Type `exit` and validate current changes with `N`.
   - For `/etc/apache2/sites-available/netrounds-ssl.conf`, press `D`, then `Z`.

- - Back up the current version:

    ```
    cp /etc/apache2/sites-available/netrounds-ssl.conf \
    /etc/apache2/sites-available/netrounds-ssl-41.conf
    ```

  - Replace the Python 3.6 path with one for Python 3.10:

    ```
    sed -i 's/python3.6/python3.10/g' /etc/apache2/sites-available/netrounds-ssl.conf
    ```

  - Compare the two versions of the file:

    ```
    diff /etc/apache2/sites-available/netrounds-ssl-41.conf \
    /etc/apache2/sites-available/netrounds-ssl.conf
    ```

  - Type exit and validate current changes with `N`.

- For `/etc/apache2/sites-available/netrounds.conf`, press `D`, then `Z`.

  - Back up the current version:

    ```
    cp /etc/apache2/sites-available/netrounds.conf \
    /etc/apache2/sites-available/netrounds-41.conf
    ```

  - Replace the Python 3.6 path with one for Python 3.10:

    ```
    sed -i 's/python3.6/python3.10/g' /etc/apache2/sites-available/netrounds.conf
    ```

  - Compare the two versions of the file:

    ```
    diff /etc/apache2/sites-available/netrounds-41.conf \
     /etc/apache2/sites-available/netrounds.conf
    ```

  - Type `exit` and validate current changes with `N`.

- For `/etc/openvpn/netrounds.conf`, press `D`, then `Z`.

  - Update the file to have:

        - the value of `cert` pointing to the same name CRT file but under `openvpn/issued`

        - the value of `key` pointing to the same name KEY file but under `openvpn/private`.

        Type `exit` and validate current changes with `N`.

    - For `/etc/netrounds/test-agent-gateway.yaml`, press `N`.

    - For `/etc/netrounds/metrics.yaml`, press `N`.

2. Restore post-installation config
   Run the following commands:

```
cd ./paa-cc-backup-<PAA CC version>-yyyy-dd-mm_hh-MM-ss
sudo tar -xzf paa_cc_plugins.tar.gz -C /
sudo tar -xzf paa_cc_license.tar.gz -C /
sudo tar -xzf paa_cc_rrd.tar.gz -C /
sudo tar -xzf paa_cc_openvpn.tar.gz -C /
```

3. Run the database migration (and change ownership for one file)
   Run the following commands:

```
sudo ncc migrate
sudo chown -R netrounds: /var/lib/netrounds/openvpn
sudo ncc services restart
sudo -u postgres psql paa-plugins < paa_plugins_data.sql
```

4. Install the Test Agent repositories and plugins
   Run the following commands:

```
export TA_APPLIANCE_BUILD=4.2.0.X
export TA_APPLICATION_BUILD=4.2.0.X
export PLUGIN_BUILD=4.2.0-X

# Compute SHA256 checksums for the repositories and verify that they match the
# SHA256 checksums provided on the download page
sha256sum paa-test-agent_${TA_APPLIANCE_BUILD}_all.deb
sha256sum paa-test-agent-application_${TA_APPLICATION_BUILD}_all.deb
sha256sum paa-test-agent-plugins_${PLUGIN_BUILD}_all.deb

# Start the installation
```

```
sudo apt-get install ./paa-test-agent_${TA_APPLIANCE_BUILD}_all.deb
sudo apt-get install ./paa-test-agent-application_${TA_APPLICATION_BUILD}_all.deb
sudo apt-get install ./paa-test-agent-plugins_${PLUGIN_BUILD}_all.deb

# Restart the services
sudo ncc services restart
```

Do the following to enable the latest version of all plugins in all accounts:

```
ncc plugins edit enabled-version --all-plugins --latest-version --all-accounts
```

For more information on how to manage plugins using the Control Center CLI, see the in-app help under "Plugins".

Log in to the Control Center GUI and go to the **Test Agents** view. Next to each Test Agent for which an upgrade is available, an up-arrow icon appears. Click that icon to go ahead with the upgrade.

5. If the ConfD service was enabled in 4.1, install ConfD. Follow the instructions in the Installation Guide, chapter Installing Control Center and Related Tasks, section "Installing ConfD", with one modification: The installation of ConfD from deb packages needs to be done with the command

```
sudo CONFD_FORCE_CC_CONFIGURE="true"  apt-get install ./paa-netconf-yang_4.2.0.0/*.deb
```

All other commands in the Installation Guide should be applied unchanged.

NOTE: Some additional configuration is needed; again, see the Installation Guide for details.

Configuring the Metrics Service

If the metrics service was enabled in 4.1, the following steps need to be applied in order to restore the data collected in the TimescaleDB database.

1. Enable and start the timescaledb service:

```
sudo ncc services enable timescaledb
sudo ncc services start timescaledb
```

2. Restore TimescaleDB from backup:

```
cd <backup folder location>/paa-cc-backup-<PAA CC version>-yyyy-dd-mm_hh-MM-ss
 sudo tar zxf paa_cc_timescaledb.tar.gz \
      -C /var/lib/netrounds/rrd/timescaledb/pgbackrest/repo
 sudo /etc/netrounds/backup-restore-timescaledb.sh \
      --restore paa-cc-timescaledb/data-yyyy-dd-mm_hh-MM-ss
```

3. Enable and start the metrics service:

```
sudo ncc services enable metrics
 sudo ncc services start metrics
```

Rollback in Case of Failed Upgrade

If a Control Center upgrade fails, here is how to return the system to its state immediately before the upgrade:

- Make a clean Ubuntu installation according to the Installation Guide, chapter Installing Required OS and Software.
- Install the version of Control Center that you were using before the upgrade. Again, follow the Installation Guide, chapter Installing Control Center and Related Tasks.
- Recover your data from backup as explained in the Operations Guide, chapter Restoring Product Data from Backup.

Applying Ubuntu Updates

NOTE: If you want to apply updates to Ubuntu 22.04, you first need to hold rrd packages with the command

```
sudo apt-mark hold librrd8 python3-rrdtool rrdtool
```

Then you can apply the updates with

```
sudo apt dist-upgrade
```

The hold command is necessary because the operating system will by default *remove* Paragon Active Assurance packages:

(output from `sudo apt dist-upgrade` below)

```
...
The following packages will be REMOVED:
paa-callexecuter paa-common paa-license-daemon paa-test-agent-compat paa-test-agent-login paa-
webapp
The following packages will be upgraded:
librrd8 python3-rrdtool rrdtool
3 upgraded, 0 newly installed, 6 to remove and 0 not upgraded.
...
```