

# Upgrading Control Center from Version 2.34

Published  
2022-06-08

RELEASE  
3.2

# Table of Contents

**Introduction**

**Scenario A: Upgrade of Ubuntu 16.04 to Ubuntu 18.04**

**Scenario B: Fresh Ubuntu 18.04 Installation**

**Troubleshooting**

# Introduction

This document concerns upgrading of Paragon Active Assurance Control Center from version 2.34 to a later version.

The upgrade entails special procedures as it involves upgrading the Ubuntu OS from 16.04 to 18.04. The document covers two scenarios:

- Upgrade of Ubuntu 16.04 (with Control Center installed) to Ubuntu 18.04.
- Fresh installation of Ubuntu 18.04 followed by installation of Control Center and transfer of backup data from an old Control Center instance to the new instance.

For other upgrades, please refer to the [Upgrade Guide](#).

## Scenario A: Upgrade of Ubuntu 16.04 to Ubuntu 18.04

- Begin by disabling the `apache2` and `netrounds-callexecuter` services:

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

- Stop all Paragon Active Assurance services:

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

- Take backups of Paragon Active Assurance product data.

**NOTE:** This is the backup procedure described in the Lifecycle Management Guide, chapter [Backing Up Product Data](#), only more briefly worded.

Run these commands:

```
# Back up the PostgreSQL database
pg_dump --help
pg_dump -h localhost -U netrounds netrounds > ncc_postgres.sql

# (Alternatively, to save in binary format:)
# pg_dump -h localhost -U netrounds -Fc netrounds > ncc_postgres.binary

# Back up OpenVPN keys
sudo tar -czf ncc_openvpn.tar.gz /var/lib/netrounds/openvpn

# Note: Be sure to store these in a safe place.

# Back up RRD files (metrics data)
# Check the file size before compressing the RRDs. Use of the tar command is not
# recommended if the RRDs are larger than 50 GB; see note below.
du -hs /var/lib/netrounds/rrd
sudo tar -czf ncc_rrd.tar.gz /var/lib/netrounds/rrd
```

**NOTE:** The `pg_dump` command will ask for a password which can be found in `/etc/netrounds/netrounds.conf` under "postgres database". The default password is "netrounds".

**NOTE:** For a large-scale setup (> 50 GB), making a tarball of the RRD files might take too long, and taking a snapshot of the volume can be a better idea. Possible solutions for doing this include: using a file system that supports snapshots, or taking a snapshot of the virtual volume if the server is running in a virtual environment.

- Check the integrity of the database using the supplied script `netrounds_2.35_validate_db.sh`.



**WARNING:** If this script outputs warnings, **do not** attempt the database migration procedure described "below" on page 4. Contact Juniper support by filing a ticket at <https://support.juniper.net/support/requesting-support> (supplying the output from the script) to have the problems with the database resolved before you proceed with the upgrade.

- Take backups of the Control Center configuration files:

- /etc/apache2/sites-available/netrounds-ssl.conf
- /etc/apache2/sites-available/netrounds.conf
- /etc/netrounds/netrounds.conf
- /etc/netrounds/probe-connect.conf
- /etc/netrounds/restol.conf
- /etc/netrounds/secret\_key
- /etc/netrounds/test-agent-gateway.yaml
- /etc/openvpn/netrounds.conf

For example:

```
sudo cp /etc/apache2/sites-available/netrounds-ssl.conf /etc/apache2/sites-available/
netrounds-ssl.conf.old
```

- Upgrade Ubuntu to version 18.04. A typical upgrade procedure is as follows (adapted from <https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes>):
  - To upgrade on a server system:
    - Install update-manager-core if it is not already installed.
    - Make sure the Prompt line in /etc/update-manager/release-upgrades is set to 'lts' (to ensure that the OS is upgraded to the 18.04, the next LTS version after 16.04).
    - Launch the upgrade tool with the command `sudo do-release-upgrade`.
    - Follow the on-screen instructions. As far as Paragon Active Assurance is concerned, you can keep the defaults throughout. (It may of course happen that you need to make different choices for reasons unrelated to Paragon Active Assurance.)
- Once Ubuntu has been upgraded, reboot the system. Then perform the following steps:
- Upgrade PostgreSQL.
  - Update PostgreSQL database files from version 9.5 to version 10:

```
sudo pg_dropcluster 10 main --stop    # Shut down server and completely delete cluster#
"main" version 10 (this prepares for the upgrade# in the next command)
sudo pg_upgradecluster 9.5 main      # Upgrade cluster "main" version 9.5 to latest#
```

```
available version (10)
sudo pg_dropcluster 9.5 main          # Completely delete cluster "main" version 9.5
```

- Remove the outdated version of PostgreSQL:

```
sudo apt purge postgresql-9.5 postgresql-client-9.5 postgresql-contrib-9.5
```

- Update Paragon Active Assurance packages.
- Compute the checksum for the tarball containing the new Control Center version and verify that it is equal to the SHA256 checksum provided on the download page:

```
sha256sum paa-control-center_${CC_VERSION}.tar.gz
```

- Unpack the Control Center tarball:

```
export CC_VERSION=<enter new version here>
tar -xzf netrounds-control-center_${CC_VERSION}.tar.gz
```

- Install new Control Center packages:

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

- Remove the obsolete packages:

**NOTE:** It is vital to remove these packages.

```
# Test Agent Lite support
sudo apt purge netrounds-agent-login

# Unsupported jsonfield package
sudo apt remove python-django-jsonfield
```

- Run the database migration:

**NOTE:** Before doing the migration, you must ensure that the database integrity check described ["above" on page 2](#) completes without error.

```
sudo ncc migrate
```

The `ncc migrate` command takes considerable time to execute (many minutes). It should print the following (details omitted below):

```
Migrating database...
Operations to perform:
  <...>
Synchronizing apps without migrations:
  <...>
Running migrations:
  <...>
Creating cache table...
<...>
Syncing test scripts...
<Updating script ...>
```

- *(Optional)* Update the ConfD package if you need ConfD:

```
tar -xzf netrounds-confd_${NCC_VERSION}.tar.gz
sudo apt install ./netrounds-confd_${NCC_VERSION}_all.deb
```

- Compare the previously backed-up configuration files with the newly installed ones, and manually merge the contents of the two sets of files (they should remain in the same locations).
- Enable the `apache2`, `kafka`, and `netrounds-callexecuter` services:

```
sudo systemctl enable apache2 kafka netrounds-callexecuter
```

- Start Paragon Active Assurance services:

```
sudo systemctl start --all "netrounds-*" apache2 kafka openvpn@netrounds
```

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

```
sudo systemctl reload apache2
```

- Install new Test Agent repositories:

```
TA_APPLIANCE_VERSION=<enter version number here>
TA_APPLICATION_VERSION=<enter version number here>

# For versions prior to 3.0:
# Verify the integrity of the repositories (response should be "OK")
shasum -c netrounds-test-agent_${TA_APPLIANCE_VERSION}_all.sha256
shasum -c netrounds-test-agent-application_${TA_APPLICATION_VERSION}.sha256.sum

# For version 3.0 and later:
# Compute checksums for the repositories and verify that they match the
# SHA256 checksums provided on the download page
sha256sum paa-test-agent_${TA_APPLIANCE_VERSION}_all.deb
sha256sum paa-test-agent-application_${TA_APPLICATION_VERSION}.tar.gz

# Start the installation
sudo apt-get install \
./netrounds-test-agent_${TA_APPLIANCE_VERSION}_all.deb

sudo cp netrounds-test-agent-application_${TA_APPLICATION_VERSION}.tar.gz \
/usr/lib/python2.7/dist-packages/netrounds/static/test_agent/
```

- Since support for Test Agent Lite was dropped in version 2.35, you should remove the old Test Agent Lite packages if you have them installed:

```
sudo rm -rf /usr/lib/python2.7/dist-packages/netrounds/static/test_agent/netrounds-test-agent-
lite*
```

## Scenario B: Fresh Ubuntu 18.04 Installation

- On the Ubuntu 16.04 instance, take backups of Paragon Active Assurance product data.



**NOTE:** This is the backup procedure described in the Lifecycle Management Guide, chapter "Backing Up Product Data", only more briefly worded.

Run these commands:

```
# Back up the PostgreSQL database
pg_dump --help
pg_dump -h localhost -U netrounds netrounds > ncc_postgres.sql

# (Alternatively, to save in binary format:)
# pg_dump -h localhost -U netrounds -Fc netrounds > ncc_postgres.binary

# Back up OpenVPN keys
sudo tar -czf ncc_openvpn.tar.gz /var/lib/netrounds/openvpn
# Note: Be sure to store these in a safe place.

# Back up RRD files (metrics data)
# Check the file size before compressing the RRDs. Use of the tar command is not
# recommended if the RRDs are larger than 50 GB; see note below.
du -hs /var/lib/netrounds/rrd
sudo tar -czf ncc_rrd.tar.gz /var/lib/netrounds/rrd
```

**NOTE:** The `pg_dump` command will ask for a password which can be found in `/etc/netrounds/netrounds.conf` under "postgres database". The default password is "netrounds".

**NOTE:** For a large-scale setup (> 50 GB), making a tarball of the RRD files might take too long, and taking a snapshot of the volume can be a better idea. Possible solutions for doing this include: using a file system that supports snapshots, or taking a snapshot of the virtual volume if the server is running in a virtual environment.

- On the Ubuntu 16.04 instance, take backups of the Control Center configuration files:
  - `/etc/apache2/sites-available/netrounds-ssl.conf`
  - `/etc/apache2/sites-available/netrounds.conf`
  - `/etc/netrounds/netrounds.conf`

- /etc/netrounds/probe-connect.conf
- /etc/openvpn/netrounds.conf

For example:

```
sudo cp /etc/apache2/sites-available/netrounds-ssl.conf /etc/apache2/sites-available/
netrounds-ssl.conf.old
```

- On the Ubuntu 16.04 instance, back up the license file.
- The new instance needs to satisfy at least the same hardware requirements as the old one.
- On the new instance, install Ubuntu 18.04. We recommend the following tutorial:
  - <https://ubuntu.com/tutorials/install-ubuntu-server>

As far as Paragon Active Assurance is concerned, you can keep the defaults throughout. (It may of course happen that you need to make different choices for reasons unrelated to Paragon Active Assurance.)

- Once Ubuntu 18.04 is installed, reboot the system.
- The following disk partitioning is recommended, especially for snapshot backups (but it is up to you as a user to decide):
  - Recommended partitioning for lab setup:
    - /: Whole disk, ext4.
  - Recommended partitioning for production setup:
    - /: 10% of disk space, ext4.
    - /var: 10% of disk space, ext4.
    - /var/lib/netrounds/rrd: 80% of disk space, ext4.
  - No encryption
- Set the time zone to UTC, for example as follows:

```
sudo timedatectl set-timezone Etc/UTC
```

- Set all locales to en\_US.UTF-8.

- One way to do this is to manually edit the file `/etc/default/locale`. Example:

```
LANG=en_US.UTF-8
LC_ALL=en_US.UTF-8
LANGUAGE=en_US.UTF-8
```

- Make sure the following line is NOT commented out in the `/etc/locale.gen`:

```
en_US.UTF-8 UTF-8
```

- Regenerate the locale files to make sure the selected language is available:

```
sudo apt-get install locales
sudo locale-gen
```

- Make sure that traffic on the following ports are allowed to and from Control Center:
  - Inbound:
    - TCP port 443 (HTTPS): Web interface
    - TCP port 80 (HTTP): Web interface (used by Speedtest, redirects other URLs to HTTPS)
    - TCP port 830: ConfD (optional)
    - TCP port 6000: Encrypted OpenVPN connection for Test Agent Appliances
    - TCP port 6800: Encrypted WebSocket connection for Test Agent Applications
  - Outbound:
    - TCP port 25 (SMTP): Mail delivery
    - UDP port 162 (SNMP): Sending SNMP traps for alarms
    - UDP port 123 (NTP): Time synchronization
- Install NTP:
  - First disable `timedatectl`:

```
sudo timedatectl set-ntp no
```

- Run this command:

```
timedatectl
```

and verify that

```
systemd-timesyncd.service active: no
```

- Now you can run the NTP installation:

```
sudo apt-get install ntp
```

- Make sure that the configured NTP servers are reachable:

```
ntpq -np
```

The output should normally be "all ones" expressed in octal.<sup>1</sup>

- Install PostgreSQL and set up a user for Control Center:

```
sudo apt-get update
```

```
sudo apt-get install postgresql
```

```
sudo -u postgres psql -c "CREATE ROLE netrounds WITH ENCRYPTED PASSWORD 'netrounds' SUPERUSER LOGIN;"
```

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

Using an external PostgreSQL server is not recommended.

- Install and configure an email server.

<sup>1</sup> In the output, the "reach" value for the NTP servers is an octal value indicating the outcome of the last eight NTP transactions. If all eight were successful, the value will be octal 377 (= binary 0b11111111). However, when you have just installed NTP, it is likely that fewer than eight NTP transactions have occurred, so that the value will be smaller: one of 1, 3, 7, 17, 37, 77, or 177 if all transactions were successful.

- Control Center will send emails to users:
  - when they are invited to an account,
  - when sending email alarms (i.e. if email rather than SNMP is used for this purpose), and
  - when sending periodic reports.
- Run the command

```
sudo apt-get install postfix
```

- For a simple setup where postfix can send directly to the destination email server, you can set **General type of mail configuration** to "Internet Site", and **System mail name** can usually be left as-is. Otherwise, postfix needs to be configured according to the environment. For guidance, refer to the official Ubuntu documentation at <https://help.ubuntu.com/lts/serverguide/postfix.html>.
- Install Control Center on the Ubuntu 18.04 instance.

This procedure also installs the Paragon Active Assurance REST API.

```
export CC_VERSION=<enter new version here>

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

# Unpack the tarball
tar -xzf netrounds-control-center_${CC_VERSION}.tar.gz

# Make sure packages are up to date
sudo apt-get update

# Start the installation
sudo apt-get install ./netrounds-control-center_${CC_VERSION}/*.deb
```

- Stop all Paragon Active Assurance services:

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

- Restore database backup:

```
sudo -u postgres psql --set ON_ERROR_STOP=on netrounds < ncc_postgres.sql
```

- Run the database migration:

**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` or `tmux` so that the migrate command will continue running even if the ssh session breaks.

```
sudo ncc migrate
```

The `ncc migrate` command takes considerable time to execute (many minutes). It should print the following (details omitted below):

```
Migrating database...
Operations to perform:
  <...>
Synchronizing apps without migrations:
  <...>
Running migrations:
  <...>
Creating cache table...
  <...>
Syncing test scripts...
<Updating script ...>
```

- Transfer the backup data to the 18.04 instance using `scp` or some other tool.
- Restore the OpenVPN keys:

```
# Remove any existing OpenVPN keys
sudo rm -rf /var/lib/netrounds/openvpn

# Unpack the backed-up keys
sudo tar -xzf ncc_openvpn.tar.gz -C /
```

- Restore RRD data:

```
# Remove any existing RRDs
sudo rm -rf /var/lib/netrounds/rrd

# Unpack the backed-up RRDs
sudo tar -xzf ncc_rrd.tar.gz -C /
```

- Compare the backed-up configuration files with the newly installed ones, and manually merge the contents of the two sets of files (they should remain in the same locations).
- Activate the product license using the license file taken from the old instance:

```
ncc license activate ncc_license.txt
```

- Start Paragon Active Assurance services:

```
sudo systemctl start --all "netrounds-*" apache2 kafka openvpn@netrounds
```

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

```
sudo systemctl reload apache2
```

- Install new Test Agent repositories:

```
TA_APPLIANCE_VERSION=<enter version number here>
TA_APPLICATION_VERSION=<enter version number here>

# For versions prior to 3.0:
# Verify the integrity of the repositories (response should be "OK")
shasum -c netrounds-test-agent_${TA_APPLIANCE_VERSION}_all.sha256
shasum -c netrounds-test-agent-application_${TA_APPLICATION_VERSION}.sha256.sum

# For version 3.0 and later:
# Compute checksums for the repositories and verify that they match the
# SHA256 checksums provided on the download page
sha256sum paa-test-agent_${TA_APPLIANCE_VERSION}_all.deb
sha256sum paa-test-agent-application_${TA_APPLICATION_VERSION}.tar.gz
```

```
# Start the installation
sudo apt-get install \
./netrounds-test-agent_${TA_APPLIANCE_VERSION}_all.deb

sudo cp netrounds-test-agent-application_${TA_APPLICATION_VERSION}.tar.gz \
/usr/lib/python2.7/dist-packages/netrounds/static/test_agent/
```

- (Optional) Follow the [NETCONF & YANG API Orchestration Guide](#) to install and configure ConfD if you need it.

## Troubleshooting

### IN THIS SECTION

- [Problems Starting ConfD | 14](#)
- [Problems Starting callexecuter | 14](#)
- [Web Server Does Not Respond | 15](#)
- [Restarting of Paragon Active Assurance Services Fails | 17](#)

### Problems Starting ConfD

If you have problems starting ConfD after the upgrade, please contact your Juniper partner or your local Juniper account manager or sales representative in order to get a new subscription.

### Problems Starting callexecuter

Check the callexecuter logs with the command

```
sudo journalctl -xeu netrounds-callexecuter
```



You may see an error like the following:

```
Jun 03 09:53:27 myhost django-admin[6290]: ERROR netrounds.manager.callexecuter Unhandled
exception in CallExecuter.run [name=netrounds.manager.callexecuter, thread=140364632504128,
process=8238, funcName=handle, le
Jun 03 09:53:27 myhost django-admin[6290]: Traceback (most recent call last):
Jun 03 09:53:27 myhost django-admin[6290]:   File "debian/tmp/usr/lib/python2.7/dist-packages/
netrounds/manager/management/commands/runcallexecuter.py", line 65, in handle
Jun 03 09:53:27 myhost django-admin[6290]:   File "debian/tmp/usr/lib/python2.7/dist-packages/
netrounds/manager/calldispatcher.py", line 164, in run
Jun 03 09:53:27 myhost django-admin[6290]:   File "debian/tmp/usr/lib/python2.7/dist-packages/
netrounds/manager/models.py", line 204, in inwait
Jun 03 09:53:27 myhost django-admin[6290]:   File "debian/tmp/usr/lib/python2.7/dist-packages/
netrounds/manager/models.py", line 42, in __unicode__
Jun 03 09:53:27 myhost django-admin[6290]: AttributeError: 'unicode' object has no attribute
'iteritems'
```

What has happened is that the `netrounds-callexecuter*.deb` package was upgraded without making sure the `netrounds-callexecuter` systemd service was stopped and disabled. The database is in the wrong state; it needs to be restored from backup, and the upgrade needs to be repeated.

Do the following to disable and stop the `netrounds-callexecuter` service:

```
sudo systemctl disable netrounds-callexecuter
sudo systemctl stop netrounds-callexecuter
```

## Web Server Does Not Respond

Check the apache logs with the command

```
tail -n 50 /var/log/apache2/netrounds_error.log
```

If you see the following error, it means that Control Center version 2.34 is running on Ubuntu 18.04, that is, Control Center has not been successfully upgraded. The solution is to upgrade Control Center to a later version as described in this document.

```
# Timestamps, pids, etc. stripped away below
```

```
Target WSGI script '/usr/lib/python2.7/dist-packages/netrounds/wsgi.py' cannot be loaded as Python module.
```

```
Exception occurred processing WSGI script '/usr/lib/python2.7/dist-packages/netrounds/wsgi.py'.
```

```
Traceback (most recent call last):
```

```
File "/usr/lib/python2.7/dist-packages/netrounds/wsgi.py", line 6, in<module>
```

```
    application = get_wsgi_application()
```

```
File "/usr/lib/python2.7/dist-packages/django/core/wsgi.py", line 13, in get_wsgi_application
```

```
    django.setup(set_prefix=False)
```

```
File "/usr/lib/python2.7/dist-packages/django/__init__.py", line 27, in setup
```

```
    apps.populate(settings.INSTALLED_APPS)
```

```
File "/usr/lib/python2.7/dist-packages/django/apps/registry.py", line 85, in populate
```

```
    app_config = AppConfig.create(entry)
```

```
File "/usr/lib/python2.7/dist-packages/django/apps/config.py", line 94, in create
```

```
    module = import_module(entry)
```

```
File "/usr/lib/python2.7/importlib/__init__.py", line 37, in import_module
```

```
    __import__(name)
```

```
File "/usr/lib/python2.7/dist-packages/grappelli/dashboard/__init__.py", line 1, in<module>
```

```
    from grappelli.dashboard.dashboards import *
```

```
File "/usr/lib/python2.7/dist-packages/grappelli/dashboard/dashboards.py", line 14, in<module>
```

```
    from grappelli.dashboard import modules
```

```
File "/usr/lib/python2.7/dist-packages/grappelli/dashboard/modules.py", line 9, in<module>
```

```
    from django.contrib.contenttypes.models import ContentType
```

```
File "/usr/lib/python2.7/dist-packages/django/contrib/contenttypes/models.py", line 139,
```

```
in<module>
```

```
    class ContentType(models.Model):
```

```
File "/usr/lib/python2.7/dist-packages/django/db/models/base.py", line 110, in __new__
```

```
    app_config = apps.get_containing_app_config(module)
```

```
File "/usr/lib/python2.7/dist-packages/django/apps/registry.py", line 247, in
```

```
get_containing_app_config
```

```
    self.check_apps_ready()
```

```
File "/usr/lib/python2.7/dist-packages/django/apps/registry.py", line 125, in check_apps_ready
```

```
    raise AppRegistryNotReady("Apps aren't loaded yet.")
```

```
AppRegistryNotReady: Apps aren't loaded yet.
```

## Restarting of Paragon Active Assurance Services Fails

Restarting the `netrounds-*` services with

```
sudo systemctl start --all "netrounds-*" apache2 openvpn@netrounds
```

produces the following message:

```
Failed to start netrounds-agent-ws-server.service: Unit netrounds-agent-ws-server.service is masked.  
Failed to start netrounds-agent-daemon.service: Unit netrounds-agent-daemon.service is masked.
```

This means that the services mentioned have been masked in the course of the package removal process and require manual cleanup. The cleanup procedure is shown below:

```
sudo apt-get purge netrounds-agent-login  
sudo find /etc/systemd/system -name "netrounds-agent-*.service" -delete  
sudo systemctl daemon-reload
```