

# Migrate from 4.5.x to 5.x

- [What will be migrated](#)
- [Pre-requisites](#)
- [Prepare New Appliance](#)
- [Prepare Old Appliance](#)
- [Config Migration](#)
- [Database Migration](#)
- [Start GL on new appliance](#)

The stable release of CentOS 7 has reached its end of life. Therefore it was needed to move GreenLight to a new Operating System.

Because of this, AlmaLinux distribution has been chosen. *AlmaLinux OS is an open-source, community-driven Linux operating system that fills the gap left by the discontinuation of the CentOS Linux stable release.*



- GreenLight 5 comes as a new virtual appliance
- Upgrading from 4.5.x to v5 release is a side-by-side migration (no in-place upgrade possible)

## What will be migrated

- Database
- Notes User ID and password file
- names.nsf
- license file
- general container config files

## Pre-requisites

On **New Appliance**, please make sure that the following is checked and prepared:



- Setup the new appliance and perform make sure that the basic setup tasks have been performed: *Hostname, DNS, partitioning, ...*
- Database Size:  
*/opt/panagenda/pgdata* should be the same size at least, but will need more than on the old appliance, since the backup file will be stored there as well. [How to adjust partitions](#)
  - Rule of thumb: [new pgdata disk required free] = [old pgdata disk used] \* 1.3
- */opt/panagenda/appdata* should be the same size

On **Old Appliance**, please make sure that the following items checked



- Make sure you have uploaded the new GreenLight license file (the file will be automatically copied over during the migration). Please contact [support@panagenda.com](mailto:support@panagenda.com) in order to get an updated license file.
- Make sure you have at least GreenLight Version  $\geq$  4.5.x installed

## Prepare New Appliance

The new appliance should most likely use much of the same fundamental container config as the old one.

```
# Old Appliance: check for custom config (e.g. CORS settings, ...)
cat /opt/panagenda/appdata/gl/config
```

This file may not exist, please create it on the new appliance if necessary. Attention on host name change!

```
# New Appliance: update custom config based on entries from the old appliance
vim /opt/panagenda/appdata/gl/config
```

You might stumble across a mistake here that we previously had in our config. Docker ignored it in the past, but now flags it and our app won't start. Please check for this and correct!

- 1) These network config parameters are optional and will not be there in most customer installations. Our default subnets are 192.168.237.0 and 192.168.238.0 and they will come out of the config.default file. Should this subnet config exist in customer environments though, we need to correct the subnet format and bring the two parameters along (DOCKER\_BRIDGE + NETWORK\_SUBNET)
- 2) There might also be a parameter PANAGENDA\_DOCKER\_STORAGE\_DRIVER present. **DO NOT copy this over.** Just leave the whole line out. The new appliance has a much newer docker version with a new storage driver for performance and OS compatibility.

```
[root@ifa-dev-01 ~]#
[root@ifa-dev-01 ~]# cat /opt/panagenda/appdata/ifa/config
PANAGENDA_TIMEZONE=Europe/Berlin
PANAGENDA_HOSTNAME=ifa-dev-01.panagenda.local
PANAGENDA_DOCKER_BRIDGE=172.17.0.1/16
PANAGENDA_COMPOSE_NETWORK_SUBNET=172.18.0.1/16
PANAGENDA_DOCKER_STORAGE_DRIVER=overlay
[root@ifa-dev-01 ~]#
```

```
# New Appliance: bring config changes into effect
gl start
```

Check if everything comes up properly and the containers run normally (docker ps). If it looks good, you can check the postgres container if the subnet settings have been applied successfully.

```
# New Appliance: check postgres container
docker inspect gl_postgres
```

Near the end of the output you should see something like the following: "IPv4Address": "172.18.0.2"

Here the subnet config and the hosts individual IP come together.

```
# Stop all containers, except postgres
docker stop gl_nginx gl_tomcat gl_data_miner gl_metabase gl_cron -t 120
```

## Prepare Old Appliance

Please execute the following commands on the console of the old appliance

```

# define the name of the postgres container
PGCONTAINER=gl_postgres

# create database user for the migration
docker exec -u postgres $PGCONTAINER psql postgres -c "CREATE ROLE migrator WITH SUPERUSER LOGIN REPLICATION
PASSWORD 'panagenda'"

# check if listening properly
netstat -tln

```

The last command should show 0.0.0.0:5432 in state LISTEN

```

[root@ifa-dev-01 ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      5542/docker-proxy
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1140/sshd
tcp        0      0 0.0.0.0:3000           0.0.0.0:*               LISTEN      4970/docker-proxy
tcp        0      0 0.0.0.0:5432           0.0.0.0:*               LISTEN      4479/docker-proxy
tcp        0      0 0.0.0.0:443            0.0.0.0:*               LISTEN      5524/docker-proxy
tcp6       0      0 :::80                  :::*                   LISTEN      5548/docker-proxy

```

Now, lets stop all containers (except the database) on the old appliance

```

# Stop all containers, except postgres
docker stop gl_nginx gl_tomcat gl_data_miner gl_metabase gl_cron -t 120

```

## Config Migration

**On Old appliance:** create the backup

```

# go to volumes folder
cd /opt/panagenda/appdata/volumes/

# pack an archive with everything and store it in /root/
tar -czvf /root/appdata.tar.gz gl_nginx ./notes/names.nsf ./notes/user.id*

```

**On New appliance:** restore the backup

```

# go to volumes folder
cd /opt/panagenda/appdata/volumes/

# rename default names.nsf
mv ./notes/names.nsf ./notes/names.orig_nsf

# copy the file over (replace OLDHOST with hostname/ip of the old appliance)
scp OLDHOST:/root/appdata.tar.gz /root/

# extract files (will be placed in correct folders automatically)
tar -xvzf /root/appdata.tar.gz

# clean-up (optional)
rm -f /root/appdata.tar.gz

```

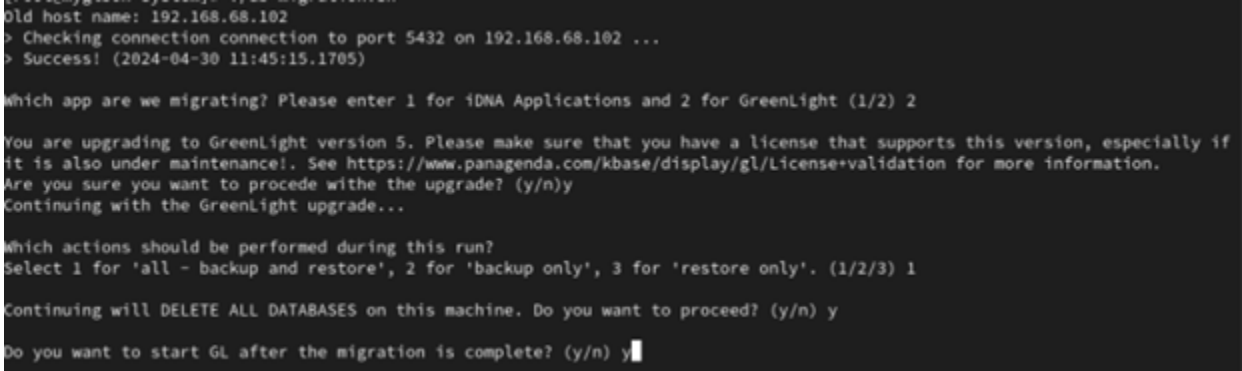
## Database Migration

## On New appliance

Execute the statements in this chapter from a command line via VNC if possible! (SSH sessions - putty - might be interrupted)

```
# execute the DB Migration Script
/opt/panagenda/appdata/system/db-migration.sh
```

- Select "2" - for GreenLight
- Enter "y" to confirm that you have received already a new license file from panagenda
- Select "1" - to perform a full backup and restore
- Enter "y" to delete the default (empty) database on the new appliance
- Enter "y" to start Greenlight after the migration is complete



```
Old host name: 192.168.68.102
> Checking connection connection to port 5432 on 192.168.68.102 ...
> Success! (2024-04-30 11:45:15.1705)

Which app are we migrating? Please enter 1 for iDNA Applications and 2 for GreenLight (1/2) 2

You are upgrading to GreenLight version 5. Please make sure that you have a license that supports this version, especially if
it is also under maintenance!. See https://www.panagenda.com/kbase/display/gl/License+validation for more information.
Are you sure you want to procede with the upgrade? (y/n)y
Continuing with the GreenLight upgrade...

Which actions should be performed during this run?
Select 1 for 'all - backup and restore', 2 for 'backup only', 3 for 'restore only'. (1/2/3) 1

Continuing will DELETE ALL DATABASES on this machine. Do you want to proceed? (y/n) y

Do you want to start GL after the migration is complete? (y/n) y
```

**Whenever done, the new Appliance is up and running and you can make use of it.**

The Old Appliance can be switched off

## Start GL on new appliance

In case you have chosen that GreenLight should not start after the migration is complete (last item in the previous chapter) you need to manually start it.

```
# restart all GL containers:
gl start
```