Running iDNA on your own Linux

- Overview
- Responsibilities
- Operating Systems
  - Recommended Distributions
  - Supported Distributions
  - Unsupported Setups
  - Not Supported
  - Evaluation on Windows (WSL2)
- Host Requirements
  - Access and Software
  - Network
  - Domino Name Resolution / DNS
  - Dedicated Host
- Sizing
- Partitioning
- Next Steps

Overview

Starting with version 3.5.0, iDNA Applications runs on a Linux host that you provide. You prepare the host with a supported Linux distribution and Docker; the panagenda installer then sets up iDNA Applications on it as a set of Docker containers. This gives you more flexibility in choosing your virtualization platform and makes it easier to keep the operating system iDNA runs on up to date.

As of version 3.5.0, panagenda no longer supplies virtual appliance images for VMware or Hyper-V. Existing iDNA appliances keep running and are upgraded with the same installer (see Upgrading iDNA Applications).

This page lists the requirements for the Linux host. Share it with the team that provides your Linux servers, and have them prepare the host before you install iDNA.

The installation at a glance:

1. Provide a Linux host that meets the requirements on this page.
2. Create and mount the filesystems for iDNA (see Partitioning).
3. Install Docker Engine.
4. Download and run the iDNA installer (see Installing iDNA Applications on Linux).
5. Log in to iDNA and configure it.

Responsibilities

- **You** provide and operate the host: the operating system and its security updates, Docker, monitoring, and backups of the host and of the iDNA database.
- **panagenda** provides iDNA Applications as Docker containers, and the installer that installs and upgrades them. Each new iDNA version also brings updated components inside the containers.

Operating Systems

Recommended Distributions

panagenda recommends these distributions. They are among the most widely used Linux distributions, and panagenda tests iDNA Applications on them:

- **Red Hat Enterprise Linux 9 and 10**
- **AlmaLinux 9 and 10**
- **Ubuntu 24.04 LTS and 26.04 LTS**

panagenda tests on Red Hat Enterprise Linux 9, AlmaLinux 10, Ubuntu 24.04 LTS and Ubuntu 26.04 LTS. AlmaLinux is binary-compatible with Red Hat Enterprise Linux, so each of these tests also covers the same version of the other distribution.

Note on version RHEL 10: Red Hat Enterprise Linux 10 and most compatible distributions require a processor with the x86-64-v3 feature level (Intel Haswell from 2013 or newer, or a comparable AMD processor). On a virtual machine, the hypervisor must pass these processor features through; an older VMware EVC mode, for example, keeps the operating system from starting. AlmaLinux 10 also offers a build for older processors.

Supported Distributions

iDNA Applications runs on any Linux distribution that meets these requirements:

1. An x86_64 processor (64-bit Intel or AMD).
2. A current Docker Engine with the Docker Compose plugin (version 2).
3. Docker starts at boot. systemd is recommended: the setting that makes Docker wait for the iDNA filesystems is a systemd setting (see Partitioning).
4. bash and the standard tools the installer checks for (see Access and Software).

5. Writable directories `/opt` and `/usr/bin`.
6. Ports 80 and 443 free.

panagenda support (support@panagenda.com) helps you install and run iDNA Applications on any distribution that meets these requirements, including distributions panagenda does not test itself. Examples are Rocky Linux, Oracle Linux, CentOS Stream, Debian and SUSE Linux Enterprise.

Unsupported Setups

Setups that do not meet these requirements but may still work. You are welcome to try them, but panagenda does not recommend them for production and cannot guarantee that they run stably or keep working after an upgrade. panagenda support covers iDNA Applications itself, starting at the web interface, but not the setup.

Not Supported

- ARM processors (for example AWS Graviton or Apple Silicon): the iDNA containers exist for x86_64 only.
- Distributions with a read-only system partition, such as Fedora CoreOS, Flatcar or Ubuntu Core: the installer needs to write to `/usr/bin`.
- Podman instead of Docker Engine: the installer and the iDNA containers need Docker Engine (`docker compose`, `docker info` and the Docker socket).
- Windows and macOS as hosts: containers need a Linux kernel. For WSL2, see Evaluation on Windows (WSL2).

Evaluation on Windows (WSL2)

For evaluation and tests, iDNA Applications can run in the Windows Subsystem for Linux (WSL2), with a recommended distribution and Docker. It is not intended for production: the WSL2 environment depends on a Windows user session, and ports 80 and 443 have to be forwarded from Windows.

Host Requirements

Access and Software

- **Root access:** the installer runs as root, directly or through `sudo`.
- **Docker Engine with the Docker Compose plugin** (Compose version 2). Install it from Docker's own repository, following Docker's guide for your distribution:
  - Red Hat Enterprise Linux: docs.docker.com/engine/install/rhel
  - CentOS Stream: docs.docker.com/engine/install/centos
  - Ubuntu: docs.docker.com/engine/install/ubuntu

  Docker does not list AlmaLinux, Rocky Linux and Oracle Linux separately; use the instructions for Red Hat Enterprise Linux or CentOS. Docker's guides first remove Podman and runc, which these distributions may have installed. On Ubuntu, the Docker snap package is not supported. Afterwards, both of these commands must work:

  ```
  docker version
  docker compose version
  ```

  Docker must start at boot; the iDNA containers start automatically together with Docker:

  ```
  systemctl enable --now docker
  ```

- **Standard tools:** tar, grep, awk, sed, hostname, getent, timeout, ip, fdisk, one of ss, lsof, netstat or fuser, and timedatectl. A normal server installation includes them; the installer names any that are missing. You also need unzip to unpack the download. With curl, the installer also checks at the end that the server address answers; without it, it skips this check.
- **SELinux** can stay in enforcing mode. With a standard Docker installation, no SELinux changes are needed. Only if SELinux support is enabled in Docker (`"selinux-enabled": true` in `/etc/docker/daemon.json`) does the installer need the package policycoreutils-python-utils, and it asks before it changes anything. On Ubuntu, AppArmor needs no changes.

Network

- **Ports 80 and 443** must be free on the host. The iDNA web server uses both; port 80 redirects to HTTPS on port 443. Users and administrators reach iDNA over HTTPS.
- **Firewall:** Docker opens the ports of its containers with its own firewall rules, so firewalld or ufw on the host usually need no extra rule. Firewalls between your users and the host must allow port 443.
- **Domino servers:** the host needs the same network access to your Domino servers as before, for example Notes RPC on port 1352 (see System Requirements, Access and Permissions). The host must also resolve the names of your Domino servers (see Domino Name Resolution / DNS).
- **Internal Docker network:** iDNA uses a private /24 subnet for the communication between its containers, by default 192.168.238.0/24. If your network already uses this subnet, change it after the installation (see Configuration File).
- **Internet access:** the online installer downloads the iDNA Docker images from docker.panagenda.com over HTTPS (no login needed). If the host reaches the internet through a proxy, configure Docker's proxy settings (docs.docker.com/engine/daemon/proxy). The offline installer contains the images and needs no internet access.
- **Server address:** decide on the DNS name under which users will open iDNA in their browser, for example `idna-apps.acme.com`, and register it in DNS so that it points to the host before the installation. iDNA uses this address for its web interface and its Metabase reports, in the links of Metabase emails, and in iDNA alert emails. The installer suggests names, checks them in DNS and lets you choose (see Server Address). The host's own name does not have to match, and the installer does not rename the host. An IP address also works, but every link breaks when it

changes. Do not use `idna` as the server address or as its first part (for example `idna.acme.com`): iDNA uses the server address as the host name of one of its containers, which can then no longer reach the container called `idna`. Avoid `idna` as the host's own name too, because the installer suggests server addresses based on it.
- **Time zone:** set the time zone of the host (`timedatectl set-timezone`). The installer copies it into the iDNA configuration.

Domino Name Resolution / DNS

iDNA connects to your Domino servers most reliably when the host can resolve the common name of each Domino server in DNS, the same way Notes clients usually find their servers. The common name is the first part of the server's hierarchical name, for example `MailServer01` for `MailServer01/ACME`. Without this, iDNA works with the network names of the servers, but you have to know them and keep them up to date in iDNA, which often delays the setup. When a connection fails, iDNA's data collection also tries the common name, which avoids many collection problems.

The network name of a server is often different from its common name, for example `msrveu01.internal.acme.com`. A DNS alias (CNAME record) connects the two:

```
mailserver01.internal.acme.com  CNAME  msrveu01.internal.acme.com
```

For the short name to resolve, the DNS search list of the host must contain the domain of these aliases, here `internal.acme.com`. Hosts with a static network configuration often have no search domain; use the one your Notes clients get.

Check it on the host for each Domino server:

```
getent hosts MailServer01
```

The output must show the IP address of the Domino server. `ping MailServer01` shows the address too, in its first line; whether ping gets a reply does not matter.

Use DNS for this, not `/etc/hosts`. iDNA runs in Docker containers, which use the DNS servers and search domains of the host. Whether they also see the entries in the host's `/etc/hosts` depends on how the host resolves names: with the default network setup of Ubuntu (systemd-resolved) they do, with that of RHEL and AlmaLinux they do not. Entries in `/etc/hosts` also have to be maintained by hand on this one host.

If the common names cannot be resolved in your network, enter the network name or IP address of each server in iDNA instead: `https://<server address>/idna/sys/servers`, column [Hostname].

Dedicated Host

We recommend a host that runs only iDNA Applications. To manage the iDNA containers, the iDNA cron container has access to the Docker daemon (`/var/run/docker.sock`). This gives it full control over every container on the host, including containers of other applications. The installer points this out and asks for confirmation if other containers are already running on the host when iDNA is installed for the first time.

Sizing

For most environments, start with:

- 4 CPU cores
- 16 GB RAM
- 300 GB disk space

RAM and disk space grow with the number of users, the number of databases and the collection period. For larger environments, see the sizing table in System Requirements, Virtual Hardware; it applies to your own Linux host as well.

Partitioning

As on the former iDNA appliance, keep the iDNA directories that grow on filesystems of their own. If the log files or the operating system fill their filesystem, the database keeps working, and the filesystems that grow can be extended without touching the root filesystem.

Example for the 300 GB default:

Filesystem	Recommendation	Size in the example	Why
`/` (root)	required	70 GB	The operating system (about 20 GB) and Docker, which keeps the iDNA images and containers in `/var/lib/docker`. During an upgrade, the old and the new images exist side by side, and the offline installer unpacks about 8 GB in `/tmp`. Plan the root filesystem generously: a full root filesystem is hard to fix, while the iDNA filesystems are easy to extend.
`/opt/panagenda/appdata`	recommended	30 GB	The Notes database designs iDNA downloads, stored as zipped DXL files. Fills during the first design collection and grows slowly afterwards: about 5 MB per database, 30 to 80 GB in large environments.

`/opt/panagenda/logs`	recommended	5 GB	The log files of all iDNA services. iDNA deletes them after 7 to 14 days; a filesystem of their own keeps an unusual amount of log output away from the operating system and the database.
`/opt/panagenda/pgdata`	strongly recommended	175 GB	The iDNA database, the part that grows continuously. A filesystem of its own keeps a full root or log filesystem from affecting it.
unassigned	recommended	20 GB	Free space in the LVM volume group, to extend a filesystem quickly when needed.

Rules for the filesystems:

- **Use LVM,** so that the filesystems can be extended.
- **Mount the filesystems before you run the installer,** with entries in `/etc/fstab`. Otherwise the installer writes into the directories on the root filesystem, and a filesystem mounted there later hides that data.
- **Make Docker wait for the filesystems at boot.** If `/opt/panagenda/pgdata` is not mounted when Docker starts, the database container finds an empty directory and creates a new, empty database on the root filesystem; iDNA then starts without its data. This setting keeps Docker from starting until the filesystems are mounted (list the filesystems you created):

```
mkdir -p /etc/systemd/system/docker.service.d
cat > /etc/systemd/system/docker.service.d/panagenda-mounts.conf <<'EOF'
[Unit]
RequiresMountsFor=/opt/panagenda/pgdata /opt/panagenda/appdata /opt/panagenda/logs
EOF
systemctl daemon-reload
```

Next Steps

- New installation: Installing iDNA Applications on Linux
- Upgrading an existing installation or appliance: Upgrading iDNA Applications