Instalação e Configurações do Authentik

Instalação Authentik em docker compose

Link: https://docs.goauthentik.io/install-config/install/docker-compose/

em 10/04/2026

Docker Compose installation

This installation method is for test setups and small-scale production setups.

Requirements

- A host with at least 2 CPU cores and 2 GB of RAM
- Podman or Docker Compose (Compose v2, see instructions for upgrade)

Video

View our video about installing authentik on Docker.

https://www.youtube.com/embed/O1qUbrk4Yc8?si=HiSBjmJYhE_oJhB1start=22

Download the Compose file

To download the latest `compose.yml` open your terminal, navigate to the directory of your choice, and then run the following command:

- Linux
- macOS

```
wget https://docs.goauthentik.io/compose.yml
```

Generate PostgreSQL password and secret key

If this is a fresh authentik installation, you need to generate a PostgreSQL password and a secret key. Use a secure password generator of your choice such as `pwgen`, or you can use `openssl` as below.

Run the following commands to generate a PostgreSQL password and secret key and write them to your `.env` file:

```
echo "PG_PASS=$(openssl rand -base64 36 | tr -d '\n')" >> .env
echo "AUTHENTIK_SECRET_KEY=$(openssl rand -base64 60 | tr -d '\n')" >> .env
```

INFO

Because of a PostgreSQL limitation, only passwords up to 99 chars are supported. See: https://www.postgresql.org/message-id/09512C4F-8CB9-4021-B455-EF4C4F0D55A0@amazon.com

To enable error reporting, run the following command:

```
echo "AUTHENTIK_ERROR_REPORTING__ENABLED=true" >> .env
```

For an explanation about what each service in the Docker Compose file does, see Architecture.

Configure custom ports

By default, authentik listens internally on port 9000 for HTTP and 9443 for HTTPS. To use different exposed ports such as 80 and 443, you can set the following variables in `.env`:

```
COMPOSE_PORT_HTTP=80
COMPOSE_PORT_HTTPS=443
```

See Configuration to change the internal ports. Be sure to run `docker compose up -d` to rebuild with the new port numbers.

Docker socket

By default, the authentik Docker Compose file mounts the Docker socket to the authentik worker container:

```
- /var/run/docker.sock:/var/run/docker.sock
```

This is used for automatic deployment and management of authentik Outposts.

Mounting the Docker socket to a container comes with some inherent security risks. To reduce these risks, you can utilize a Docker Socket Proxy as an additional layer of protection.

Alternatively, you can remove this mount and instead manually deploy and manage outposts.

Email configuration (optional but recommended)

It is also recommended to configure global email settings. These are used by authentik to notify administrators about alerts, configuration issues and new releases. They can also be used by Email stages to send verification/recovery emails.

For more information, refer to our Email configuration documentation.

Install and start authentik

WARNING

All internal operations use UTC. Times displayed in the UI are automatically localized for the user. Do not update or mount `/etc/timezone` or `/etc/localtime` in the authentik containers; it will cause problems with OAuth and SAML authentication, as seen this GitHub issue.

After you have downloaded the `docker-compose.yml` file, generated a password and a secret key, and optionally configured your global email, run these commands to retrieve and install the current version of authentik:

```
docker compose pull
docker compose up -d
```

The `compose.yml` file statically references the latest version available at the time of downloading the compose file. Each time you upgrade to a newer version of authentik, you download a new `compose.yml` file, which points to the latest available version. For more information, refer to the **Upgrading** section in the Release Notes.

Access authentik

To start the initial setup, navigate to `http://<your server's IP or hostname>:9000/if/flow/initial-setup/`.

Initial setup in browser

You will get a `Not Found` error if initial setup URL doesn't include the trailing forward slash `/`. Also verify that the authentik server, worker, and PostgreSQL database are running and healthy. Review additional tips in our troubleshooting docs.

There you are prompted to set a password for the `akadmin` user (the default user).

First steps in authentik

You are now ready to add your first application and its provider. Then you'll want to add a new user.

To view a typical workflow for adding applications and users, with helpful context and explanations for each step, refer to the First Steps tutorial.

?? First steps

Add an application and provider, then create a user.

Help us improve this content

We welcome your knowledge and expertise. If you see areas of the documentation that you can improve (fix a typo, correct a technical detail, add additional context, etc.) we would really appreciate your contribution.

Installation and Configuration

Kubernetes installation

Integração do Nextcloud com Authentik OIDC

Link: https://integrations.goauthentik.io/chat-communication-collaboration/nextcloud/

Em 09/04/2026

What is Nextcloud

> Nextcloud is a suite of client-server software for creating and using file hosting services. Nextcloud is free and open-source, which means that anyone is allowed to install and operate it on their own private server devices.
>
> -- https://nextcloud.com/arning

WARNING

If you require server side encryption, you must use LDAP. OpenID and SAML will cause **irrevocable data loss**. Nextcloud server side encryption requires access to the user's cleartext password, which Nextcloud has access to only when using LDAP because the user enters their password directly into Nextcloud.aution

This setup only works when Nextcloud is running with HTTPS enabled. See here on how to configure this.nfo

If there's an issue with the configuration, you can log in using the built-in authentication by

visiting http://nextcloud.company/login?direct=1.

Configuration methods

It is possible to configure Nextcloud to use OIDC, SAML, or LDAP for authentication. Below are the steps to configure each method.

- OIDC
- SAML

- LDAP

OIDC

Preparation

The following placeholders are used in this guide:

- `nextcloud.company` is the FQDN of the Nextcloud installation.
- `authentik.company` is the FQDN of the authentik installation.

Info

This documentation lists only the settings that you need to change from their default values. Be aware that any changes other than those explicitly mentioned in this guide could cause issues accessing your application.

WARNING

If you require server side encryption, you must use LDAP. OpenID and SAML will cause **irrevocable data loss**.

Let's start by considering which user attributes need to be available in Nextcloud:

- name
- email
- unique user ID
- storage quota (optional)
- groups (optional)

authentik already provides some default *scopes* with *claims*, such as:

- `email` scope: includes `email` and `email_verified`
- `profile` scope: includes `name`, `given_name`, `preferred_username`, `nickname`, `groups`
- `openid` scope: a default required by the OpenID spec (contains no claims)

Create property mapping *(optional)*

If you do not need storage quota, group information, or to manage already existing users in Nextcloud, skip to the next section.
If you want to control user storage and designate Nextcloud administrators, you will need to create a property mapping.

1. Log in to authentik as an administrator and open the authentik Admin interface.
2. Navigate to **Customization** > **Property mappings** and click **Create**.
   - **Select type**: select **Scope mapping**.
   - **Create Scope Mapping**:
     - **Name**: `Nextcloud Profile`
     - **Scope name**: `nextcloud`
     - **Expression**:

```
# Extract all groups the user is a member of
groups = [group.name for group in user.groups.all()]

# In Nextcloud, administrators must be members of a fixed group called "admin".

# If a user is an admin in authentik, ensure that "admin" is appended to their group list.
if user.is_superuser and "admin" not in groups:
    groups.append("admin")

return {
    "name": request.user.name,
    "groups": groups,
     # Set a quota by using the "nextcloud_quota" property in the user's attributes
    "quota": user.group_attributes().get("nextcloud_quota", None),
     # To connect an existing Nextcloud user, set "nextcloud_user_id" to the Nextcloud username.
    "user_id": user.attributes.get("nextcloud_user_id", str(user.uuid)),
}
```

3. Click **Finish**.

Info

To set a quota, define the `nextcloud_quota` attribute for individual users or groups. For example, setting it to `1 GB` will restrict the user to 1GB of storage. If not set, storage is unlimited.

To connect to an existing Nextcloud user, set the `nextcloud_user_id` attribute to match the Nextcloud username (found under the user's `Display name` in Nextcloud).

Create an application and provider in authentik

1. Log in to authentik as an administrator and open the authentik Admin interface.
2. Navigate to **Applications** > **Applications** and click **Create with Provider** to create an application and provider pair. (Alternatively you can first create a provider separately, then create the application and connect it with the provider.)
   - **Application**: provide a descriptive name, an optional group for the type of application, the policy engine mode, and optional UI settings.
   - **Choose a Provider type**: select **OAuth2/OpenID Connect** as the provider type.
   - **Configure the Provider**: provide a name (or accept the auto-provided name), the authorization flow to use for this provider, and the following required configurations.
     - Note the **Client ID** and **slug** values because they will be required later.
     - Set a `Strict` redirect URI to `https://nextcloud.company/apps/user_oidc/code`.
     - Select any available signing key.
     - Under **Advanced protocol settings**:
       - *(optional)* If you created the `Nextcloud Profile` scope mapping, add it to **Selected Scopes**.
       - **Subject Mode**: `Based on the User's UUID`
   - **Configure Bindings** *(optional)*: you can create a binding (policy, group, or user) to manage the listing and access to applications on a user's **My applications** page.
3. Click **Submit** to save the new application and provider.

Info

Depending on your Nextcloud configuration, you may need to use `https://nextcloud.company/index.php/` instead of `https://nextcloud.company/`.

Nextcloud configuration

1. In Nextcloud, ensure that the **OpenID Connect user backend** app is installed.
2. Log in to Nextcloud as an administrator and navigate to **Settings** > **OpenID Connect**.
3. Click the **+** button and enter the following settings:
   - **Identifier**: `authentik`
   - **Client ID**: Client ID from authentik
   - **Client secret**: Client secret from authentik
   - **Discovery endpoint**: `https://authentik.company/application/o/<application_slug>/.well-known/openid-configuration`
   - **Scope**: `email profile nextcloud openid`
   - Under **Attribute mappings**:
     - **User ID mapping**: `sub` (or `user_id` for existing users)
     - **Display name mapping**: `name`
     - **Email mapping**: `email`

- **Quota mapping**: `quota` (leave blank if the `Nextcloud Profile` property mapping was skipped)
- **Groups mapping**: `groups` (leave blank if the `Nextcloud Profile` property mapping was skipped)
  Tip: Enable **Use group provisioning** to allow writing to this field.

- **Use unique user ID**: If this option is disabled, Nextcloud will use the mapped user ID as the Federated Cloud ID.

Info

If authentik and Nextcloud are running on the same host, you will need to add `'allow_local_remote_servers' => true` to your nextcloud `config.php` file. This setting allows remote servers with local addresses.

Info

To avoid a hashed Federated Cloud ID, deselect **Use unique user ID** and use `user_id` for the User ID mapping.

Danger

If you're using the `Nextcloud Profile` property mapping and want administrators to retain their ability to log in, make sure that **Use unique user ID** is disabled. If this setting is enabled, it will remove administrator users from the internal admin group and replace them with a hashed group ID named "admin," which does not have real administrative privileges.

Enabling OIDC back-channel logout

To automatically log users out of their Nextcloud sessions when they log out of authentik, enable back-channel logout.

1. Log in to Nextcloud as an administrator and navigate to **Settings** > **OpenID Connect**.
2. Under **Registered Providers**, locate the provider with the identifier used earlier.
3. Copy the `back-channel-logout-URL` value for that provider.
   For example: `https://nextcloud.company/apps/user_oidc/backchannel-logout/<identifier>`
4. In authentik, navigate to **Applications** > **Providers** and edit the Nextcloud provider.
5. Under **Protocol Settings**, set the **Logout URI** to the copied back-channel logout URL.
6. Set the **Logout Method** to `Back-channel`.

Making OIDC the default login method

Automatically redirect users to authentik when they access Nextcloud by running the following command on your Nextcloud docker host:

Opção 1 - (Tela de Login Authentik direto)
sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ config:app:set **--value=0** user_oidc allow_multiple_user_backends.

Opção 2 - (Tela de Login Authentik e Nextcloud)
sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ config:app:set **--value=1** user_oidc allow_multiple_user_backends.

Configuration verification

To confirm that authentik is correctly configured with Nextcloud, log out and then log back in by clicking **OpenID Connect**. You'll then be redirected to authentik to log in, and once authentication is successful, you'll reach the Nextcloud dashboard.

Resources

- Nextcloud docs - User authentication with LDAP
- Nextcloud OIDC App - User Documentation

Help us improve this content

We welcome your knowledge and expertise. If you see areas of the documentation that you can improve (fix a typo, correct a technical detail, add additional context, etc.) we would really appreciate your contribution.

- Edit on GitHub
- Contributor Guide
- Open an issue
- Get Enterprise Support

Integração do Nextcloud com Authentik LDAP

Link: https://integrations.goauthentik.io/chat-communication-collaboration/nextcloud/

Em 09/04/2026

What is Nextcloud

> Nextcloud is a suite of client-server software for creating and using file hosting services. Nextcloud is free and open-source, which means that anyone is allowed to install and operate it on their own private server devices.
>
> -- https://nextcloud.com/

WARNING

If you require server side encryption, you must use LDAP. OpenID and SAML will cause **irrevocable data loss**. Nextcloud server side encryption requires access to the user's cleartext password, which Nextcloud has access to only when using LDAP because the user enters their password directly into Nextcloud.

CAUTION

This setup only works when Nextcloud is running with HTTPS enabled. See here on how to configure this.

INFO

If there’s an issue with the configuration, you can log in using the built-in authentication by visiting http://nextcloud.company/login?direct=1.

Configuration methods

It is possible to configure Nextcloud to use OIDC, SAML, or LDAP for authentication. Below are the steps to configure each method.

- OIDC

- SAML
- LDAP

LDAP Configuration

Preparation

The following placeholders are used in this guide:

- `nextcloud.company` is the FQDN of the Nextcloud installation.
- `authentik.company` is the FQDN of the authentik installation.

INFO

This documentation lists only the settings that you need to change from their default values. Be aware that any changes other than those explicitly mentioned in this guide could cause issues accessing your application.

Create an application and provider in authentik

1. Log in to authentik as an administrator and open the authentik Admin interface.
2. Navigate to **Applications** > **Applications** and click **Create with Provider** to create an application and provider pair. (Alternatively you can first create a provider separately, then create the application and connect it with the provider.)
   - **Application**: provide a descriptive name, an optional group for the type of application, the policy engine mode, and optional UI settings.
   - **Choose a Provider type**: select **LDAP** as the provider type.
   - **Configure the Provider**: provide a name (or accept the auto-provided name) and the bind flow to use for this provider
   - **Configure Bindings** *(optional)*: you can create a binding (policy, group, or user) to manage the listing and access to applications on a user's **My applications** page.
3. Click **Submit** to save the new application and provider.

Create an LDAP outpost

1. Log in to authentik as an admin, and open the authentik Admin interface.
2. Navigate to **Applications** > **Outposts** and click **Create**.
   - **Name**: provide a suitable name for the outpost.
   - **Type**: `LDAP`
   - Under applications, add the newly created Nextcloud application to **Selected Applications**.

3. Click **Create**.

Nextcloud configuration

1. In Nextcloud, ensure that the **LDAP user and group backend** app is installed.
2. Log in to Nextcloud as an administrator.
3. Navigate to **Settings** > **LDAP user and group backend** and configure the following settings:
   - On the **Server** tab:
     - Click the **+** icon and enter the following settings:
       - **Host**: enter the hostname/IP address of the authentik LDAP outpost preceded by `ldap://` or `ldaps://`. If using LDAPS you will also need to specify the certificate that is being used.
       - **Port**: `389` or `636` for secure LDAP.
       - Under **Credentials**, enter the **Bind DN** of the authentik LDAP provider and the associated user password.
       - Under **Base DN**, enter the **Search base** of the authentik LDAP provider.
   - On the **Users** tab:
     - Set **Only these object classes** to `Users`.
   - On the **LDAP/AD integration** tab:
     - Uncheck **LDAP/AD Username**.
     - Set **Other Attributes** to `cn`.
     - Click **Expert** in the top right corner and enter these settings:
       - **Internal Username Attribute**: `uid`
       - **UUID Attribute for Users**: `uid`
       - **UUID Attribute for Groups**: `gidNumber`
     - Click **Advanced** in the top right corner and enter these settings:
       - Under **Connection Settings**:
         - **Configuration Active**: checked
       - Under **Directory Settings**:
         - **User Display Name Field**: `name`
         - **Base User Tree**: enter the **Search base** of the authentik LDAP provider.
         - **Group Display Name Field**: `cn`
         - **Base Group Tree**: enter the **Search base** of the authentik LDAP provider.
         - **Group-Member Association**: `gidNumber`
       - Under **Special Attributes**:
         - **Email Field**: `mailPrimaryAddress`
   - On the **Groups** tab:
     - Set **Only these object classes** to `groups`.

- Select the authentik groups that require Nextcloud access.

INFO

If Nextcloud is behind a reverse proxy, force HTTPS by adding `'overwriteprotocol' => 'https'` to the Nextcloud `config/config.php` file. See the Nextcloud admin manual for more details.

Configuration verification

To confirm that authentik is properly configured with Nextcloud, log out and log back in using LDAP credentials. If successful you will then be redirected to the Nextcloud dashboard.

Resources

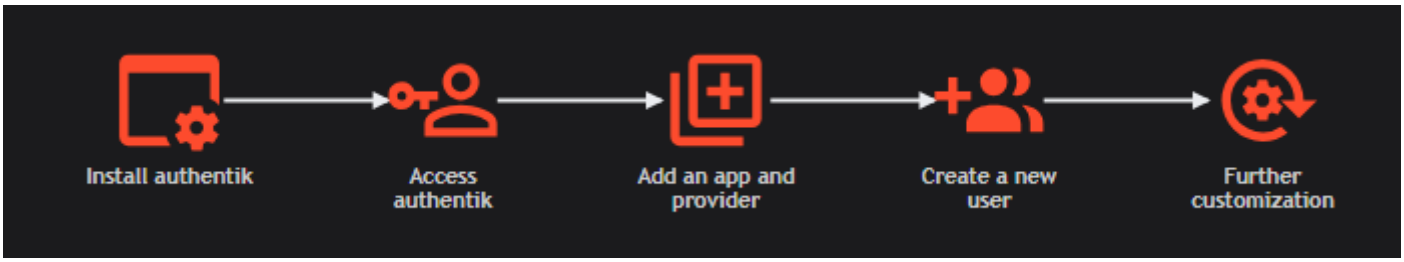
- Nextcloud docs - User authentication with LDAP
- Nextcloud OIDC App - User Documentation

Configurações iniciais do Authentik

Link: https://docs.goauthentik.io/install-config/first-steps/

First steps

After you have installed and started authentik, you are now ready to add your first application and provider, add some users, and get started with using authentik as your Identity provider.



Where are we now, and what's next?

The following tutorial assumes that you have already:

1. Installed authentik on either Docker Compose, Kubernetes, or AWS CloudFormation and confirmed that the server, worker, and the PostgreSQL database are started and running.
2. Opened authentik in your browser to the `initial-setup` flow and added credentials for a default Admin account. (Docker, Kubernetes), or AWS CloudFormation.

Initial setup in browser
You will get a `Not Found` error if the initial setup URL doesn't include the forward slash `/` at the very end of the URL. Also verify that the authentik server, worker, and PostgreSQL database are running and healthy. Review additional tips in our troubleshooting docs.

Other optional pre-installation configurations that you might have already completed include:

- Configured your global email address.
- Configured your PostgreSQL settings (read-replica, connections, etc.).
- Configured a reverse proxy.
- Configured your media storage settings or optionally AWS S3 file storage.

- Added additional custom configurations environment variables.
- Verified your configuration settings.

Install your first application and provider

Now that you have your authentik instance installed and configured with the required settings, you can add your first application and provider. After that, we'll walk through how to add your first user.

Security Best Practice

In a production environment, best practice is to first create a group, then create the user(s), and then add the application. Then you can configure the application to have a binding to a specific group or user. The binding controls the access to the application (whether or not it is displayed on a user's My Applications page).

authentik supports integration with any application; refer to our Integrations documentation to view integrations guides for over 180 of the most common ones.

In this guide we'll be setting up Grafana as an example application.

Why Grafana?

1. Log in to authentik as an administrator and open the authentik Admin interface.

A. In the Admin interface, navigate to **Applications** > **Applications** and click **Create with Provider** to create an application and provider pair.

About application and provider pairs

Every application that you add to authentik requires a provider, which is used to configure the specific protocol between the application and authentik, for example OAuth2/OIDC, SAML, LDAP, or others.

B. Provide the details for the application (Grafana) and provider (OAuth2/OIDC).

- **Configure the Application**:
  - **Name**: provide a descriptive name (such as Grafana).
    - **Group**: select an optional group for the application; groups are used to visually separate applications. For example, you can choose to group applications that you use for coding from those you use for internal communication.
  - **Policy engine mode**: select **Any** for this tutorial; the mode determnes how strictly policies are adhered to.

    - **TIP**: in authentik, policies are used in authentik to fine-tune access to applications, flows, stages and many other authentik components. It is not required to use a policy at all, though. The *policy engine mode* setting of **Any** means that as long as a single policy passes (or if there are no policies bound to the application), then access to the application is granted. The mode **ALL** means that every one of any policies bound to the application must pass in order for a user to have access to the application.
  - **UI Settings**: optional UI settings that are displayed about the application, including the launch URL, and three settings to display extra information about the application on the **My Applications** page: an optional icon, the publisher of the application, and a brief description.
- **Choose a Provider type**: select **OAuth2/OpenID Connect** as the provider type.
- **Configure the Provider**:
  - **Name**: Provide a name (or accept the auto-provided name).
  - **Authorization flow**: Select the default `implicit` authorization flow to use for this provider.
    - **TIP**: The authorization flow is where the various steps, or *stages* of authorization are defined and executed. The defined set of stages construct the workflows of authentication, authorization, etc.
  - **Protocol settings** provide the following required configurations:
    - Note the **Client ID**, **Client Secret**, and **Slug** values because they will be required later when you configure Grafana to use authentik.
    - Set a `Strict` redirect URI to `https://grafana.company/login/generic_oauth`.
      - **TIP**: The Redirect URI is where the application will go as soon as authentik's authorization flow is successfully completed.
  - **Logout URI**: set to `https://grafana.company/logout`.
  - **Logout Method**: set to `Front-channel`.
    - **TIP**: With OAuth2, front-channel logout is considered the default because most application (including Grafana) do not support back-channel logout.
  - **Signing key**: select any available signing key.
    - **TIP**: authentik generates a key that you can use, called the `authentik Self-signed Certificate`, if you do not have a specific signing key for an application.
- **Configure Bindings** *(optional)*: for this tutorial, skip this step because you do not yet have a user. Later, after you create your first user, you can create a binding to manage the display and access to applications on a user's **My applications** page.
  - **TIP**: By creating a binding between an application and a specific user, you are ensuring that the application is accessible only to that user and any other users or groups for whom you created a binding. Learn more about how bindings are used in authentik in our Bindings overview.

  For any fields not mentioned above, you can leave the default value.

C. Click **Submit** to save the new application and provider.

2. Configure Grafana to use authentik as its IdP

For some applications, you log into the application and configure settings there; with Grafana you simply edit your Grafana Docker Compose file. Here you add basic configuration settings as well as the **Client ID**, **Client Secret**, and the **Slug** values that you obtained when you configured the application and provider in authentik in Step 1. above.

A. In the Grafana Docker Compose file, set the following environment variables:

Tips

These values below are for a Grafana instance running in Docker; for standalone or Helm Chart instances refer to our Grafana integration guide.

Note that `authentik.company` is a placeholder that we use in our example settings; replace this with the domain that authentik is running on in your environment.

```
environment:
    GF_AUTH_GENERIC_OAUTH_ENABLED: "true"
    GF_AUTH_GENERIC_OAUTH_NAME: "authentik"
    GF_AUTH_GENERIC_OAUTH_CLIENT_ID: "<Client ID from above>"
    GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET: "<Client Secret from above>"
    GF_AUTH_GENERIC_OAUTH_SCOPES: "openid profile email"
    GF_AUTH_GENERIC_OAUTH_AUTH_URL: "https://authentik.company/application/o/authorize/"
    GF_AUTH_GENERIC_OAUTH_TOKEN_URL: "https://authentik.company/application/o/token/"
    GF_AUTH_GENERIC_OAUTH_API_URL: "https://authentik.company/application/o/userinfo/"
    GF_AUTH_SIGNOUT_REDIRECT_URL:
"https://authentik.company/application/o/<application_slug>/end-session/"
    # Optionally enable auto-login (bypasses Grafana login screen)
    GF_AUTH_OAUTH_AUTO_LOGIN: "true"
    # Optionally map user groups to Grafana roles
    GF_AUTH_GENERIC_OAUTH_ROLE_ATTRIBUTE_PATH: "contains(groups[*], 'Grafana Admins') &&
'Admin' || contains(groups[*], 'Grafana Editors') && 'Editor' || 'Viewer'"
    # Required if Grafana is running behind a reverse proxy
    GF_SERVER_ROOT_URL: "https://grafana.company"
```

B. Save your Grafana Docker Compose file, and then launch the stack and access Grafana via your browser at the configured URL.

C. To confirm that authentik is properly configured with the new application, log out of Grafana and then log back in using the **Sign in with authentik** button. You should be redirected to authentik to provide credentials.

Add your first user

Now that you can access the authentik Admin interface, and you have added an application and provider, let's add a new user.

1. Log in to authentik as an administrator and open the authentik Admin interface.

A. Navigate to **Directory > Users**, and click **New User**.

B. Fill in the ***required*** fields:

- **Username**: This value must be unique across all users.
- **TIP**: With OAuth2, front-channel logout is considered the default because most application (including Grafana) do not support back-channel logout.
- **Path**: The path where the user will be created. By default the new user is created in the `users` directory, but you can change that later by editing the user.
  - **TIP**: Paths are basically directories, that are used to organize your users (for example HR vs Sales, etc.). Paths do not ipoact access; they are purely organizational. Note that the top-level **users** directory displays all users in that directory and all sub-directories.

For information about the ***optional*** fields below, refer to our documentation on managing users.

- **Name**: The display name of the user.
- **Email**: The email address of the user. This is required for many integrations.
- **Is active**: Define the newly created user account as active.
- **Attributes**: You can leave this empty for this tutorial. This field can be used to store custom attributes for the user, in YAML or JSON format. These attributes can then be used within property mappings and policies.

C. Click **Create**.

2. Verify that the new user was created

- Look for the new user in the list on the **Directory** > **Users** page.

What's next?

Now that you have added your first application, and a new user, here are some typical next steps:

- Assign your new user to appropriate groups and roles.
- Configure federated or external sources (an existing source of user credentials and other user data).
- Set up MFA
- Define property mappings.
- Create a custom flow.

- Install an Enterprise license
- Create a policy to control access, force MFA use, etc..

Things to know and troubleshooting tips

Review the following information to learn more about the basics of setting up authentik and for troubleshooting tips.

Modifying the Docker Compose file

Especially when you are just starting out with authentik, we recommend that you use the default `docker-compose.yml` file that comes with the download, instead of trying to write the file from scratch. After you have successfully installed, configured, and accessed authentik, you can edit the file to do more advanced configurations, as documented in the Configuration section.

Reverse proxy

Typically authentik is set up with a reverse proxy in front of it. If you already have a reverse proxy that you are using to handle your incoming network traffic, you can simply use that same reverse proxy for authentik, by adding a few configuration values. For more details see the Reverse proxy guide.

The `latest` tag is deprecated

The `:latest` tag has been deprecated and will never be updated from the 2025.2 release. Instead, use a specific version tag for authentik instances' container images, such as `:2025.12`.

Using bindings to allow or restrict access to applications

Note that if you do not define any bindings, then all users have access to the application. To control access, you can create a binding. For more information about user access, refer to our documentation about authorization and hiding an application.

Upgrades

When you are ready to upgrade to the latest version, be sure to read our Upgrade documentation and refer to the Release Notes for the specific version.

Help us improve this content

We welcome your knowledge and expertise. If you see areas of the documentation that you can improve (fix a typo, correct a technical detail, add additional context, etc.) we would really appreciate your contribution.

- Edit on GitHub
- Contributor Guide
- Open an issue
-

Tradução de mensagens no Authentik

Procedimentos necessários

1) Acessar o container do authentik server e instalar o pacote gettext

docker container exec -ti authentik-server bash

apt update; apt upgrade; apt install gettext

2) Acessar o diretório com arquivo para tradução de português brasileiro:

locale/pt_BR/LC_MESSAGES/django.po

Para facilitar as traduções e poder usar o vim para edição do arquivo, fazer o mapeamento persistente no docker-compose.

```
volumes:
    - ./volumes/locale/LC_MESSAGES:/locale/pt_BR/LC_MESSAGES
```

3) Após alterar o arquivo django.po deve ser compilado no container:

python manage.py compilemessages

4) Após finalizar os processos no container, regerar as configurações authentik com os comandos:

docker compose build, e depois reiniciar com docker compose up e down.