DP-SALA/D Stack

Documentation of the Debian, Portainer, SWAG, Authelia, LLDAP, Authy/Duo Stack. Updates with new configuration and such when needed.

- Requirements
  - Minimum Requirements for the DP-SALA/D Stack
- DP Setup
  - Debian 12 Setup
  - Portainer CE Setup
- Configuration of DP
  - Debian Configuration
  - Portainer Configuration
- SALA/D Setup
  - Overview of the SALA/D Stack
  - SWAG Configuration
  - Authelia Configuration
  - LLDAP Configuration
  - Authy/Duo Configuration
  - One File Docker Compose
- Maintenance of the DP-SALA/D Stack
  - Required Maintenance for the DP-SALA/D Stack
- How do I [Insert Question Here]?
  - How do I add a new application to my stack?
  - How do I add LLDAP Authentication to [SERVICE]?
  - How do I handle user creation?

Requirements

Requirements of the DP-SALAD Stack

Minimum Requirements for the DP-SALA/D Stack

A DP-SALA/D Stack is rather lightweight in comparison to other stacks out there, like the very similar Unraid-SALA/D stack.

This is a compilation of the minimum specs for each part of the stack.

Debian 12:

- 1GHz Processor from 2010 or later
- 2 GB RAM
- 10 GB Storage

Portainer:

- 1GHz Processor from 2010 or later
- 2 GB RAM
- 20 GB Storage

SWAG:

- 1GHz Processor from 2010 or later
- RAM requirement depends on expected requests, 4 GB recommended.
- 10 GB Storage

Authelia:

- Requires that a REDIS server is setup in addition to whatever else.

Authy/Duo:

- Authy:
  - Hooks into Authelia for 2FA aaS, only requires a bit of setup.
- Duo:
  - Hooks into Authelia for 2FA aaS, however, free tier can have max 10 users, but setup is a little easier on the user/admin.

Overall Minimum:

- Most Processors within the last 5 years should work well enough.
- 4 GB RAM minimum, although 2 GB can be gotten away with.
- 30 GB Storage minimum, especially if you plan to reverse proxy some apps.
- Basic command line knowledge.

This guide assumes that you have basic computing knowledge, including how to boot your computer into an OS from a USB stick, having a domain through a registrar, and have a GMail Address for sending Email to your potential users.

If you do not have a domain, my recommendation is to go through Google Domains or Cloudflare. These are the two most reputable registrars that do not markup pricing for renewing the domain you are leasing.

I also recommend that you set up a GMail address specifically for the purpose of this stack, as Authelia will require a way to let their users password reset, as well as setup 2FA for themselves. You may also use iCloud+'s custom domain email, however, the guide will not cover this setup, and you will have to do some research on sending email through iCloud if you choose to go this route, and you will have to find the proper settings before configuring Authelia.

DP Setup

Setup of the Debian and Portainer parts of the DP-SALAD Stack

Debian 12 Setup

Visit https://debian.org/download to get the latest version of Debian, as of this writing, Debian 12.1.0. This is a netinst, meaning that it will require internet connection to download the OS files.

Load your Debian ISO onto a USB stick, either directly, or using Rufus/Balena Etcher to write the ISO to stick.

Plug it into whatever is becoming your server, boot to the USB stick.

Install Debian using the ISO. Make sure that you are connected to the internet before trying to install Debian.

1. Select Graphical Installer
2. Follow the setup process up until hostname, and put in something that you will recognize when port-forwarding or tunneling.
3. Put in the domain name that you have leased as your domain name, like: "example.com"
4. Setup a root password. (Template uses DPSALADSTACK)
5. Setup the name for the first non-admin user. (Template uses dpsalad)
6. Setup a password for the non-admin user. (Template uses DPSALADSTACKUSER)
7. Continue the setup, and select "Guided - use entire disk" when Partitioning Disk, continue and use "All files in one partition" as the partitioning scheme.
8. Allow changes to be written to disk.
9. Continue with the installer until Software Selection. See the following points for the selections.
    1. Debian desktop environment (keep selected)
    2. GNOME (deselect)
    3. xfce (select)
    4. SSH server (select)
    5. standard system utilities (keep selected)
10. Continue with the installer until the reboot.
11. Once the reboot is complete, you should have a full working Debian system!
12. If you used the template, please change the password of both the root and regular user using passwd.

Debian had you set up a root password, but when you try using sudo, it will not let you do anything. Debian by default does not let the first user have sudo access. You will have to temporarily log in as root to fix this. Use the following commands/instructions to add your user to the sudoers file.

1. su -

2. nano /etc/sudoers
3. Add the following to the file(no quotes): "username ALL=(ALL) ALL"
4. Exit nano using Ctrl+X and then type 'Y' and enter.
5. Restart your PC. (systemctl reboot)

Once your machine restarts, you should now be able to use the sudo command.

If you'd like to disable the GUI and work in CLI only (highly recommended, as GUI uses more resources), follow these instructions:

1. sudo systemctl set-default multi-user.target
2. sudo reboot

Now time to run updates, just to guarantee the system is up-to-date. Debian uses APT as it's package manager of choice. Run these commands:

1. sudo apt-get update
2. sudo apt-get upgrade

And now, your system is up-to-date!

Next, install Docker. Follow the following instructions to install Docker:

1. sudo apt-get update
2. sudo apt-get install ca-certificates curl gnupg
3. sudo install -m 755 -d /etc/apt/keyrings
4. curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
5. sudo chmod a+r /etc/apt/keyrings/docker.gpg
6. ```
   echo \
     "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
   https://download.docker.com/linux/debian \
     "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
     sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
   ```
7. sudo apt-get update
8. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

Once this is done, you are able to move on to the next section of this guide.

Warning: If you use anything other than the netinst for Debian, your Apt sources may be invalid. Should this be the case, there is an example of the sources needed in /usr/share/doc/apt/examples/sources.list. Use the following command to copy this list of default sources to restore Apt to working order.
   ```

```
sudo cp /usr/share/doc/apt/examples/sources.list /etc/apt/sources.list
```

Portainer CE Setup

Time to setup Portainer, which is installed using Docker. Follow the below instructions to install Portainer onto your machine.

1. sudo docker volume create portainer_data
2. sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
3. Visit https://[ip-address]:9443 to log into Portainer CE.
4. Change the admin username if you wish. (Don't forget it if you change it though!) [Template uses DPSALADSTACK]
5. Use a good secure password(12+ characters) for the administrator password. (Don't forget this either!)
6. Click "Getting Started" to finish setting up Portainer CE.
7. If you are going to use the one-click Compose Install, make sure to create a network using the following command:

   ```
   sudo docker network create [INSERT NETWORK NAME HERE]
   ```

   Make sure to remove the square brackets around the network name.

Configuration of DP

Configurations, recommended and required for the DP part of the stack

Debian Configuration

Sudo should have been setup after the install, however, nothing much else is required for the setup end in Debian. If you want, here is a list of recommended software to install to make the command line a little easier to learn:

1. TLDR (sudo apt-get install tldr, tldr -u)

Portainer Configuration

Portainer is ready right out of the box. There should be nothing you need to do to configure it, unless you are running a "clustered setup" involving more than one computer, which can be done for this stack, but does add a layer of complexity, and allows for scalability for the amount of applications you can offer to either yourself or your users. A "clustered setup" is on the Portainer CE wiki and is out of the scope of this document.

If you are using the VM Template, Portainer has the custom templates loaded so that if you need to install a newer version, or change something and restart, they are there.

SALA/D Setup

Instructions to set up the SALA/D Part of the stack.

Overview of the SALA/D Stack

The SALA/D stack, or SWAG, Authelia, LLDAP, Authy/Duo stack is a simple way to get a webserver/reverse proxy, an identity and access control management system, as well as 2FA; all working together to host whatever you'd like. Here is what does what:

- SWAG
  - Handles the webserver and reverse proxy.
  - All configuration is handled via .conf files and text.
  - This guide also demonstrates how to set up the Docker Mods Auto Reload, GeoIPBlock, and ...
- Authelia
  - Handles weblogin and the front end of 2FA.
  - This has one configuration file, and hooks into LLDAP.
- LLDAP
  - Handles the "identity" part of the Authentication solution.
  - Requires that a domain is setup, and has a bit of a learning curve.
  - This has a bit of configuration involved whenever you setup a new application with LDAP integrations, however, LLDAP's GitHub page usually has sample configurations on how to setup LDAP auth for an application.
- Authy/Duo
  - Pick one between Authy or Duo.
    - Authy
      - This allows for 2FA via an 8 digit code. The configuration is handled via Authelia, with the end user setting up 2FA through the Authy App or through Bit/Vaultwarden. The user can self register their 2FA in this way, through email.
    - Duo
      - This allows for 2FA via a push notification on your phone. The configuration is handled via Authelia.

SWAG Configuration

Configuring the Secure Web Application Gateway has a lot of options and potential. It is primarily used to have one gateway for all your services hosted through two ports. Assuming you've used the Docker Compose file, you should be able to view and edit all the configuration files through /home/[USER]/appdata/swag. This location now has multiple places you can go.

1. /home/[USER]/appdata/swag/nginx - This holds all of the webserver configuration files, this is where you will go when adding a new application/service.
    1. /home/[USER]/appdata/swag/nginx/proxy-confs - This is where all the reverse proxy magic happens, and will likely be the most used folder.
    2. /home/[USER]/appdata/swag/nginx/site-confs - If you want to host a static site using SWAG, this is the conf file that you will modify to make it function. Highly recommend learning how NGINX conf files work.
    3. The files in this nginx directory are all usually not touched, but used for the rest of configuration to connect Authelia to SWAG for logins.
2. /home/[USER]/appdata/swag/etc/letsencrypt - This holds all the relevant data for your SSL certificates using Let's Encrypt. The current certificates are in the "live" folder, under the domain name folder.
3. /home/[USER]/appdata/swag/dns-conf - This holds the API keys or access information in order for SWAG to setup Let's Encrypt certificates or other DNS information. There are many providers available, however, Cloudflare will most likely be used. In order for a SSL Certificate to be issued, you will have to set this up.
4. /home/[USER]/appdata/swag/www - This folder holds all of the static site stuff, if you decide to make a static site, you can host it through SWAG's configurations. This does take some patience, as getting the stuff exactly right is a bit tricky.

See How do I add a new application to my stack? for more information as to how to use the SWAG proxy-confs folder to add a new application to your hosting repertoire.

Authelia Configuration

Authelia, is our version of an SSO, as once you've logged into Authelia, you have access to all protected links.

Attached is the example configuration.yml file for Authelia.

You can copy/paste this configuration, and then change all values surrounded in square brackets. Make sure to set passwords/tokens where REDACTED occurs. I recommend strings of length 26.

This example configuration is perfect for 2FA with Authy out of the box, if you'd like to enable Duo, please see Authy/Duo Configuration for more details.

```
# yamllint disable rule:comments-indentation
---
###############################################################################
#                   Authelia Configuration                    #
###############################################################################

## Note: the container by default expects to find this file at /config/configuration.yml.

## Certificates directory specifies where Authelia will load trusted certificates (public portion) from in addition to
## the system certificates store.
## They should be in base64 format, and have one of the following extensions: *.cer, *.crt, *.pem.
# certificates_directory: /config/certificates/

## The theme to display: light, dark, grey, auto.
theme: dark

## The secret used to generate JWT tokens when validating user identity by email confirmation. JWT Secret can also be
## set using a secret: https://www.authelia.com/c/secrets
jwt_secret: REDACTED

## Default redirection URL
```

```
##
## If user tries to authenticate without any referer, Authelia does not know where to redirect the user to at the end
## of the authentication process. This parameter allows you to specify the default redirection URL Authelia will use
## in such a case.
##
## Note: this parameter is optional. If not provided, user won't be redirected upon successful authentication.
default_redirection_url: https://[domain.tld]/

## Set the default 2FA method for new users and for when a user has a preferred method configured that has been
## disabled. This setting must be a method that is enabled.
## Options are totp, webauthn, mobile_push.
default_2fa_method: "totp"

##
## Server Configuration
##
server:

  ## The address to listen on.
  host: 0.0.0.0

  ## The port to listen on.
  port: 9091

  ## Set the single level path Authelia listens on.
  ## Must be alphanumeric chars and should not contain any slashes.
  path: "authelia"

  ## Set the path on disk to Authelia assets.
  ## Useful to allow overriding of specific static assets.
  # asset_path: /config/assets/

  ## Enables the pprof endpoint.
  enable_pprof: false

  ## Enables the expvars endpoint.
  enable_expvars: false
```

```
  ## Disables writing the health check vars to /app/.healthcheck.env which makes healthcheck.sh return exit code 0.
  ## This is disabled by default if either /app/.healthcheck.env or /app/healthcheck.sh do not exist.
  disable_healthcheck: false

  ## Authelia by default doesn't accept TLS communication on the server port. This section overrides this behaviour.
  tls:
    ## The path to the DER base64/PEM format private key.
    key: ""

    ## The path to the DER base64/PEM format public certificate.
    certificate: ""

    ## The list of certificates for client authentication.
    client_certificates: []

  ## Server headers configuration/customization.
  headers:

    ## The CSP Template. Read the docs.
    csp_template: ""

  ## Server Buffers configuration.
  # buffers:

    ## Buffers usually should be configured to be the same value.
    ## Explanation at https://www.authelia.com/c/server#buffer-sizes
    ## Read buffer size adjusts the server's max incoming request size in bytes.
    ## Write buffer size does the same for outgoing responses.

    ## Read buffer.
    # read: 4096

    ## Write buffer.
    # write: 4096

  ## Server Timeouts configuration.
  # timeouts:
```

```
    ## Read timeout.
    # read: 6s

    ## Write timeout.
    # write: 6s

    ## Idle timeout.
    # idle: 30s

##
## Log Configuration
##
log:
  ## Level of verbosity for logs: info, debug, trace.
  level: debug

  ## Format the logs are written as: json, text.
  # format: json

  ## File path where the logs will be written. If not set logs are written to stdout.
  # file_path: /config/authelia.log

  ## Whether to also log to stdout when a log_file_path is defined.
  # keep_stdout: false

##
## Telemetry Configuration
##
telemetry:

  ##
  ## Metrics Configuration
  ##
  metrics:
    ## Enable Metrics.
    enabled: false

    ## The address to listen on for metrics. This should be on a different port to the main server.port value.
    address: tcp://0.0.0.0:9959
```

```
    ## Metrics Server Buffers configuration.
    # buffers:

      ## Read buffer.
      # read: 4096

      ## Write buffer.
      # write: 4096

    ## Metrics Server Timeouts configuration.
    # timeouts:

      ## Read timeout.
      # read: 6s

      ## Write timeout.
      # write: 6s

      ## Idle timeout.
      # idle: 30s

##
## TOTP Configuration
##
## Parameters used for TOTP generation.
totp:
  ## Disable TOTP.
  disable: false

  ## The issuer name displayed in the Authenticator application of your choice.
  issuer: [domain.tld]

  ## The TOTP algorithm to use.
  ## It is CRITICAL you read the documentation before changing this option:
  ## https://www.authelia.com/c/totp#algorithm
  algorithm: sha1

  ## The number of digits a user has to input. Must either be 6 or 8.
  ## Changing this option only affects newly generated TOTP configurations.
```

```
  ## It is CRITICAL you read the documentation before changing this option:
  ## https://www.authelia.com/c/totp#digits
  digits: 8

  ## The period in seconds a one-time password is valid for.
  ## Changing this option only affects newly generated TOTP configurations.
  period: 30

  ## The skew controls number of one-time passwords either side of the current one that are valid.
  ## Warning: before changing skew read the docs link below.
  skew: 1
  ## See: https://www.authelia.com/c/totp#input-validation to read
  ## the documentation.

  ## The size of the generated shared secrets. Default is 32 and is sufficient in most use cases, minimum is 20.
  secret_size: 32

##
## WebAuthn Configuration
##
## Parameters used for WebAuthn.
webauthn:
  ## Disable Webauthn.
  disable: false

  ## Adjust the interaction timeout for Webauthn dialogues.
  timeout: 60s

  ## The display name the browser should show the user for when using Webauthn to login/register.
  display_name: Authelia

  ## Conveyance preference controls if we collect the attestation statement including the AAGUID from the
device.
  ## Options are none, indirect, direct.
  attestation_conveyance_preference: indirect

  ## User verification controls if the user must make a gesture or action to confirm they are present.
  ## Options are required, preferred, discouraged.
  user_verification: preferred
```

```
##
## Duo Push API Configuration
##
## Parameters used to contact the Duo API. Those are generated when you protect an application of type
## "Partner Auth API" in the management panel.
# duo_api:
  # disable: false
  # hostname: api-123456789.example.com
  # integration_key: ABCDEF
  ## Secret can also be set using a secret: https://www.authelia.com/c/secrets
  # secret_key: 1234567890abcdefghifjkl
  # enable_self_enrollment: false

##
## NTP Configuration
##
## This is used to validate the servers time is accurate enough to validate TOTP.
ntp:
  ## NTP server address.
  address: "time.cloudflare.com:123"

  ## NTP version.
  version: 4

  ## Maximum allowed time offset between the host and the NTP server.
  max_desync: 3s

  ## Disables the NTP check on startup entirely. This means Authelia will not contact a remote service at all if
you
  ## set this to true, and can operate in a truly offline mode.
  disable_startup_check: false

  ## The default of false will prevent startup only if we can contact the NTP server and the time is out of sync
with
  ## the NTP server more than the configured max_desync. If you set this to true, an error will be logged but
startup
  ## will continue regardless of results.
  disable_failure: false

##
```

```
## Authentication Backend Provider Configuration
##
## Used for verifying user passwords and retrieve information such as email address and groups users belong to.
##
## The available providers are: `file`, `ldap`. You must use only one of these providers.
authentication_backend:

  ## Password Reset Options.
  password_reset:
    ## Disable both the HTML element and the API for reset password functionality.
    disable: false

    ## External reset password url that redirects the user to an external reset portal. This disables the internal reset
    ## functionality.
    custom_url: ""

  ## The amount of time to wait before we refresh data from the authentication backend. Uses duration notation.
  ## To disable this feature set it to 'disable', this will slightly reduce security because for Authelia, users will
  ## always belong to groups they belonged to at the time of login even if they have been removed from them in LDAP.
  ## To force update on every request you can set this to '0' or 'always', this will increase processor demand.
  ## See the below documentation for more information.
  ## Duration Notation docs:  https://www.authelia.com/c/common#duration-notation-format
  ## Refresh Interval docs: https://www.authelia.com/c/1fa#refresh-interval
  refresh_interval: 5m

  ##
  ## LDAP (Authentication Provider)
  ##
  ## This is the recommended Authentication Provider in production
  ## because it allows Authelia to offload the stateful operations
  ## onto the LDAP service.
  ldap:
    ## The LDAP implementation, this affects elements like the attribute utilised for resetting a password.
    ## Acceptable options are as follows:
    ## - 'activedirectory' - For Microsoft Active Directory.
    ## - 'custom' - For custom specifications of attributes and filters.
```

```
    ## This currently defaults to 'custom' to maintain existing behaviour.
    ##
    ## Depending on the option here certain other values in this section have a default value, notably all of the
    ## attribute mappings have a default value that this config overrides, you can read more about these default values
    ## at https://www.authelia.com/c/ldap#defaults
    implementation: custom

    ## The url to the ldap server. Format: <scheme>://<address>[:<port>].
    ## Scheme can be ldap or ldaps in the format (port optional).
    url: ldap://lldap:3890

    ## The dial timeout for LDAP.
    timeout: 5s

    ## Use StartTLS with the LDAP connection.
    start_tls: false

    # tls:
      ## The server subject name to check the servers certificate against during the validation process.
      ## This option is not required if the certificate has a SAN which matches the host portion of the url option.
      # server_name: ldap.example.com

      ## Skip verifying the server certificate entirely. In preference to setting this we strongly recommend you add the
      ## certificate or the certificate of the authority signing the certificate to the certificates directory which is
      ## defined by the `certificates_directory` option at the top of the configuration.
      ## It's important to note the public key should be added to the directory, not the private key.
      ## This option is strongly discouraged but may be useful in some self-signed situations where validation is not
      ## important to the administrator.
      # skip_verify: false

      ## Minimum TLS version for the connection.
      # minimum_version: TLS1.2

      ## Maximum TLS version for the connection.
      # maximum_version: TLS1.3

      ## The certificate chain used with the private_key if the server requests TLS Client Authentication
```

```
## i.e. Mutual TLS.
# certificate_chain: |
  # -----BEGIN CERTIFICATE-----
  # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
  # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
  # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
  # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
  # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSlTqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnl98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5EIU7
  # ysoRzvD12o9QJ/IfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2fIBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----
```

```
    ## The private key used with the certificate_chain if the server requests TLS Client Authentication
    ## i.e. Mutual TLS.
    # private_key: |
      #  -----BEGIN RSA PRIVATE KEY-----
      # MIIEpAIBAAKCAQEA8q/elLl/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
      # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtflCCNK9fBtCfcO3AwDtdZ
      # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
      # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
      # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
      # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
      # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
      # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5fIsunSBwLEpTrdB
      # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
      # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
      # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
      # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
      # Ma4O1cWtGb6RqcNNN/Iukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
      # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KlQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
      # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
      # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
      # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
      # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUIQ3oyokMNwFqvlNos
      # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
      # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziiZ9ov1wbdAQcWRj7RaWnssPFpX
      # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
      # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
      # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
      # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjlODdSi5+icysQGuULKt86h/woz2SQ
      # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
      # DO NOT USE==
      # -----END RSA PRIVATE KEY-----

    ## The distinguished name of the container searched for objects in the directory information tree.
    ## See also: additional_users_dn, additional_groups_dn.
    base_dn: dc=[domain],dc=[tld]

    ## The attribute holding the username of the user. This attribute is used to populate the username in the
session
    ## information. For your information, Microsoft Active Directory usually uses 'sAMAccountName' and
OpenLDAP usually
```

```
    ## uses 'uid'. Beware that this attribute holds the unique identifiers for the users binding the user and the
    ## configuration stored in database. Therefore only single value attributes are allowed and the value must never be
    ## changed once attributed to a user otherwise it would break the configuration for that user. Technically,
    ## non-unique attributes like 'mail' can also be used but we don't recommend using them, we instead advise to use
    ## a filter to perform alternative lookups and the attributes mentioned above (sAMAccountName and uid) to
    ## follow https://www.ietf.org/rfc/rfc2307.txt.
    username_attribute: uid

    ## The additional_users_dn is prefixed to base_dn and delimited by a comma when searching for users.
    ## i.e. with this set to OU=Users and base_dn set to DC=a,DC=com; OU=Users,DC=a,DC=com is searched for users.
    additional_users_dn: ou=people

    ## The users filter used in search queries to find the user profile based on input filled in login form.
    ## Various placeholders are available in the user filter which you can read about in the documentation which can
    ## be found at: https://www.authelia.com/c/ldap#users-filter-replacements
    ##
    ## Recommended settings are as follows:
    ## - Microsoft Active Directory: (&({username_attribute}={input})(objectCategory=person)(objectClass=user))
    ## - OpenLDAP:
    ##   - (&({username_attribute}={input})(objectClass=person))
    ##   - (&({username_attribute}={input})(objectClass=inetOrgPerson))
    ##
    ## To allow sign in both with username and email, one can use a filter like
    ## (&(|({username_attribute}={input})({mail_attribute}={input}))(objectClass=person))
    users_filter: (&({username_attribute}={input})(objectClass=person))

    ## The additional_groups_dn is prefixed to base_dn and delimited by a comma when searching for groups.
    ## i.e. with this set to OU=Groups and base_dn set to DC=a,DC=com; OU=Groups,DC=a,DC=com is searched for groups.
    additional_groups_dn: ou=groups

    ## The groups filter used in search queries to find the groups based on relevant authenticated user.
    ## Various placeholders are available in the groups filter which you can read about in the documentation which can
    ## be found at: https://www.authelia.com/c/ldap#groups-filter-replacements
```

```
    ##
    ## If your groups use the `groupOfUniqueNames` structure use this instead:
    ##    (&(uniqueMember={dn})(objectClass=groupOfUniqueNames))
    groups_filter: (&(member={dn})(objectClass=groupOfUniqueNames))

    ## The attribute holding the name of the group.
    group_name_attribute: cn

    ## The attribute holding the mail address of the user. If multiple email addresses are defined for a user, only the
    ## first one returned by the LDAP server is used.
    mail_attribute: mail

    ## The attribute holding the display name of the user. This will be used to greet an authenticated user.
    display_name_attribute: displayName

    ## Follow referrals returned by the server.
    ## This is especially useful for environments where read-only servers exist. Only implemented for write operations.
    # permit_referrals: false

    ## The username and password of the admin user.
    user: cn=admin,ou=people,dc=[domain],dc=[tld]
    ## Password can also be set using a secret: https://www.authelia.com/c/secrets
    password: "REDACTED"

  ##
  ## File (Authentication Provider)
  ##
  ## With this backend, the users database is stored in a file which is updated when users reset their passwords.
  ## Therefore, this backend is meant to be used in a dev environment and not in production since it prevents Authelia
  ## to be scaled to more than one instance. The options under 'password' have sane defaults, and as it has security
  ## implications it is highly recommended you leave the default values. Before considering changing these settings
  ## please read the docs page below:
  ## https://www.authelia.com/r/passwords#tuning
  ##
  ## Important: Kubernetes (or HA) users must read https://www.authelia.com/t/statelessness
```

```
  ##
  # file:
    # path: /config/users_database.yml
    # watch: false
    # search:
      # email: false
      # case_insensitive: false
    # password:
      # algorithm: argon2
      # argon2:
        # variant: argon2id
        # iterations: 3
        # memory: 65536
        # parallelism: 4
        # key_length: 32
        # salt_length: 16
      # scrypt:
        # iterations: 16
        # block_size: 8
        # parallelism: 1
        # key_length: 32
        # salt_length: 16
      # pbkdf2:
        # variant: sha512
        # iterations: 310000
        # salt_length: 16
      # sha2crypt:
        # variant: sha512
        # iterations: 50000
        # salt_length: 16
      # bcrypt:
        # variant: standard
        # cost: 12


##
## Password Policy Configuration.
##
password_policy:
```

```
  ## The standard policy allows you to tune individual settings manually.
  standard:
    enabled: true

    ## Require a minimum length for passwords.
    min_length: 8

    ## Require a maximum length for passwords.
    max_length: 0

    ## Require uppercase characters.
    require_uppercase: true

    ## Require lowercase characters.
    require_lowercase: true

    ## Require numeric characters.
    require_number: true

    ## Require special characters.
    require_special: true

  ## zxcvbn is a well known and used password strength algorithm. It does not have tunable settings.
  zxcvbn:
    enabled: false

    ## Configures the minimum score allowed.
    min_score: 3

##
## Access Control Configuration
##
## Access control is a list of rules defining the authorizations applied for one resource to users or group of
users.
##
## If 'access_control' is not defined, ACL rules are disabled and the 'bypass' rule is applied, i.e., access is
allowed
## to anyone. Otherwise restrictions follow the rules defined.
##
## Note: One can use the wildcard * to match any subdomain.
```

```
## It must stand at the beginning of the pattern. (example: *.example.com)
##
## Note: You must put patterns containing wildcards between simple quotes for the YAML to be syntactically
correct.
##
## Definition: A 'rule' is an object with the following keys: 'domain', 'subject', 'policy' and 'resources'.
##
## - 'domain' defines which domain or set of domains the rule applies to.
##
## - 'subject' defines the subject to apply authorizations to. This parameter is optional and matching any user if
not
##    provided. If provided, the parameter represents either a user or a group. It should be of the form
##    'user:<username>' or 'group:<groupname>'.
##
## - 'policy' is the policy to apply to resources. It must be either 'bypass', 'one_factor', 'two_factor' or 'deny'.
##
## - 'resources' is a list of regular expressions that matches a set of resources to apply the policy to. This
parameter
##   is optional and matches any resource if not provided.
##
## Note: the order of the rules is important. The first policy matching (domain, resource, subject) applies.
access_control:
  ## Default policy can either be 'bypass', 'one_factor', 'two_factor' or 'deny'. It is the policy applied to any
  ## resource if there is no policy to be applied to the user.
  default_policy: deny

  # networks:
    # - name: internal
    #   networks:
        # - 10.10.0.0/16
        # - 192.168.2.0/24
    # - name: VPN
    #   networks: 10.9.0.0/16

  rules:
    ## Rules applied to everyone
    ## This below policy allows for auth to be accessed at all times,
    ## useful when trying to login without the main page available.
    - domain:
        - 'auth.[domain.tld]'
```

```
  policy: bypass
## This below policy example will exclude a user from using 2FA,
## useful if the user has no way of configuring 2FA.
- domain:
    - '[domain.tld]'
    - '*.[domain.tld]'
  subject:
    - "user:[USERNAME/UID]"
  policy: one_factor
## This below policy example will exclude all users of the people
## group from using 2FA, NOT RECOMMENDED
- domain:
    - '[domain.tld]'
    - '*.[domain.tld]'
  subject:
    - "group:people"
  policy: one_factor
## This below policy will require all users of the LLDAP_Admin group
## in LLDAP to use 2FA BY DEFAULT, this is the strongest policy available
- domain:
    - '[domain.tld]'
    - '*.[domain.tld]'
  subject:
    - "group:lldap_admin"
  policy: two_factor

## Domain Regex examples. Generally we recommend just using a standard domain.
# - domain_regex: '^(?P<User>\w+)\.example\.com$'
#   policy: one_factor
# - domain_regex: '^(?P<Group>\w+)\.example\.com$'
#   policy: one_factor
# - domain_regex:
  #  - '^appgroup-.*\.example\.com$'
  #  - '^appgroup2-.*\.example\.com$'
#   policy: one_factor
# - domain_regex: '^.*\.example\.com$'
#   policy: two_factor

# - domain: 'secure.example.com'
#   policy: one_factor
```

```
## Network based rule, if not provided any network matches.
#   networks:
    # - internal
    # - VPN
    # - 192.168.1.0/24
    # - 10.0.0.1

# - domain:
    # - 'secure.example.com'
    # - 'private.example.com'
#   policy: two_factor

# - domain: 'singlefactor.example.com'
#   policy: one_factor

## Rules applied to 'admins' group
# - domain: 'mx2.mail.example.com'
#   subject: 'group:admins'
#   policy: deny

# - domain: '*.example.com'
#   subject:
    # - 'group:admins'
    # - 'group:moderators'
#   policy: two_factor

## Rules applied to 'dev' group
# - domain: 'dev.example.com'
#   resources:
    # - '^/groups/dev/.*$'
#   subject: 'group:dev'
#   policy: two_factor

## Rules applied to user 'john'
# - domain: 'dev.example.com'
#   resources:
    # - '^/users/john/.*$'
#   subject: 'user:john'
#   policy: two_factor
```

```
    ## Rules applied to user 'harry'
    # - domain: 'dev.example.com'
    #   resources:
        # - '^/users/harry/.*$'
    #   subject: 'user:harry'
    #   policy: two_factor

    ## Rules applied to user 'bob'
    # - domain: '*.mail.example.com'
    #   subject: 'user:bob'
    #   policy: two_factor
    # - domain: 'dev.example.com'
    #   resources:
    #     - '^/users/bob/.*$'
    #   subject: 'user:bob'
    #   policy: two_factor

##
## Session Provider Configuration
##
## The session cookies identify the user once logged in.
## The available providers are: `memory`, `redis`. Memory is the provider unless redis is defined.
session:
  ## The name of the session cookie.
  name: authelia_session

  ## The domain to protect.
  ## Note: the authenticator must also be in that domain.
  ## If empty, the cookie is restricted to the subdomain of the issuer.
  domain: [domain.tld]

  ## Sets the Cookie SameSite value. Possible options are none, lax, or strict.
  ## Please read https://www.authelia.com/c/session#same_site
  same_site: lax

  ## The secret to encrypt the session data. This is only used with Redis / Redis Sentinel.
  ## Secret can also be set using a secret: https://www.authelia.com/c/secrets
  secret: REDACTED

  ## The value for expiration, inactivity, and remember_me_duration are in seconds or the duration notation
```

```
format.
  ## See: https://www.authelia.com/c/common#duration-notation-format
  ## All three of these values affect the cookie/session validity period. Longer periods are considered less secure
  ## because a stolen cookie will last longer giving attackers more time to spy or attack.

  ## The time before the cookie expires and the session is destroyed if remember me IS NOT selected.
  expiration: 1h

  ## The inactivity time before the session is reset. If expiration is set to 1h, and this is set to 5m, if the user
  ## does not select the remember me option their session will get destroyed after 1h, or after 5m since the last
time
  ## Authelia detected user activity.
  inactivity: 5m

  ## The time before the cookie expires and the session is destroyed if remember me IS selected.
  ## Value of -1 disables remember me.
  remember_me_duration: 2M

  ##
  ## Redis Provider
  ##
  ## Important: Kubernetes (or HA) users must read https://www.authelia.com/t/statelessness
  ##
  redis:
    host: redis
    port: 6379
    ## Use a unix socket instead
    # host: /var/run/redis/redis.sock

    ## Username used for redis authentication. This is optional and a new feature in redis 6.0.
    # username: authelia

    ## Password can also be set using a secret: https://www.authelia.com/c/secrets
    password: REDACTED

    ## This is the Redis DB Index https://redis.io/commands/select (sometimes referred to as database number,
DB, etc).
    database_index: 0

    ## The maximum number of concurrent active connections to Redis.
```

```
    maximum_active_connections: 10

    ## The target number of idle connections to have open ready for work. Useful when opening connections is
slow.
    minimum_idle_connections: 0

    ## The Redis TLS configuration. If defined will require a TLS connection to the Redis instance(s).
    # tls:
      ## The server subject name to check the servers certificate against during the validation process.
      ## This option is not required if the certificate has a SAN which matches the host option.
      # server_name: myredis.example.com

      ## Skip verifying the server certificate entirely. In preference to setting this we strongly recommend you
add the
      ## certificate or the certificate of the authority signing the certificate to the certificates directory which is
      ## defined by the `certificates_directory` option at the top of the configuration.
      ## It's important to note the public key should be added to the directory, not the private key.
      ## This option is strongly discouraged but may be useful in some self-signed situations where validation is
not
      ## important to the administrator.
      # skip_verify: false

      ## Minimum TLS version for the connection.
      # minimum_version: TLS1.2

      ## Maximum TLS version for the connection.
      # maximum_version: TLS1.3

      ## The certificate chain used with the private_key if the server requests TLS Client Authentication
      ## i.e. Mutual TLS.
      # certificate_chain: |
        # -----BEGIN CERTIFICATE-----
        # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
        # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
        # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
        # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
        # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
        # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
        # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
        # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
```

```
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSITqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnI98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5ElU7
  # ysoRzvD12o9QJ/lfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2flBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----

## The private key used with the certificate_chain if the server requests TLS Client Authentication
## i.e. Mutual TLS.
# private_key: |
  #  -----BEGIN RSA PRIVATE KEY-----
  # MIIEpAIBAAKCAQEA8q/elLI/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
  # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtflCCNK9fBtCfcO3AwDtdZ
  # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
  # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
  # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
  # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
  # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
```

```
    # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5flsunSBwLEpTrdB
    # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
    # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
    # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
    # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
    # Ma4O1cWtGb6RqcNNN/lukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
    # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KIQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
    # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
    # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
    # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
    # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUIQ3oyokMNwFqvlNos
    # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
    # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziiZ9ov1wbdAQcWRj7RaWnssPFpX
    # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
    # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
    # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
    # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjIODdSi5+icysQGuULKt86h/woz2SQ
    # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
    # DO NOT USE==
    # -----END RSA PRIVATE KEY-----

## The Redis HA configuration options.
## This provides specific options to Redis Sentinel, sentinel_name must be defined (Master Name).
# high_availability:
  ## Sentinel Name / Master Name.
  # sentinel_name: mysentinel

  ## Specific username for Redis Sentinel. The node username and password is configured above.
  # sentinel_username: sentinel_specific_user

  ## Specific password for Redis Sentinel. The node username and password is configured above.
  # sentinel_password: sentinel_specific_pass

  ## The additional nodes to pre-seed the redis provider with (for sentinel).
  ## If the host in the above section is defined, it will be combined with this list to connect to sentinel.
  ## For high availability to be used you must have either defined; the host above or at least one node below.
  # nodes:
    # - host: sentinel-node1
    #   port: 6379
    # - host: sentinel-node2
```

```
      #   port: 6379

    ## Choose the host with the lowest latency.
    # route_by_latency: false

    ## Choose the host randomly.
    # route_randomly: false

##
## Regulation Configuration
##
## This mechanism prevents attackers from brute forcing the first factor. It bans the user if too many attempts are made
## in a short period of time.
regulation:
  ## The number of failed login attempts before user is banned. Set it to 0 to disable regulation.
  max_retries: 3

  ## The time range during which the user can attempt login before being banned. The user is banned if the
  ## authentication failed 'max_retries' times in a 'find_time' seconds window. Find Time accepts duration notation.
  ## See: https://www.authelia.com/c/common#duration-notation-format
  find_time: 10m

  ## The length of time before a banned user can login again. Ban Time accepts duration notation.
  ## See: https://www.authelia.com/c/common#duration-notation-format
  ban_time: 12h

##
## Storage Provider Configuration
##
## The available providers are: `local`, `mysql`, `postgres`. You must use one and only one of these providers.
storage:
  ## The encryption key that is used to encrypt sensitive information in the database. Must be a string with a minimum
  ## length of 20. Please see the docs if you configure this with an undesirable key and need to change it, you MUST use
  ## the CLI to change this in the database if you want to change it from a previously configured value.
  encryption_key: REDACTED
```

```
  ##
  ## Local (Storage Provider)
  ##
  ## This stores the data in a SQLite3 Database.
  ## This is only recommended for lightweight non-stateful installations.
  ##
  ## Important: Kubernetes (or HA) users must read https://www.authelia.com/t/statelessness
  ##
  # local:
    ## Path to the SQLite3 Database.
    # path: /config/db.sqlite3

  ##
  ## MySQL / MariaDB (Storage Provider)
  ##
  mysql:
    host: mysql
    port: 3306
    database: authelia
    username: authelia
    ## Password can also be set using a secret: https://www.authelia.com/c/secrets
    password: REDACTED
    timeout: 5s

    ## MySQL TLS settings. Configuring this requires TLS.
    # tls:
      ## The server subject name to check the servers certificate against during the validation process.
      ## This option is not required if the certificate has a SAN which matches the host option.
      # server_name: mysql.example.com

      ## Skip verifying the server certificate entirely. In preference to setting this we strongly recommend you
add the
      ## certificate or the certificate of the authority signing the certificate to the certificates directory which is
      ## defined by the `certificates_directory` option at the top of the configuration.
      ## It's important to note the public key should be added to the directory, not the private key.
      ## This option is strongly discouraged but may be useful in some self-signed situations where validation is
not
      ## important to the administrator.
      # skip_verify: false
```

```
## Minimum TLS version for the connection.
# minimum_version: TLS1.2

## Maximum TLS version for the connection.
# maximum_version: TLS1.3

## The certificate chain used with the private_key if the server requests TLS Client Authentication
## i.e. Mutual TLS.
# certificate_chain: |
  # -----BEGIN CERTIFICATE-----
  # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
  # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
  # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
  # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
  # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSlTqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnl98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5ElU7
  # ysoRzvD12o9QJ/lfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
```

```
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2fIBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----

## The private key used with the certificate_chain if the server requests TLS Client Authentication
## i.e. Mutual TLS.
# private_key: |
  #  -----BEGIN RSA PRIVATE KEY-----
  # MIIEpAIBAAKCAQEA8q/elLI/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
  # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtflCCNK9fBtCfcO3AwDtdZ
  # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
  # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
  # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
  # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
  # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
  # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5fIsunSBwLEpTrdB
  # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
  # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
  # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
  # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
  # Ma4O1cWtGb6RqcNNN/lukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
  # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KlQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
  # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
  # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
  # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
  # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUlQ3oyokMNwFqvlNos
  # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
  # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziiZ9ov1wbdAQcWRj7RaWnssPFpX
  # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
  # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
  # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
  # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjlODdSi5+icysQGuULKt86h/woz2SQ
  # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
  # DO NOT USE==
  # -----END RSA PRIVATE KEY-----

##
```

```
  ## PostgreSQL (Storage Provider)
  ##
  # postgres:
    # host: 127.0.0.1
    # port: 5432
    # database: authelia
    # schema: public
    # username: authelia
    ## Password can also be set using a secret: https://www.authelia.com/c/secrets
    # password: mypassword
    # timeout: 5s

    ## PostgreSQL TLS settings. Configuring this requires TLS.
    # tls:
      ## The server subject name to check the servers certificate against during the validation process.
      ## This option is not required if the certificate has a SAN which matches the host option.
      # server_name: postgres.example.com

      ## Skip verifying the server certificate entirely. In preference to setting this we strongly recommend you
add the
      ## certificate or the certificate of the authority signing the certificate to the certificates directory which is
      ## defined by the `certificates_directory` option at the top of the configuration.
      ## It's important to note the public key should be added to the directory, not the private key.
      ## This option is strongly discouraged but may be useful in some self-signed situations where validation is
not
      ## important to the administrator.
      # skip_verify: false

      ## Minimum TLS version for the connection.
      # minimum_version: TLS1.2

      ## Maximum TLS version for the connection.
      # maximum_version: TLS1.3

      ## The certificate chain used with the private_key if the server requests TLS Client Authentication
      ## i.e. Mutual TLS.
      # certificate_chain: |
        # -----BEGIN CERTIFICATE-----
        # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
        # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
```

```
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
  # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
  # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
  # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
  # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSlTqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnl98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5ElU7
  # ysoRzvD12o9QJ/lfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2fIBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----

## The private key used with the certificate_chain if the server requests TLS Client Authentication
## i.e. Mutual TLS.
# private_key: |
  #  -----BEGIN RSA PRIVATE KEY-----
  # MIIEpAIBAAKCAQEA8q/elLI/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
```

```
      # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtflCCNK9fBtCfcO3AwDtdZ
      # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
      # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
      # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
      # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
      # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
      # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5flsunSBwLEpTrdB
      # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
      # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
      # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
      # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
      # Ma4O1cWtGb6RqcNNN/lukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
      # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KlQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
      # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
      # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
      # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
      # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUlQ3oyokMNwFqvlNos
      # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
      # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziiZ9ov1wbdAQcWRj7RaWnssPFpX
      # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
      # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
      # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
      # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjIODdSi5+icysQGuULKt86h/woz2SQ
      # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
      # DO NOT USE==
      # -----END RSA PRIVATE KEY-----

##
## Notification Provider
##
## Notifications are sent to users when they require a password reset, a Webauthn registration or a TOTP
registration.
## The available providers are: filesystem, smtp. You must use only one of these providers.
notifier:
  ## You can disable the notifier startup check by setting this to true.
  disable_startup_check: false

  ##
  ## File System (Notification Provider)
  ##
```

```
  ## Important: Kubernetes (or HA) users must read https://www.authelia.com/t/statelessness
  ##
#  filesystem:
#    filename: /config/notification.txt

  ##
  ## SMTP (Notification Provider)
  ##
  ## Use a SMTP server for sending notifications. Authelia uses the PLAIN or LOGIN methods to authenticate.
  ## [Security] By default Authelia will:
  ##   - force all SMTP connections over TLS including unauthenticated connections
  ##     - use the disable_require_tls boolean value to disable this requirement
  ##       (only works for unauthenticated connections)
  ##   - validate the SMTP server x509 certificate during the TLS handshake against the hosts trusted certificates
  ##     (configure in tls section)
  smtp:
    ## The SMTP host to connect to.
    host: smtp.gmail.com

    ## The port to connect to the SMTP host on.
    port: 587

    ## The connection timeout.
    timeout: 5s

    ## The username used for SMTP authentication.
    username: [emailaddressrelay]@gmail.com

    ## The password used for SMTP authentication.
    ## Can also be set using a secret: https://www.authelia.com/c/secrets
    password: REDACTED

    ## The sender is used to is used for the MAIL FROM command and the FROM header.
    ## If this is not defined and the username is an email, we use the username as this value. This can either be
just
    ## an email address or the RFC5322 'Name <email address>' format.
    sender: "Authelia <admin@[domain.tld]>"

    ## HELO/EHLO Identifier. Some SMTP Servers may reject the default of localhost.
    identifier: [domain.tld]
```

```
    ## Subject configuration of the emails sent. {title} is replaced by the text from the notifier.
    subject: "[Authelia] {title}"

    ## This address is used during the startup check to verify the email configuration is correct.
    ## It's not important what it is except if your email server only allows local delivery.
    startup_check_address: [emailaddressrelay]@gmail.com

    ## By default we require some form of TLS. This disables this check though is not advised.
    # disable_require_tls: false

    ## Disables sending HTML formatted emails.
    # disable_html_emails: false

    tls:
      ## The server subject name to check the servers certificate against during the validation process.
      ## This option is not required if the certificate has a SAN which matches the host option.
      server_name: smtp.gmail.com

      ## Skip verifying the server certificate entirely. In preference to setting this we strongly recommend you
add the
      ## certificate or the certificate of the authority signing the certificate to the certificates directory which is
      ## defined by the `certificates_directory` option at the top of the configuration.
      ## It's important to note the public key should be added to the directory, not the private key.
      ## This option is strongly discouraged but may be useful in some self-signed situations where validation is
not
      ## important to the administrator.
      # skip_verify: false

      ## Minimum TLS version for the connection.
      # minimum_version: TLS1.2

      ## Maximum TLS version for the connection.
      # maximum_version: TLS1.3

      ## The certificate chain used with the private_key if the server requests TLS Client Authentication
      ## i.e. Mutual TLS.
      # certificate_chain: |
        # -----BEGIN CERTIFICATE-----
        # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
```

```
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
  # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
  # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
  # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
  # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSlTqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnl98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5ElU7
  # ysoRzvD12o9QJ/lfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2fIBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----

## The private key used with the certificate_chain if the server requests TLS Client Authentication
## i.e. Mutual TLS.
# private_key: |
  #  -----BEGIN RSA PRIVATE KEY-----
```

```
      # MIIEpAIBAAKCAQEA8q/elLl/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
      # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtflCCNK9fBtCfcO3AwDtdZ
      # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
      # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
      # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
      # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
      # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
      # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5flsunSBwLEpTrdB
      # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
      # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
      # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
      # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
      # Ma4O1cWtGb6RqcNNN/lukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
      # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KlQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
      # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
      # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
      # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
      # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUIQ3oyokMNwFqvlNos
      # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
      # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziiZ9ov1wbdAQcWRj7RaWnssPFpX
      # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
      # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
      # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
      # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjlODdSi5+icysQGuULKt86h/woz2SQ
      # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
      # DO NOT USE==
      # -----END RSA PRIVATE KEY-----

##
## Identity Providers
##
# identity_providers:

  ##
  ## OpenID Connect (Identity Provider)
  ##
  ## It's recommended you read the documentation before configuration of this section:
  ## https://www.authelia.com/c/oidc
  # oidc:
    ## The hmac_secret is used to sign OAuth2 tokens (authorization code, access tokens and refresh tokens).
```

```
## HMAC Secret can also be set using a secret: https://www.authelia.com/c/secrets
# hmac_secret: this_is_a_secret_abc123abc123abc

## The issuer_certificate_chain is an optional PEM encoded certificate chain. It's used in conjunction with the
## issuer_private_key to sign JWT's. All certificates in the chain must be within the validity period, and every
## certificate included must be signed by the certificate immediately after it if provided.
# issuer_certificate_chain: |
  # -----BEGIN CERTIFICATE-----
  # MIIC5jCCAc6gAwIBAgIRAK4Sj7FiN6PXo/urPfO4E7owDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAPKv3pSyP4ozGEiVLJ14dIWFCEGEgq7WUMI0SZZqQA2ID0L59U/Q
  # /Usyy7uC9gfMUzODTpANtkOjFQcQAsxlR1FOjVBrX5QgjSvXwbQn3DtwMA7XWSl6
  # LuYx2rBYSlMSN5UZQm/RxMtXfLK2b51WgEEYDFi+nECSqKzR4R54eOPkBEWRfvuY
  # 91AMjlhpivg8e4JWkq4LVQUKbmiFYwIdK8XQiN4blY9WwXwJFYs5sQ/UYMwBFi0H
  # kWOh7GEjfxgoUOPauIueZSMSlQp7zqAH39N0ZSYb6cS0Npj57QoWZSY3ak87ebcR
  # Nf4rCvZLby7LoN7qYCKxmCaDD3x2+NYpWH8CAwEAAaM1MDMwDgYDVR0PAQH/BAQD
  # AgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcN
  # AQELBQADggEBAHSITqIQSNzonFl3DzxHPEzr2hp6peo45buAAtu8FZHoA+U7Icfh
  # /ZXjPg7Xz+hgFwM/DTNGXkMWacQA/PaNWvZspgRJf2AXvNbMSs2UQODr7Tbv+Fb4
  # lyblmMUNYFMCFVAMU0eIxXAFq2qcwv8UMcQFT0Z/35s6PVOakYnAGGQjTfp5Ljuq
  # wsdc/xWmM0cHWube6sdRRUD7SY20KU/kWzl8iFO0VbSSrDf1AlEhnLEkp1SPaxXg
  # OdBnl98MeoramNiJ7NT6Jnyb3zZ578fjaWfThiBpagItI8GZmG4s4Ovh2JbheN8i
  # ZsjNr9jqHTjhyLVbDRlmJzcqoj4JhbKs6/I^invalid DO NOT USE=
  # -----END CERTIFICATE-----
  # -----BEGIN CERTIFICATE-----
  # MIIDBDCCAeygAwIBAgIRALJsPg21kA0zY4F1wUCIuoMwDQYJKoZIhvcNAQELBQAw
  # EzERMA8GA1UEChMIQXV0aGVsaWEwHhcNNzAwMTAxMDAwMDAwWhcNNzEwMTAxMDAw
  # MDAwWjATMREwDwYDVQQKEwhBdXRoZWxpYTCCASIwDQYJKoZIhvcNAQEBBQADggEP
  # ADCCAQoCggEBAMXHBvVxUzYk0u34/DINMSF+uiOekKOAjOrC6Mi9Ww8ytPVO7t2S
  # zfTvM+XnEJqkFQFgimERfG/eGhjF9XIEY6LtnXe8ATvOK4nTwdufzBaoeQu3Gd50
  # 5VXr6OHRo//ErrGvFXwP3g8xLePABsi/fkH3oDN+ztewOBMDzpd+KgTrk8ysv2ou
  # kNRMKFZZqASvCgv0LD5KWvUCnL6wgf1oTXG7aztduA4oSkUP321GpOmBC5+5ElU7
  # ysoRzvD12o9QJ/IfEaulIX06w9yVMo60C/h6A3U6GdkT1SiyTIqR7v7KU/IWd/Qi
  # Lfftcj91VhCmJ73Meff2e2S2PrpjdXbG5FMCAwEAAaNTMFEwDgYDVR0PAQH/BAQD
  # AgKkMA8GA1UdJQQIMAYGBFUdJQAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
  # Z7AtA3mzFc0InSBA5fiMfeLXA3owDQYJKoZIhvcNAQELBQADggEBAEE5hm1mtlk/
  # kviCoHH4evbpw7rxPxDftIQlqYTtvMM4eWY/6icFoSZ4fUHEWYyps8SsPu/8f2tf
  # 71LGgZn0FdHi1QU2H8m0HHK7TFw+5Q6RLrLdSyk0PltJ71s9en7r8pX820nAFEHZ
  # HkOSfJZ7B5hFgUDkMtVM6bardXAhoqcMk4YCU96e9d4PB4eI+xGc+mNuYvov3RbB
```

```
  # D0s8ICyojeyPVLerz4wHjZu68Z5frAzhZ68YbzNs8j2fIBKKHkHyLG1iQyF+LJVj
  # 2PjCP+auJsj6fQQpMGoyGtpLcSDh+ptcTngUD8JsWipzTCjmaNqdPHAOYmcgtf4b
  # qocikt3WAdU^invalid DO NOT USE=
  # -----END CERTIFICATE-----

## The issuer_private_key is used to sign the JWT forged by OpenID Connect.
## Issuer Private Key can also be set using a secret: https://www.authelia.com/c/secrets
# issuer_private_key: |
  # -----BEGIN RSA PRIVATE KEY-----
  # MIIEpAIBAAKCAQEA8q/elLI/ijMYSJUsnXh0hYUIQYSCrtZQwjRJlmpADYgPQvn1
  # T9D9SzLLu4L2B8xTM4NOkA22Q6MVBxACzGVHUU6NUGtfICCNK9fBtCfcO3AwDtdZ
  # KXou5jHasFhKUxI3lRlCb9HEy1d8srZvnVaAQRgMWL6cQJKorNHhHnh44+QERZF+
  # +5j3UAyOWGmK+Dx7glaSrgtVBQpuaIVjAh0rxdCI3huVj1bBfAkVizmxD9RgzAEW
  # LQeRY6HsYSN/GChQ49q4i55lIxKVCnvOoAff03RlJhvpxLQ2mPntChZlJjdqTzt5
  # txE1/isK9ktvLsug3upgIrGYJoMPfHb41ilYfwIDAQABAoIBAQDTOdFf2JjHH1um
  # aPgRAvNf9v7Nj5jytaRKs5nM6iNf46ls4QPreXnMhqSeSwj6lpNgBYxOgzC9Q+cc
  # Y4ob/paJJPaIJTxmP8K/gyWcOQlNToL1l+eJ20eQoZm23NGr5flsunSBwLEpTrdB
  # ENqqtcwhW937K8Pxy/Q1nuLyU2bc6Tn/ivLozc8n27dpQWWKh8537VY7ancIaACr
  # LJJLYxKqhQpjtBWAyCDvZQirnAOm9KnvIHaGXIswCZ4Xbsu0Y9NL+woARPyRVQvG
  # jfxy4EmO9s1s6y7OObSukwKDSNihAKHx/VIbvVWx8g2Lv5fGOa+J2Y7o9Qurs8t5
  # BQwMTt0BAoGBAPUw5Z32EszNepAeV3E2mPFUc5CLiqAxagZJuNDO2pKtyN29ETTR
  # Ma4O1cWtGb6RqcNNN/lukfkdk27Q5nC9VJSUUPYelOLc1WYOoUf6oKRzE72dkMQV
  # R4bf6TkjD+OVR17fAfkswkGahZ5XA7j48KIQ+YC4jbnYKSxZTYyKPjH/AoGBAP1i
  # tqXt36OVlP+y84wWqZSjMelBIVa9phDVGJmmhz3i1cMni8eLpJzWecA3pfnG6Tm9
  # ze5M4whASleEt+M00gEvNaU9ND+z0wBfi+/DwJYIbv8PQdGrBiZFrPhTPjGQUldR
  # lXccV2meeLZv7TagVxSi3DO6dSJfSEHyemd5j9mBAoGAX8Hv+0gOQZQCSOTAq8Nx
  # 6dZcp9gHlNaXnMsP9eTDckOSzh636JPGvj6m+GPJSSbkURUIQ3oyokMNwFqvlNos
  # fTaLhAOfjBZI9WnDTTQxpugWjphJ4HqbC67JC/qIiw5S6FdaEvGLEEoD4zoChywZ
  # 9oGAn+fz2d/0/JAH/FpFPgsCgYEAp/ipZgPzziZ9ov1wbdAQcWRj7RaWnssPFpX
  # jXwEiXT3CgEMO4MJ4+KWIWOChrti3qFBg6i6lDyyS6Qyls7sLFbUdC7HlTcrOEMe
  # rBoTcCI1GqZNlqWOVQ65ZIEiaI7o1vPBZo2GMQEZuq8mDKFsOMThvvTrM5cAep84
  # n6HJR4ECgYABWcbsSnr0MKvVth/inxjbKapbZnp2HUCuw87Ie5zK2Of/tbC20wwk
  # yKw3vrGoE3O1t1g2m2tn8UGGASeZ842jZWjIODdSi5+icysQGuULKt86h/woz2SQ
  # 27GoE2i5mh6Yez6VAYbUuns3FcwIsMyWLq043Tu2DNkx9ijOOAuQzw^invalid..
  # DO NOT USE==
  # -----END RSA PRIVATE KEY-----

## The lifespans configure the expiration for these token types.
# access_token_lifespan: 1h
# authorize_code_lifespan: 1m
```

```
    # id_token_lifespan: 1h
    # refresh_token_lifespan: 90m

    ## Enables additional debug messages.
    # enable_client_debug_messages: false

    ## SECURITY NOTICE: It's not recommended changing this option and values below 8 are strongly
discouraged.
    # minimum_parameter_entropy: 8

    ## SECURITY NOTICE: It's not recommended changing this option, and highly discouraged to have it set to
'never'
    ## for security reasons.
    # enforce_pkce: public_clients_only

    ## Cross-Origin Resource Sharing (CORS) settings.
    # cors:
      ## List of endpoints in addition to the metadata endpoints to permit cross-origin requests on.
      # endpoints:
        #  - authorization
        #  - token
        #  - revocation
        #  - introspection
        #  - userinfo

      ## List of allowed origins.
      ## Any origin with https is permitted unless this option is configured or the
      ## allowed_origins_from_client_redirect_uris option is enabled.
      # allowed_origins:
        # - https://example.com

      ## Automatically adds the origin portion of all redirect URI's on all clients to the list of allowed_origins,
      ## provided they have the scheme http or https and do not have the hostname of localhost.
      # allowed_origins_from_client_redirect_uris: false

    ## Clients is a list of known clients and their configuration.
    # clients:
      # -
        ## The ID is the OpenID Connect ClientID which is used to link an application to a configuration.
        # id: myapp
```

```
        ## The description to show to users when they end up on the consent screen. Defaults to the ID above.
        # description: My Application

        ## The client secret is a shared secret between Authelia and the consumer of this client.
        # secret: this_is_a_secret

        ## Sector Identifiers are occasionally used to generate pairwise subject identifiers. In most cases this is not
        ## necessary. Read the documentation for more information.
        ## The subject identifier must be the host component of a URL, which is a domain name with an optional
port.
        # sector_identifier: example.com

        ## Sets the client to public. This should typically not be set, please see the documentation for usage.
        # public: false

        ## The policy to require for this client; one_factor or two_factor.
        # authorization_policy: two_factor

        ## The consent mode controls how consent is obtained.
        # consent_mode: auto

        ## This value controls the duration a consent on this client remains remembered when the consent mode
is
        ## configured as 'auto' or 'pre-configured'.
        # pre_configured_consent_duration: 1w

        ## Audience this client is allowed to request.
        # audience: []

        ## Scopes this client is allowed to request.
        # scopes:
          # - openid
          # - groups
          # - email
          # - profile

        ## Redirect URI's specifies a list of valid case-sensitive callbacks for this client.
        # redirect_uris:
        # - https://oidc.example.com:8080/oauth2/callback
```

```
      ## Grant Types configures which grants this client can obtain.
      ## It's not recommended to define this unless you know what you're doing.
      # grant_types:
        # - refresh_token
        # - authorization_code

      ## Response Types configures which responses this client can be sent.
      ## It's not recommended to define this unless you know what you're doing.
      # response_types:
        # - code

      ## Response Modes configures which response modes this client supports.
      # response_modes:
        # - form_post
        # - query
        # - fragment

      ## The algorithm used to sign userinfo endpoint responses for this client, either none or RS256.
      # userinfo_signing_algorithm: none
...
```

LLDAP Configuration

LLDAP is the Light Lightweight Directory Access Protocol, and for our purposes, will store all the information that we have on our users, like username, email, and hashed password. These are the typical values, and only a few more exist in this light implementation of LDAP. I include a sample config file if you don't want to set values in the Docker Compose file.

As with the Authelia Configuration, please change things in square brackets to your relevant values. This file uses REPLACE_WITH to tell you where to replace other such values like secrets/tokens.

```
## Default configuration for Docker.
## All the values can be overridden through environment variables, prefixed
## with "LLDAP_". For instance, "ldap_port" can be overridden with the
## "LLDAP_LDAP_PORT" variable.

## Tune the logging to be more verbose by setting this to be true.
## You can set it with the LLDAP_VERBOSE environment variable.
# verbose=false

## The host address that the LDAP server will be bound to.
## To enable IPv6 support, simply switch "ldap_host" to "::":
## To only allow connections from localhost (if you want to restrict to local self-hosted services),
## change it to "127.0.0.1" ("::1" in case of IPv6)".
ldap_host = "0.0.0.0"

## The port on which to have the LDAP server.
ldap_port = 3890

## The host address that the HTTP server will be bound to.
## To enable IPv6 support, simply switch "http_host" to "::".
## To only allow connections from localhost (if you want to restrict to local self-hosted services),
## change it to "127.0.0.1" ("::1" in case of IPv6)".
http_host = "0.0.0.0"

## The port on which to have the HTTP server, for user login and
## administration.
```

```
http_port = 17170

## The public URL of the server, for password reset links.
#http_url = "http://localhost"

## Random secret for JWT signature.
## This secret should be random, and should be shared with application
## servers that need to consume the JWTs.
## Changing this secret will invalidate all user sessions and require
## them to re-login.
## You should probably set it through the LLDAP_JWT_SECRET environment
## variable from a secret ".env" file.
## This can also be set from a file's contents by specifying the file path
## in the LLDAP_JWT_SECRET_FILE environment variable
## You can generate it with (on linux):
## LC_ALL=C tr -dc 'A-Za-z0-9!#%&'\''()*+,-./:;<=>?@[\]^_{|}~' </dev/urandom | head -c 32; echo ''
jwt_secret = "REPLACE_WITH_RANDOM"

## Base DN for LDAP.
## This is usually your domain name, and is used as a
## namespace for your users. The choice is arbitrary, but will be needed
## to configure the LDAP integration with other services.
## The sample value is for "example.com", but you can extend it with as
## many "dc" as you want, and you don't actually need to own the domain
## name.
ldap_base_dn = "dc=[DOMAIN],dc=[TLD]"

## Admin username.
## For the LDAP interface, a value of "admin" here will create the LDAP
## user "cn=admin,ou=people,dc=example,dc=com" (with the base DN above).
## For the administration interface, this is the username.
ldap_user_dn = "admin"

## Admin email.
## Email for the admin account. It is only used when initially creating
## the admin user, and can safely be omitted.
ldap_user_email = "admin@[DOMAIN.TLD]"

## Admin password.
## Password for the admin account, both for the LDAP bind and for the
```

```
## administration interface. It is only used when initially creating
## the admin user.
## It should be minimum 8 characters long.
## You can set it with the LLDAP_LDAP_USER_PASS environment variable.
## This can also be set from a file's contents by specifying the file path
## in the LLDAP_LDAP_USER_PASS_FILE environment variable
## Note: you can create another admin user for user administration, this
## is just the default one.
ldap_user_pass = "REPLACE_WITH_PASSWORD"

## Database URL.
## This encodes the type of database (SQlite, MySQL, or PostgreSQL)
## , the path, the user, password, and sometimes the mode (when
## relevant).
## Note: SQlite should come with "?mode=rwc" to create the DB
## if not present.
## Example URLs:
##  - "postgres://postgres-user:password@postgres-server/my-database"
##  - "mysql://mysql-user:password@mysql-server/my-database"
##
## This can be overridden with the LLDAP_DATABASE_URL env variable.
database_url = "sqlite:///data/users.db?mode=rwc"

## Private key file.
## Contains the secret private key used to store the passwords safely.
## Note that even with a database dump and the private key, an attacker
## would still have to perform an (expensive) brute force attack to find
## each password.
## Randomly generated on first run if it doesn't exist.
## Alternatively, you can use key_seed to override this instead of relying on
## a file.
## Env variable: LLDAP_KEY_FILE
key_file = "/data/private_key"

## Seed to generate the server private key, see key_file above.
## This can be any random string, the recommendation is that it's at least 12
## characters long.
## Env variable: LLDAP_KEY_SEED
#key_seed = "RanD0m STR1ng"
```

```
## Ignored attributes.
## Some services will request attributes that are not present in LLDAP. When it
## is the case, LLDAP will warn about the attribute being unknown. If you want
## to ignore the attribute and the service works without, you can add it to this
## list to silence the warning.
#ignored_user_attributes = [ "sAMAccountName" ]
#ignored_group_attributes = [ "mail", "userPrincipalName" ]

## Options to configure SMTP parameters, to send password reset emails.
## To set these options from environment variables, use the following format
## (example with "password"): LLDAP_SMTP_OPTIONS__PASSWORD
[smtp_options]
## Whether to enabled password reset via email, from LLDAP.
#enable_password_reset=true
## The SMTP server.
#server="smtp.gmail.com"
## The SMTP port.
#port=587
## How the connection is encrypted, either "NONE" (no encryption), "TLS" or "STARTTLS".
#smtp_encryption = "TLS"
## The SMTP user, usually your email address.
#user="sender@gmail.com"
## The SMTP password.
#password="password"
## The header field, optional: how the sender appears in the email. The first
## is a free-form name, followed by an email between <>.
#from="LLDAP Admin <sender@gmail.com>"
## Same for reply-to, optional.
#reply_to="Do not reply <noreply@localhost>"

## Options to configure LDAPS.
## To set these options from environment variables, use the following format
## (example with "port"): LLDAP_LDAPS_OPTIONS__PORT
[ldaps_options]
## Whether to enable LDAPS.
#enabled=true
## Port on which to listen.
#port=6360
## Certificate file.
#cert_file="/data/cert.pem"
```

```
## Certificate key file.
#key_file="/data/key.pem"
```

Authy/Duo Configuration

This is like one of those "Choose Your Own Adventure" books! Pick one, not the other, it may be hard to swap later on. You may be able to give your user a choice too! 2FA is handled by either Authelia using an 8 digit code that generates through Authy or it can be passed off to Duo to handle, and then "reinjected" back into the authentication process later on. I'll outline steps for both, and you can choose the method(s) you want your users to be able to 2FA.

Authy is probably the easiest, and the cheapest way to configure 2FA, especially if you are going to have more than 5 users.

The default for the SALA/D Stack for 2FA for the more technical people is SHA1, 8 Digits.

All this means for the user is what applications are supported, list is below:

- Bitwarden/Vaultwarden (Password Manager, Vaultwarden can be selfhosted!, 2FA is setup in addition to account)
- Authy (Recommended application, named as the best for the stack)
- Aegis (Supported, never used. YMMV)

These three options are supported for 2FA for Authelia. It is recommended to actually set this up, as it prevents complete lockouts.

Duo is another way to configure 2FA. If you enable Duo 2FA, you'll need to consult the documentation for Duo setup via Authelia, I've linked some setup links here.

https://www.authelia.com/configuration/second-factor/duo/

https://www.authelia.com/overview/authentication/push-notification/

You'll need to modify the configuration of Authelia to finish the setup of 2FA via Duo.

One File Docker Compose

```
version: "2.1"
services:
  swag:
    image: lscr.io/linuxserver/swag:latest
    container_name: swag
    cap_add:
      - NET_ADMIN
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Phoenix
      - URL=[sakura.nt]
      - VALIDATION=dns
      - SUBDOMAINS=wildcard
      - DNSPLUGIN=cloudflare
      - EMAIL=[sakurant@gmail.com]
      - ONLY_SUBDOMAINS=false
      #- EXTRA_DOMAINS= #optional
      - STAGING=false
      - DOCKER_MODS=linuxserver/mods:universal-docker|linuxserver/mods:swag-auto-
reload|linuxserver/mods:swag-dashboard|linuxserver/mods:swag-dbip
    volumes:
      - /home/[INSERT USER FOLDER HERE]/appdata/swag:/config
    ports:
      - 443:443
      - 80:80
    networks:
      - [INSERT NETWORK NAME HERE]
    restart: unless-stopped

  authelia:
    container_name: authelia
    image: authelia/authelia:latest
```

```
  expose:
    - 9091
  volumes:
    - /home/[INSERT USER FOLDER HERE]/appdata/authelia:/config
  networks:
    - [INSERT NETWORK NAME HERE]
  restart: unless-stopped
  depends_on:
    - redis
    - mariadb

lldap:
  container_name: lldap
  image: lldap/lldap:stable
  ports:
    # For LDAP
    - "3890:3890"
    # For the web front-end
    - "17170:17170"
  volumes:
    - /home/[INSERT USER FOLDER HERE]/appdata/lldap:/data
  networks:
    - [INSERT NETWORK NAME HERE]
  environment:
    - TZ=America/Phoenix
    - LLDAP_JWT_SECRET=REPLACE_WITH_RANDOM
    - LLDAP_LDAP_USER_PASS=REPLACE_WITH_PASSWORD
    - LLDAP_LDAP_BASE_DN=dc=sakura,dc=nt

redis:
  container_name: redis
  image: bitnami/redis:latest
  expose:
    - 6379
  volumes:
    - /home/[INSERT USER FOLDER HERE]/appdata/redis:/bitnami/
  environment:
    REDIS_PASSWORD: "YOUR_REDIS_PASSWORD"
  networks:
    - [INSERT NETWORK NAME HERE]
```

```
    restart: unless-stopped

  mariadb:
    container_name: mariadb
    image: linuxserver/mariadb:latest
    expose:
      - 3306
    volumes:
      - /home/[INSERT USER FOLDER HERE]/appdata/mariadb:/config
    environment:
      MYSQL_ROOT_PASSWORD: "YOUR_MYSQL_ROOT_PASSWORD"
      MYSQL_ROOT_USER: root
      MYSQL_DATABASE: authelia
      MYSQL_USER: authelia
      MYSQL_PASSWORD: "YOUR_MYSQL_USER_PASSWORD"
    networks:
      - [INSERT NETWORK NAME HERE]
    restart: unless-stopped

  networks:
    [INSERT NETWORK NAME HERE]:
      driver: bridge
      external: true
```

This is the Docker Compose file used for the VM Template, and can be used as a custom template in Portainer.

The custom network listed under "Networks" is used for all the applications that you want to connect through the stack.

Please replace the ALL_CAPS_PORTIONS and the pieces in square brackets. These are values that you will want to set yourself.

For things like passwords and such, I'd recommend a 26 character random string, using alphanumeric characters in both upper and lower case. These values are crucial that they remain both secret and lengthy. You should not need to change these, and some of them will break the application if you try to change them. I'd recommend that you backup your compose file to the cloud in case you need to redo the setup.

Maintenance of the DP-SALA/D Stack

Suggested maintenance for the DP-SALA/D Stack

Required Maintenance for the DP-SALA/D Stack

Debian 12:

- At least monthly, run a "sudo apt-get update && sudo apt-get update" to update the base applications for Debian, this applies security and feature upgrades for your OS, which is important.
- At least bi-weekly, save all the configuration files and important data to a secure, private, personal location. You could even use the Google Drive space of the GMail address that you setup, and save the previous backups. You could probably keep the last 5 backups just to be safe, but minimum of 3.

Portainer:

- Update whenever there is an update. If you login and it tells you there is an update available, update right then and there. Portainer is just a docker container that lets you manage other docker containers, but make sure to keep this as up to date as possible.
- You generally don't have to keep a backup of the Portainer data volume, unless you've made some crazy modification to it's configuration that was different from the standard install.

SWAG:

- At least monthly, restart the container. This allows it to pull new configurations from the GitHub page, as well as renew your SSL certificates for your webserver.

Authelia:

- At least monthly, restart the container. This allows it to pull updates for the container.
- Make sure that your configuration for this is backed up, since multiple sensitive things are within this file, and recovering them will be impossible.

LLDAP:

- This is probably the lowest maintenance thing out of the stack. Just make sure that it is backed up completely.

Authy/Duo:

- Set it and forget it. This is typically included in Authelia, so another reason to make sure that config file is completely backed up properly.

How do I [Insert Question Here]?

Quick Q&A for common problems/questions about the DP-SALA/D Stack

How do I add a new application to my stack?

How do I add LLDAP Authentication to [SERVICE]?

First, check to see if LDAP authentication is supported by the application that you are trying to add. Some services/applications either don't support LDAP auth out of the box, or there is some plugin/addon to add for LDAP auth to function. Do a quick Google search to see if LDAP auth is supported, and if it requires a secure connection. Canvas in particular requires LDAPS (LDAP over SSL) and will not function unless configured correctly.

Second, if the app supports LDAP Auth, it usually requires a dedicated user to manage the application, usually called a "Bind User" which you will setup in LLDAP. Try to pick a username that will be descriptive for the service it will bind to. The password that is used should be lengthy and mixtures of characters. The next required parts are the "configuration" parts, which are detailed below:

- User DN/Bind User/User: "cn=[USER],ou=people,dc=[DOMAIN],dc=[TLD]"
- Base/User Search Base: "ou=people,dc=[DOMAIN],dc=[TLD]"
- User Filter(no grouping, UID access): (&(uid={username})(objectClass=person)(ou=people,dc=[DOMAIN],dc=[TLD]))
- User Filter(group access, UID or Email access): (&(objectClass=person)(memberof=cn=[GROUP NAME],ou=groups,dc=[DOMAIN],dc=[TLD])(|(uid={username})(mail={username})))
- User Filter(group access, UID access): (&(objectClass=person)(memberof=cn=[GROUP NAME],ou=groups,dc=[DOMAIN],dc=[TLD]))
- Admin Filter(group access): (memberof=cn=[GROUP NAME]_administrator,ou=groups,dc=[DOMAIN],dc=[TLD])

These are some typical settings, however, depending on the service or app, there may be an example configuration out there, for example, Authelia for LLDAP has an example config:

https://github.com/lldap/lldap/blob/main/example_configs/authelia_config.yml

Here is a list of the attributes that you can use:

uid (username), mail (email address), cn (this could be a username, or a full name, be careful), and displayName (usually the full name of the user).

Not all of these are good for use cases, however, the most commonly used ones are going to be uid and email.

If this didn't cover what was needed for adding LDAP auth connectivity to your service, consult Google, or the LLDAP GitHub Example Configs, linked here:

https://github.com/lldap/lldap/tree/main/example_configs which should give you a good starting point to see how to configure the LDAP authentication for your service.

Here is my recommendation to how user management works:

1. Setup dedicated groups that match the application/service name, add all users who use that to the group.
2. Filter the user based on the group, if they are in that group, they have permission to access the application.

How do I handle user creation?

Once the stack is all setup, you may want to add users. I outline my way of handling user creation and management, but you may do this anyway you like, I've done it this way given the constraints of how I host, and handle having users access some or all of my services that I host.

User Creation:

1. Ask for the user's email address, full name, and if they would like something different for their display name.
2. Using their full name, I create a username. Say if the user was named: Alice Example, their username would be aexa. If I already had a duplicate of the username, I'd append a number, starting at 0.
3. I would create a temporary password for the user, and then have them go to auth.domain.tld to then "Forgot Password" so that they can setup a password themselves, and have it conform to my password standards I have set.
4. I then go to the user page, and add them to their respective groups to let them have access to the services.
5. I'd let the user know that some apps will take a bit to pull their information, while some create their profile on sign in. This is completely normal, and give it up to 24 hours for changes to fully propagate through the system.

This is one way of doing it, but I do have a standard, which I hold to the highest degree. This is important, as it does make finding the user just a little easier when trying to figure out what goes wrong.