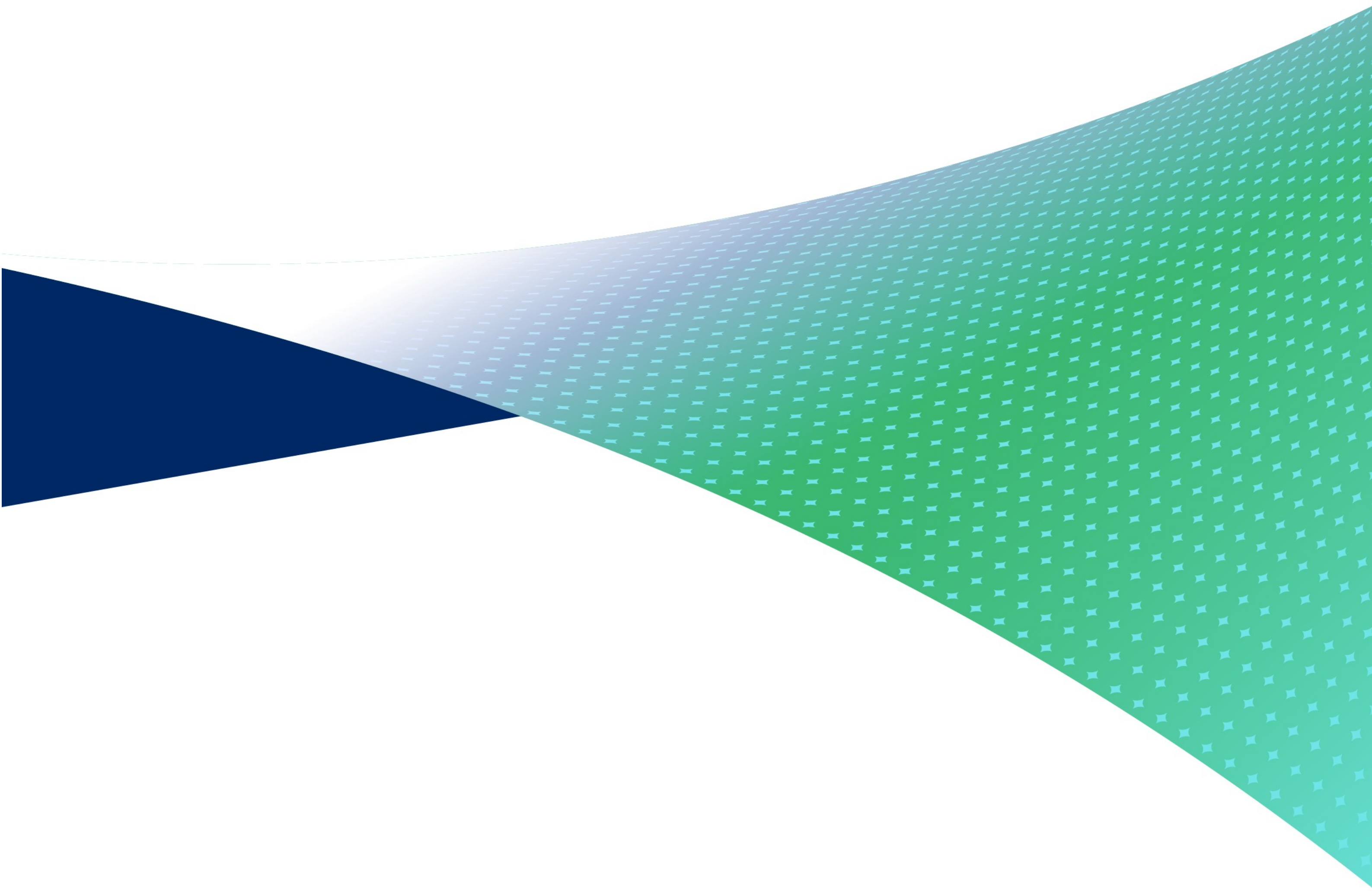


ABBYY® Timeline 5.3

Deployment Guide on Linux Systems

© 2022 ABBYY Development Inc.

© 2022 ABBYY. All rights reserved.

ABBYY® Timeline 5.3 © 2022 ABBYY Development Inc.

ABBYY is either registered trademark or trademark of ABBYY Software Ltd. in the United States and/or other countries.

All other product names, trademarks and registered trademarks are the property of their respective owners.

Information in this document is subject to change without notice and does not bear any commitment on the part of ABBYY.

The software described in this document is supplied under a license agreement. The software may only be used or copied in strict accordance with the terms of the agreement. It is a breach of the United States copyright law and international laws to copy the software onto any medium unless specifically allowed in the license agreement or nondisclosure agreements.

No part of this document may be reproduced or transmitted in any from or by any means, electronic or other, for any purpose, without the express written permission of ABBYY.

Table of Contents

About This Document 4

Introducing ABBYY Timeline 5

System Requirements and Prerequisites 6

Installation, Removal and Upgrade 8

Installing Timeline 8

Performing Health Check 14

Upgrading from Previous Versions 14

Uninstalling Timeline 16

Timeline 5.3.0 patch installation 16

Administering 18

Configuring Timeline Using Environment File 18

HTTPS Configuration with SSL 24

Network Connection Settings 26

Background Upload of Zipped CSV Files to Timeline 27

Log Rotation 27

Known issues 30

Technical Support 34

© 2022 ABBYY Development Inc.

About This Document

This deployment guide is intended for system administrators and engineers. It includes instructions for installation and configuration of ABBYY Timeline 5.3 on Linux.

© 2022 ABBYY Development Inc.

Introducing ABBYY Timeline

ABBYY Timeline is a process intelligence platform comprising over 25 process analysis tools. More than a mere process tracker, ABBYY Timeline will monitor all events within your company in real time and build a detailed map of each process, all while maintaining the flow of your business.

ABBYY Timeline employs an exciting new patent-pending approach to process intelligence called Timeline Analysis which allows users to load events from a variety of systems and in different formats which it then automatically organizes into its corresponding process instances and allows them to be analyzed with a variety of visualization, discovery and query techniques. ABBYY Timeline accepts event data from any number of systems of record and automatically reconstructs the underlying business process logic behind the data. A variety of pre-built analyses are ready to quantify your process performance, identify your process execution issues and perform root cause analysis. The ABBYY Timeline platform also supports operational monitoring through its continuous assessment of new event data to determine if any adverse conditions occur and can immediately notify you or other business operations personnel so you can act.

The ABBYY Timeline engine consumes data from a variety of sources to detect and present detailed views of your business processes. This is often the same exact data being used today for other simpler analyses. This new insight is delivered via a variety of new process and timeline visualization tools developed to not only make these new insights easier to understand but also to allow users to manipulate the information to gain a deeper understanding of those processes. Users armed with this insight have concrete facts on which to take actions to improve operational efficiency by promoting clearly superior best practices and eliminating costly inefficiencies that previously went undetected.

Using advanced algorithms, ABBYY Timeline extracts and reads the time stamps used to record specific events along your processes. The software then visually models these time stamps in such a way that you can instantly identify deviations from an ideal process flow – to find the root cause of a problem that may be costing your business money.

ABBYY Timeline is aimed for use by anyone involved in business process improvements of any type of scale or nature.

System Requirements and Prerequisites

Operating system	Red Hat Enterprise Linux 7.7 Red Hat Enterprise Linux 8.5
CPU	4 cores or more
RAM	16 GB or more
HDD	512 GB or more Depends on the actual amount of data loaded into the application. Production environment may require more disk space, depending on the actual volume of data loaded into the application.
Browser (to access the ABBYY Timeline website)	• Google Chrome 100 or later • Microsoft Edge 100 or later
Additional software	**Included in the installer:** • Redis 6.2.5 • NodeJS 14.17 • Python 3.8.10 • Nginx 1.20.1 **Downloaded from the Internet:** **Important.** The installer automatically downloads and installs the following additional software from the Internet. If your machine is not connected to the Internet, the program will ask you to download it manually and prompt sources. • PostgreSQL 12 Only needed if a local database usage is planned. It will not be installed on the host machine if a remote database is chosen during installation.

© 2022 ABBYY Development Inc.

	• container-selinux 2.119.2 • Docker v.19.03.9 for Red Hat Enterprise Linux 7 v.20.10.9 for Red Hat Enterprise Linux 8 • Docker Compose 1.27.4 • Docker container 18.06.0-ce (with **minideb** docker-image) **Needs to be installed and configured before Timeline installation:** • SMTP Server ABBYY Timeline needs access to a running SMTP server to be able to send verification emails, notifications, invitations, and alerts, etc.
Other requirements	The target machine has to be connected to the Internet during the installation. If it is offline, you will have to download some additional software manually, in accordance with the installer prompts.

Scaling guidelines

The exact calculation of necessary hardware requires multiple parameters such as data volume and use patterns. However, the general guidelines could be defined as following:

- If the number of concurrent users is less than 10 and the data update frequency is one per day or less, a single server should be sufficient.
- For more users or more frequent data updates, a separate server for DBMS is recommended.
- For the fault-tolerant environment, use two identical servers and any standard load balancer.

Installation, Removal and Upgrade

This section provides instructions for installing ABBYY Timeline, updating, and removing it from your computers.

Installing Timeline

Before you begin

- Obtain the Timeline installer for Linux, from either ABBYY representative, or Timeline support at support@abbyy.com .
- Verify all Linux system requirements and prerequisites 6 before starting your Timeline installation.
- Ensure that any firewall installations are not blocking ports 80, 443 and 5432 or the ports you plan to setup for the Web Server and database. The installation will not work if firewall is blocking the ports that have been specified during the Timeline installation.
- If you intend to configure HTTPS, please visit the HTTPS Configuration 24 section.

Procedure

1. Copy the Timeline installer to the local disk, typically to /tmp.
2. Open the shell.
   Important. You need root access to install Timeline. Without root, you won't have the necessary permissions to install it. Prefixing every command with **sudo** is cumbersome and it causes a problem with exported variables on the command line. If you are not using the host machine with the root user, start a new shell with root privileges using the command **sudo bash**.
3. Set the permissions of the **timeline-install-5.3.*.sh** file so that it is executable:
   ```
   chmod +x /path/to/timeline-install-5.3.*.sh
   ```
4. Execute the Timeline install script:
   ```
   path/to/timeline-install-5.3.*.sh
   ```

If the script is in the current directory, then you need to specify the dot (./) before the script file name:
./timeline-install-5.3.*.sh
Continue the installation when prompted. Installation workflow includes the following steps.

5. **Read and accept the license agreement**
   Read the information in the End-User License Agreement. After reviewing the license agreement information, to indicate acceptance of the EULA, press the **Y** key. Any other input cancels the installation. After you accept the license agreement, installation proceeds.

6. **Check and install prerequisites**

   a. **PostgreSQL instance and database**
      Timeline needs access to PostgreSQL 12. You can install PostgreSQL on a computer along with Timeline or a separate computer.
      The program will ask you about using either local or remote database.
      If you already installed PostgreSQL on another computer, select **Remote**. This option is useful if you install the program in a production environment where the host machine is accessible from outside the corporate network. If you prefer to keep Timeline and the database on the same machine, select **Local.**

      i. **Local**
         If you want to install PostgreSQL on the host machine or already installed it there, select **Local**. This option is useful if you install the program for testing purposes or environments where the host machine is not accessible from outside the corporate network, and only the HTTP/HTTPS ports open.

         You can install PostgreSQL using the native package manager on your system or let the Timeline installer install it. During the installation process, the following PostgreSQL databases are created:
         timeline
         The database contains all information about users, their activity, and projects.

         timeline-log
         The database contains detailed records of Timeline events such as security, errors, and notifications.

         timeline-000
         The database contains information about user repositories.

ii. **Remote**
When you select this option, the program will ask you about establishing a secure SSL connection to the remote PostgreSQL database. Press **Y** if you plan to use SSL, or **N** to reject.
Important. If your PostgreSQL is configured with SSL support and a root CA certificate file is used, you must provide the full path to the root CA certificate when configuring your connection settings. The certificate file will be copied to the **$TIMELINE_INSTALLATION_DIR/db-ssl** folder.
To use this option, you must prepare **timeline**, **timeline-log**, and **timeline-000** databases in the remote PostgreSQL in advance:

1. Launch PostgreSQL

2. Create a user that can own database objects.
   For example, TimelineUser.

3. Create the following databases owned by the user you created in the previous step:
   timeline
   The database contains all information about users, their activity, and projects.

   timeline-log
   The database contains detailed records of Timeline events such as security, errors, and notifications.

   timeline-000
   The database contains information about user repositories.

   For setting up access to Timeline databases, you will be asked for the connection settings in one of the further installation steps.

b. **Docker and docker-compose**
Timeline runs in Docker containers so Docker and Docker-compose should be installed on the host machine. Docker is a Linux-based virtualization tool that helps to make complex applications more portable. You can install it manually or let the Timeline installer download and install it.
Important. In Timeline 5.3, the installer automatically downloads Docker and its dependencies from the Internet. If your machine is not connected to the Internet, the program will ask you to download it manually and prompt sources.

7. **Configure web server**

   a. **HTTP and HTTPS port**
   Specify the TCP/IP port for the Timeline website. Make sure that the specified port is not being used by any other application. By default, the application listens on port 80 for HTTP and port 443 for HTTPS. If both ports are defined, HTTP requests will be redirected to HTTPS. You can change ports configuration after installation. For more information, see Configuring ABBYY Timeline Using .env File 18 section.

      i. **HTTP port (0 to disable) (80)**
      Press **Enter** to use the default 80 port or enter the port number.

      ii. **HTTPS port (0 to disable) (443)**
      Press **Enter** to use the default 443 port or enter the port number. For setting up HTTPS, see the HTTPS Configuration with SSL 24 section.

   b. **Base URL**
   Enter the base URL that hosts Timeline or press **Enter** to use the default **https://127.0.0.1** one. It must be a public IP of the server or an external fully qualified URL. The lowercase pattern is recommended. Base URL also is used for links inside email messages sent by Timeline.

   The Base URL must have the following syntax: **http[s]://hostname:port**
   Important. Do not use extra spaces and forward slash '/' at the end of the base URL.

   If you are using the default port (80 or 443), you do not need to add them to the base URL.

   Examples:
   The base URL of the HTTP endpoint, if a custom port is specified:
   http://mytimeline.com:8080

   The base URL of the HTTPS endpoint, if a custom port is specified:
   https://mytimeline.com:30443

8. **Configure mail server**

   Configure the SMTP server access to allow Timeline sending out emails in several features such as Alerting, User invitation, etc. You should provide general information to configure SMTP mail server and specify its security options. To decide which options you have to select, please refer to the documentation of your mail server. The server basic settings are set during installation. You can change SMTP Mail Server configuration after installation. For more information, see Configuring ABBYY Timeline Using .env File 18 section.

   a. **Host**

      Specify server name where SMTP mail server is installed.

   b. **Port**

      Enter SMTP mail server port number.

   c. **Username** and **Password**

      Enter SMTP mail server access credentials.

   d. **E-Mail sender**

      Enter the sender address that is used to fill the 'From' header field of e-mails.

   e. **Use TLS? (Y/N)**

      Enter **N** in case your SMTP server does not use TLS. It is a typical use case for mock, local mail services, for example, mailcatcher.

   f. **Require TLS (Y/N)**

      Enter **Y** if the initial connection should happen over an unencrypted connection and then the STARTTLS command should be used to upgrade to a secure connection. For example, Microsoft Exchange.

   g. **Reject unauthorized (Y/N)**

      Enter **Y** if your mail server uses SSL certificate issued by the Certification Authority (CA).

      Enter **N** if your mail server uses an unauthorized, e.g., self-signed, SSL certificate.

9. **Configure admin user account**
   Enter a valid e-mail address using an existing domain name that is configured to receive emails, for example, **user@domain.com**, and a password for it. This will be the first user of Timeline and the one that will have access to the ABBYY Timeline website, where other users can be administered.
   The password you specified must contain only English letters and digits from 0 to 9.
   Note. If you are upgrading Timeline, you are not prompted to enter admin user credentials because previous settings are maintained.

10. **Database**
    This step appears if you choose Remote to connect to a remote PostgreSQL instance at the beginning of the installation process. The installer will ask for the settings for accessing the **Admin DB**, **Log DB**, **User DB** databases. For each of these databases, specify the connection parameters to **timeline**, **timeline-log**, and **timeline-000** databases located on the remote PostgreSQL.

    a. **PostgreSQL host**
       If you select **Connect to existing database** on the previous step, specify server name where PostgreSQL is installed. By default, **localhost** is used.

    b. **PostgreSQL port**
       Specify the TCP/IP port for PostgreSQL. By default, TCP/IP port **5432** is used. Make sure that it is not being used by any other application.

    c. **Database username**
       Provide the credentials of the PostgreSQL user who owns the Timeline databases. For example, TimelineUser.

    d. **Database name**
       - Enter **timeline** as the database name for the **Admin DB**.
       - Enter **timeline-log** as the database name for the **Log DB**.
       - Enter **timeline-** as the database name for the **User DB**. It is the prefix of the **timeline-000** database you created in the remote PostgreSQL.

11. Perform Timeline health check[14]

During the installation process, Firewall exceptions are created, allowing interactions between components to take place inside a network. For default network connection settings see Network connection settings[26] section.

Important. If you intend to configure HTTPS, you must set up SSL configuration after installation is complete. See HTTPS Configuration with SSL[24] for details.

Performing Health Check

Check that Timeline is working properly by doing the following:

1. Make sure all docker containers are running on the host machine by using:
   sudo docker ps -a
   Note. You can ignore the status of **timeline_migrate_1** container which is used only to migrate databases and is not running after Timeline start.

2. Open a browser and enter **{timelineURL}:{port}** in the address bar, where:
   {timelineUrl} is the Base URL you specified during the Timeline installation or the public IP address or the full name of the computer on which Timeline is installed.
   {port} is the custom port assigned to the Timeline website during the installation process.
   If you are using the default port (80 or 443), you do not need to add them to the {**timelineURL}**. By default, TCP/IP port 80 or 443 is used.

   Example: http://mytimeline:8080 or https://mytimeline:30443

3. If the installation was carried out correctly, the Timeline website will open.

4. Login using the Timeline admin credentials you specified during the installation process 13.

Upgrading from Previous Versions

If you own ABBYY Timeline 5.1.2 or later, you can upgrade to ABBYY Timeline 5.3. This means that you can install a new version of ABBYY Timeline and your databases and previous settings will be maintained.

Note. Upgrade is only possible within Red Hat Enterprise Linux 7 versions. Under Red Hat Enterprise Linux 8 clear installation is required since previous versions do not support this operating system. However, it is possible to connect a remote database used in the previous Timeline version by connecting to it in the **Connect to Database** step.

Procedure

To update your Timeline installation to the latest version, you have to execute the following steps:

1. If you are using a local database, backup it and the **Storage** folder in the installation directory.
   For detailed information, see https://www.postgresql.org/docs/12/backup.html.
2. Run the Timeline installation as described in the Installing ABBYY Timeline 8 section and follow the instructions of the Installation Wizard. Press **Y** when the program will prompt you to upgrade.
3. When prompted **Import existing Timeline installation** select **opt/Timeline** or **custom path** depending on where the program is already installed.
4. Select PostgreSQL location when prompted **Do you want ABBYY Timeline to use a local PostgreSQL instance or connect to a remote database?**

   ○ When upgrading within **Red Hat Enterprise Linux 7** database settings from any location are maintained.

   ○ **Red Hat Enterprise Linux 8** requires a clean installation. To work with existing data you need to have a configured remote database from the previous Timeline version and connect to it in the **Connect to Database** step. If you choose the local instance option - a new database will be created.

5. If you plan to configure HTTPS with SSL specify HTTPS port when prompted.

6. To use existing databases, specify the values for connecting to them in the **Connect to Database** step.
   Note. In case of local database, please make sure that the correct values are entered. Incorrect user credentials (e.g. a typo in username) will cause a new database creation.

7. Complete the Timeline installation.

8. After the Timeline installation process is complete, configure HTTPS with SSL if necessary. During the installation process the **ssl.conf.tpl** file is copied to the **$TIMELINE_INSTALLATION_DIR/nginx** folder. This file stores SSL configuration settings. To setup SSL go to the **$TIMELINE_INSTALLATION_DIR/nginx** folder and do one of the following:

   a. Rename the **ssl.conf.tpl** file to **ssl.conf**.

   b. Alternatively, merge the **ssl.conf.tpl** file with **ssl.conf**
      Use the merge strategy if you made any changes to the **ssl.conf** file for a previous version of Timeline.

9. Perform Timeline health check 14.

Uninstalling Timeline

To remove Timeline, execute the following commands, as root, to remove the docker containers and images, and other files. If you are not using the host machine with the root user, start a new shell with root privileges using the command **sudo bash**.

```
1. user@host:~# docker kill $(docker ps -q)

2. user@host:~# docker rm $(docker ps -aq)

3. user@host:~# docker rmi $(docker images -aq)

4. user@host:~# docker network rm timeline_network

5. user@host:~# rm /etc/systemd/system/timeline.service

6. user@host:~# rm -rf /opt/timeline
```

Timeline 5.3.0 patch installation

If you are using the Timeline 5.3.0 version and upload from an ODBC data source fails, you need to download the patch-file. Click here to download.
This issue is fixed in later versions.

Prerequisites

- If the target machine (HOST) is not connected to the Internet, you will need an additional machine that has internet connection and docker installed (BUILDER). You need to have the **Dockerfile.odbcpatch** on the respective machine (HOST or BUILDER) depending on the HOST internet connection. All actions in the instruction are marked on which machine to perform them.
- Make sure you have root access on both machines.

Step 1 - Setup

1. HOST: Install the Timeline application.
2. HOST: Stop the timeline service:
   ```
   systemctl stop timeline
   ```

Step 2 - Patch

1. Patch with internet connection on HOST.

   a. HOST: Patch the **timeline/backend** image with the received dockerfile:

      ```
      cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t
      timeline/backend:latest .
      ```

2. Patch without internet connection on HOST.

   a. HOST: Export the **timeline/backend** image:

      ```
      docker save -o timeline-backend-latest.tar timeline/backend:latest
      ```

   b. Move the exported **timeline-backend-latest.tar** to the BUILDER machine.

   c. BUILDER: Load the **timeline/backend** image:

      ```
      docker load --input <path_to_image_tar>/timeline-backend-latest.tar
      ```

   d. BUILDER: Patch the **timeline/backend** image with the received **dockerfile**:

      ```
      cd <path_to_dockerfile> && docker build -f Dockerfile.odbcpatch -t
      timeline/backend:latest .
      ```

   e. BUILDER: Export the patched **timeline/backend** image:

      ```
      docker save -o timeline-backend-latest-patched.tar timeline/backend:latest
      ```

   f. Move the exported **timeline-backend-latest-patched.tar** to the HOST machine

   g. HOST: Load the patched **timeline/backend** image:

      ```
      docker load --input <path_to_image_tar>/timeline-backend-latest-patched.tar
      ```

Step 3 - Clean-up and start Timeline service

Perform these actions on the machine you have used for patch installation.

1. (Optional) HOST/BUILDER: Remove the exported image files

2. (Optional) HOST/BUILDER: Find the old **timeline/backend** image (repository: timeline/backend, tag: <none>):

   ```
   docker image ls
   ```

3. (Optional) HOST/BUILDER: Delete the old **timeline/backend** image by id:

   ```
   docker image rm <image_id_of_old_backend_image>
   ```

4. HOST: Restart timeline service:

   ```
   systemctl start timeline
   ```

Administering

Chapter contents

- Configuring Timeline Using Environment File 18
- HTTPS Configuration with SSL 24
- Network Connection Settings 26
- Background Upload of Zipped CSV Files to Timeline 27
- Log Rotation 27

Configuring Timeline Using Environment File

The Timeline settings may be configured after the installation.

At the system hosting Timeline, open the **opt/timeline/.env** file in any editor and set the following environment variables:

Parameters	Information	
Database connection settings		
ADMIN_DATABASE_URL LOG_DATABASE_URL USER_DATABASE_URL_PREFIX	Description	Configures access to **timeline**, **timeline-log** and **timeline-000** databases.
	Format	• **timeline** and **timeline-log** database URLs have the following format: `postgres://<username>:<password>@<IP address or postgres hostname>:<Port>/<Database name: timeline or timeline-log>` • **timeline-000** database has a similar format with the difference that the last part defining the database should not contain the number '000': `postgres://<username>:<password>@<IP address or postgres hostname>:<Port>/<Database name prefix: timeline->`

© 2022 ABBYY Development Inc.

Parameters	**Information**	
		<IP address or postgres hostname> should be the machine's IP or public name so it can be accessed from Docker containers. By default, PostgreSQL uses port **5432**. Make sure that the configured port is not being used by any other application and is open on the firewall. If using the default PostgreSQL port, it can be done by: **firewall-cmd --add-service=postgresql** and **firewall-cmd --runtime-to-permanent**
	Example	*ADMIN_DATABASE_URL=postgres://trx:x@172.18.0.1:5432/timeline* *LOG_DATABASE_URL=postgres://trx:x@172.18.0.1:5432/timeline-log* *USER_DATABASE_URL_PREFIX=postgres://trx:x@172.18.0.1:5432/timeline-*
Web server configuration		
PROXY_PORT PROXY_SSL_PORT	Description	Specifies the ports configuration available for the application on the host machine. By default, the application listens on port 80 for HTTP and port 443 for HTTPS. If both ports are defined, HTTP requests will be redirected to HTTPS. For details on SSL configuration, see section 'HTTPS configuration 24'. **Important**.

Parameters	Information	
		• Make sure that the configured ports are open on the firewall and not being used by any other application. • If you install the application in a production environment, it is strongly recommended to use HTTPS and highly discouraged HTTP.
	Format	*PROXY_PORT=<HTTP port>* *PROXY_SSL_PORT=<HTTPS port>* 0 (zero) means the port is disabled.
	Example	*PROXY_PORT=0* *PROXY_SSL_PORT=443*
BASE_URL	Description	Specifies the Base URL that hosts Timeline. The hostname should include the port number if it is not the default and the protocol (http/https) of the server where the application is going to run. The BASE_URL variable is used for links inside email messages sent by Timeline.
	Format	*BASE_URL={protocol}://hostname[:port]*
	Example	*BASE_URL=http://10.15.61.165* (if use HTTP) *BASE_URL=https://mytimeline.com* (if use HTTPS)
Mail server configuration		
MAIL_SERVER_HOST MAIL_SERVER_PORT MAIL_SERVER_USERNAME	Description	Specifies SMTP server access to allow Timeline sending out emails in several features such as Alerting, User invitation, etc.

© 2022 ABBYY Development Inc.

Parameters	Information	
MAIL_SERVER_PASSWORD MAIL_SERVER_TLS_CONNECTION MAIL_SERVER_REQUIRE_TLS MAIL_SERVER_REJECT_UNAUTHORIZED EMAIL_SENDER		Provide SMTP mail server access credentials such as host, port, username, password, e-mail sender address, and mail server security options. • **MAIL_SERVER_USERNAME** **MAIL_SERVER_PASSWORD** Keep these fields empty if the mail server requires no authentication. • **MAIL_SERVER_TLS_CONNECTION** **MAIL_SERVER_TLS_CONNECTION=true** makes the app connect to the mail server using TLS right from the start. This is the most secure option. Unfortunately, not all mail servers support this. E.g., Exchange requires unencrypted connection, and then use the STARTTLS command to upgrade. In this case, use: **MAIL_SERVER_TLS_CONNECTION=false** and **MAIL_SERVER_REQUIRE_TLS=true**. • **MAIL_SERVER_REQUIRE_TLS** To enable/disable TLS set **MAIL_SERVER_REQUIRE_TLS** to **true/false**. • **MAIL_SERVER_REJECT_UNAUTHORIZED** Set **MAIL_SERVER_REJECT_UNAUTHORIZED** to **false** if your mail server uses a self-signed certificate. Default value is true. • **EMAIL_SENDER** **EMAIL_SENDER** is used to fill the 'From' header field of e-mails.
	Format	*MAIL_SERVER_HOST=<mail server IP address or hostname>*

Parameters	Information	
		MAIL_SERVER_PORT=<mail server port> *MAIL_SERVER_USERNAME=<mail server username>* *MAIL_SERVER_PASSWORD=<mail server password>* *MAIL_SERVER_TLS_CONNECTION=<true/false>* *MAIL_SERVER_REQUIRE_TLS=<true/false>* *MAIL_SERVER_REJECT_UNAUTHORIZED=<true/false>* *EMAIL_SENDER=<mail sender e-mail>*
	Example	*MAIL_SERVER_HOST=example.smtp.server.com* *MAIL_SERVER_PORT=465* *MAIL_SERVER_USERNAME=mail_user* *MAIL_SERVER_PASSWORD=mail_password* *MAIL_SERVER_TLS_CONNECTION=false* *MAIL_SERVER_REQUIRE_TLS=true* *MAIL_SERVER_REJECT_UNAUTHORIZED=false* *EMAIL_SENDER=timeline-support@example.com*
Timeline folders		
LOGS NGINX_CONF DB_SSL PG_SSL_ROOT_CERT	Description	Specifies the locations of directories the app saves data to. Each of these should be directories on the host machine. If you specify relative paths, they will be relative to the installation directory.

Parameters	Information	
STORAGE LICENSE		- **LOGS** All Timeline logs will be placed here. Default value: **/opt/timeline/logs** - **NGINX_CONF** This is a directory for SSL configuration and certificates. Default value: **/opt/timeline/nginx** For details on SSL configuration, see section 'HTTPS configuration 24'. - **DB_SSL** This is a directory for a database certificate file. If your remote PostgreSQL is configured with SSL support and a CA Root certificate file is not presented, the certificate file must be copied to the host machine into this directory. Default value: **/opt/timeline/db-ssl** - **PG_SSL_ROOT_CERT** This is a name of the database CA Root certificate file located in the DB_SSL folder. If your remote PostgreSQL is configured with SSL support and a CA Root certificate file is used, this root certificate file must be specified in this key. - **STORAGE** This directory is used by different parts of the application to permanently or temporarily store data. Make sure that the directories are not world readable and that they are backed up regularly. Default value: **/opt/timeline/storage**

Parameters	Information	
		• **LICENSE** The path to the directory where the license file is located relative to the installation directory. Default value: **/opt/timeline/license** By default, all directories are under the installation directory.
	Example	*LOGS=/opt/timeline/logs* *NGINX_CONF=/opt/timeline/nginx* *STORAGE_DIR=/opt/timeline/storage* *LICENSE=/opt/timeline/license*

HTTPS Configuration with SSL

The application uses NGINX proxy to deliver HTTP requests from the browsers to the backend services. This proxy is responsible for SSL termination too.

To configure HTTPS, you need SSL certificates for Timeline. You can choose one of the following options:

1. Use SSL certificate issued by the Certification Authority (CA).
   This is the recommended approach for the application installation that is intended for a production environment. The connection to the server will be secure and users will not get any warnings from the browser.

2. Use a self-signed SSL certificate.
   If you do not have a signed certificate or if you only require a certificate for testing purposes, use a self-signed SSL certificate. However, in this case users will get warnings from the web browser about the use of a self-signed certificate as the server will not be considered secure.
   Note. If you install the program in a production environment, it is highly discouraged to use a self-signed SSL certificate.

Important. If you install the program in a production environment, it is strongly recommended to use HTTPS and highly discouraged HTTP.

Procedure

1. Obtain an SSL certificate and a private key.
2. Run the Timeline installation and follow the Installation Wizard.
   For more information see 'Installing Timeline 8'.
   a. To enable SSL between instances of PostgreSQL database and application provide path to your database SSL certificate in the **Database Connection** step. If your PostgreSQL is configured using SSL, provide the path to your SSL CA root certificate.
   b. To enable SSL between application and client specify HTTPS port and Base URL for HTTPS port in the `Web Server` step.
3. After the Timeline installation process is complete, do the following:
   a. Find the **ssl.conf.tpl** and **ssl.conf** files in the $TIMELINE_INSTALLATION_DIR/nginx folder and rename the **ssl.conf.tpl** file to **ssl.conf**. Alternatively, merge the **ssl.conf.tpl** file with **ssl.conf**, if you made any changes in the **ssl.conf** file for the previous Timeline version**.**
   Note. These files are copied to the $TIMELINE_INSTALLATION_DIR/nginx folder during the upgrade process. The folder is specified in the NGINX_CONF variable in .env 22. The **ssl.conf.tpl** file stores the latest SSL configuration settings.
   b. Copy your SSL certificate and private key files to the $TIMELINE_INSTALLATION_DIR/nginx folder.
      i. If your private key and certificate files are not named **cert.key** and **cert.pem**, respectively, you should change the **ssl_certificate** and **ssl_certificate_key** entries in **ssl.conf** accordingly.
      ii. If you have a password file for the SSL key, uncomment the line #ssl_password_file $TIMELINE_INSTALLATION_DIR_DIR/nginx/conf/pass.file; in **ssl.conf**. If necessary, change the path to the folder you specified during the installation process.
      iii. If intermediate certificates should be specified in addition to a primary certificate, they should be specified in the same **cert.pem** file in the following order: the primary certificate comes first, then the intermediate certificates.
   c. Open **.env** file and check the following environment variables:

i. PROXY_SSL_PORT
   Make sure the HTTPS port you want to use is specified in the PROXY_SSL_PORT variable.
   Example: PROXY_SSL_PORT=443

ii. BASE_URL
   Make sure the HTTPS protocol is specified in the BASE_URL variable.
   Example: BASE_URL=https://mytimeline.com

iii. DB_SSL
   If your remote PostgreSQL is configured with SSL support without root certificate, make sure that this variable is empty.

iv. DB_SSL=./db-ssl
   When remote PostgreSQL is configured with SSL support and a CA Root certificate file is used, make sure that this variable contains a full path to the certificate file.

v. PG_SSL_ROOT_CERT
   The name of the certificate file copied into the folder specified in the DB_SSL variable.

4. Restart the Timeline application to apply all the changes:
   systemctl restart timeline

5. Perform a health check 14.

Network Connection Settings

The table below lists the ports that are used by default to access Timeline. If you are using a software or hardware firewall, make sure that the exception settings for Timeline have been set up accordingly on the computer where it is installed.

If you reassign port numbers in PROXY_PORT and/or PROXY_SSL_PORT variables in the .env file 19, you will need to make changes to the appropriate firewall rules that you are using.

© 2022 ABBYY Development Inc.

Application name	Protocol type	Port	Traffic direction	Use
Timeline	TCP/IP	80 or the port specified during the installation (if use HTTP) 443 or the port specified during the installation (if use HTTPS)	Inbound	HTTP or HTTPS connections to the Timeline website.
PostgreSQL	TCP/IP	5432	Inbound	Connections to a PostgreSQL database server from the computer where Timeline is hosted.

Background Upload of Zipped CSV Files to Timeline

The background-upload feature involves a folder that is monitored for files copied there. Whenever a new ZIP file is detected in that folder, the application grabs it and interprets it as an uploaded archive. The folder is defined as STORAGE/sftp, where STORAGE is the variable in the .env file 23.

The upload file can be copied to the specified folder by any means. It can be the target of an SFTP upload, or it can be an otherwise shared folder.

Log Rotation

The log file generated by the application can quickly increase in size, and if you want to make sure it doesn't take up too much disk space, you can introduce log rotation. Log rotation will periodically clear the old logs, thus preventing the log file from taking up all the disk space.

© 2022 ABBYY Development Inc.

On most Linux systems, the **logrotate** command is already located at **/usr/sbin/logrotate**. The way to set up log rotation can differ based on the kind of Linux distribution you use, whether you set up the application as root or as a simple user, whether you placed the application in **/opt** or in **/home**, and the exact location where the logs are placed. You will find 2 typical use cases below:

A) When the log files are placed inside /opt/timeline or some similar place, and docker is executed as a root user:

1. Create a log rotation config file, for example at: **/etc/logrotate.d/timeline**
   The file should contain the following:

   ```
   /opt/timeline/logs/* {
       size 1G
       copytruncate
       rotate 1
   }
   ```

   The path should point to the log file generated by the docker-compose up command. This particular configuration would clear the log file when it exceeds the size of 1 MB and copy its original content to another file called **/home/<USER>/timeline/logs/docker-compose.log.1**. The next time the log rotation runs and finds that the log size exceeds 1 MB again, it overrides the **docker-compose.log.1** with its new contents and clears the original log file.

2. This assumes that the Linux system already has a log rotation installed and registered as a cron job. You can verify this by checking that an **/etc/logrotate.conf** file exists, and it contains the line **include /etc/logrotate.d**, and also that there is a file called **/etc/cron.daily/logrotate** that runs the command **/usr/sbin/logrotate /etc/logrotate.conf**. Different Linux distributions might have these files arranged in a different way.

B) When the log files are placed inside /home/<USER>, and are written by a non-root user:

1. Create a log rotation config file, for example at: **/home/<USER>/logrotate.conf**
   The file should contain the following:

   ```
   /home/<USER>/timeline/logs/* {
       size 1G
       copytruncate
       rotate 1
   }
   ```

   The path should point to the log file generated by the docker-compose up command. This configuration works the same way as described in the previous case.

2. Register a cron job to run the log rotation procedure once a day.
   Run the following command to create a user-specific cron job:
   crontab -e
   This will open a text editor where you can register cron jobs by adding lines like the following:

   ```
   0 * * * * /usr/sbin/logrotate /home/$USER/logrotate.conf --
   state /home/$USER/logrotate-state.txt
   ```

   This line would result in the **logrotate** command executed once per day using the previously defined log rotation configuration, and storing its state in **logratate-state.txt** (this can be any file, and doesn't have to exist at the beginning)

Important. For log rotation to work correctly, you have to write the log files in append mode (in bash '>>' instead of just '>'), otherwise the log file cannot be cut, since the process would keep writing at its current offset at the location that used to be the end of the file, even after the file was cleared.

Known issues

Upload from an ODBC data source fails in Timeline 5.3.0

If you are using the Timeline 5.3.0 version and upload from an ODBC data source fails, please download the patch-file to resolve this issue. For details and instructions see Patch installation 16.
This issue is fixed in later versions.

Timeline Application not accessible outside the installed machine under Red Hat Enterprise Linux

This issue only occurs on host machines running **Red Hat Enterprise Linux** as its installation uses docker network and includes the OS built-in firewall.

It can be identified when the application is not reachable from outside the network/machine, but is reachable from inside and the following command returns an HTML response:

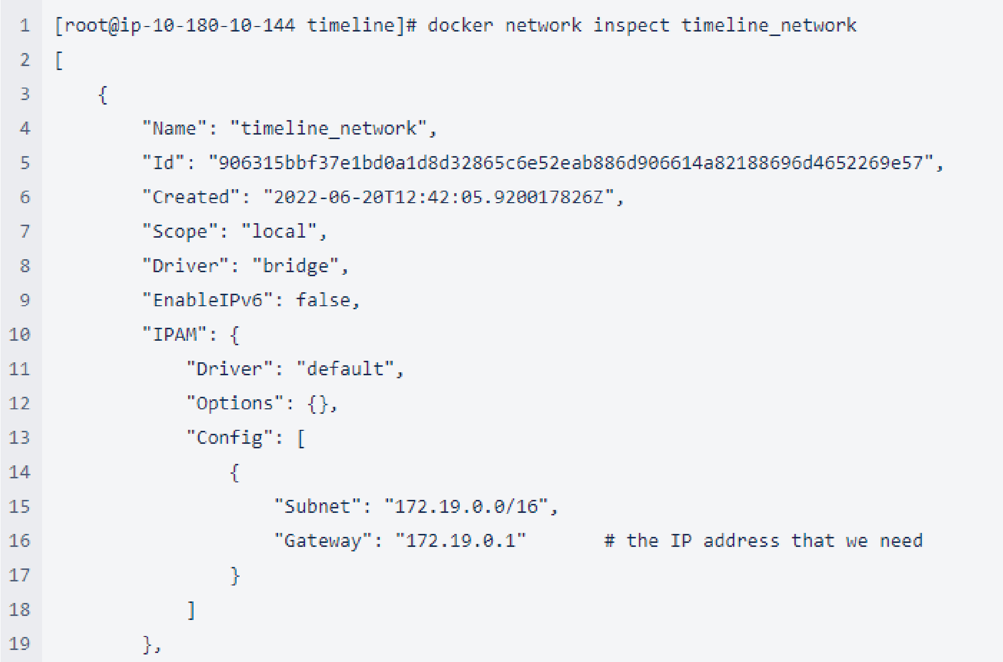
```
curl localhost
```

In this case, please recreate the Docker network:

1. Stop the Timeline service, to make sure it does not try to use Docker:
   ```
   service timeline stop
   ```
2. Remove all containers using the command below, when asked for confirmation accept, select **yes**:
   ```
   docker container prune
   ```
3. List Docker networks:
   ```
   docker network ls
   ```

a. Inspect the Docker network to get its gateway ip:

```
docker network inspect timeline_network
```



```
[root@ip-10-180-10-144 timeline]# docker network inspect timeline_network
[
    {
        "Name": "timeline_network",
        "Id": "906315bbf37e1bd0a1d8d32865c6e52eab886d906614a82188696d4652269e57",
        "Created": "2022-06-20T12:42:05.920017826Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"       # the IP address that we need
                }
            ]
        },
```

4. Delete the network related to Timeline. Use the id of **timeline_network**:

```
docker network rm timeline_network
```

5. Make sure the **timeline_network** got deleted using command from step 3:

```
docker network ls
```

6. Stop the Docker service:

```
service docker stop
```

7. Stop the PostgreSQL service:

```
service postgresql-12 stop
```

Note. Perform this step if you are using a local database. Skip it if you are using a remote database.

8. Temporarily delete all firewall rules using this command:

```
iptables --flush
```

9. Restart the firewall:

```
service firewalld stop
service firewalld start
```

10. Start the Docker service:

```
service docker start
```

11. Once the Docker service is started, create a new Docker network:

```
docker network create timeline_network
```

12. Flush the IP tables one more time after restart:

```
iptables --flush
```

© 2022 ABBYY Development Inc.

13. Inspect the network via Docker, the IP address of the gateway might change:

    ```
    network inspect timeline_network
    ```

14. Copy the gateway IP address from the network and make sure that you have consistent values in the **opt/timeline/.env** file:
    Check the values of the following variables and change them if needed:
    ADMIN_DATABASE_URL
    LOG_DATABASE_URL
    USER_DATABASE_URL_PREFIX

15. Perform the following steps if you are using a local database. Skip these steps if you are using a remote database.

    a. Copy the gateway IP address from the network and make sure that you have consistent values in the following files:

       i. **pg_hba.conf**
          Add an entry at the end to allow the connection:

          ```
          host all all $DOCKER_GATEWAY_IP md5
          ```

          Also add the IP mask. E.g.:

          ```
          host all all 172.10.0.1/24 md5
          ```

          Default path for PostgreSQL 12 conf files: **/var/lib/pgsql/12/data/**

       ii. **postgresql.conf**
          Change the listen address with the value of the new **DOCKER_GATEWAY_IP**.
          Default path for PostgreSQL 12 conf files: **/var/lib/pgsql/12/data/**

    b. Restart the PostgreSQL service:

       ```
       service postgresql-12 start
       ```

       i. If you didn’t stop your local PostgreSQL database service earlier, restart the service for it to use the new configuration files:

          ```
          service postgresql-12 stop
          ```

          then

          ```
          service postgresql-12 start
          ```

16. Make sure that Docker service is running:

    ```
    service docker status
    ```

17. Start the Timeline service:

    ```
    service timeline start
    ```

18. Make sure that all the containers are up and running (it can take some time):

    ```
    docker container ls -a
    ```

© 2022 ABBYY Development Inc.

19. Check that the Timeline application is available. The following command should return an HTML response:

    ```
    curl localhost
    ```

20. Make sure that the Timeline application is available outside, by opening the application on the DNS/hostname of the machine.

© 2022 ABBYY Development Inc.

Technical Support

Should you have any questions regarding the use of ABBYY Timeline, please e-mail the ABBYY technical support service at support@abbyy.com. Please provide the following information when contacting technical support:

- your first and last name.
- the name of your organization.
- your phone number (or fax, or e-mail).
- your ABBYY Timeline version and the build number.
- a description of the problem and the full text of the error message (if there was any).

© 2022 ABBYY Development Inc.