MYWAVE.ai

MyWave — Local Hosting Guide

A DEVELOPER'S GUIDE FOR LOCALLY HOSTING MYWAVE

MARCH 2025

Contents

Deleted: 25

Deleted: 28

Deleted: 30

Deleted: 30

1 Document Details

1.1 Document Control

Version	Date	Author	Sign-Off
1.0	20/05/2024	Ollie Hermans	Amy Johnson
1.1	01/08/2024	Vijay A	Jacky Cheung
1.1.1	6/8/2024	Art Pai	Jacky Cheung
1.2	09/09/2024	Vijay S	Jacky Cheung
1.3	15/10/2024	Vijay S	Jacky Cheung
1.4	05/02/2025	Lucas Lee	Amy Johnson
1.5	28/02/2025	Marlon Cardenas	Amy Johnson
1.6	24/03/2025	Vijay S	
1.6.1	03/04/2025	Lucas Lee	
1.7	06/05/2025	Vijay S	
1.8	02/07/2025	Marlon Cardenas	

2 Introduction

2.1 Audience

Users who wish to set up a local runtime hosting environment for MyWave solutions.

2.2 Purpose

This document aims to outline the essential steps necessary for locally hosting MyWave for testing purposes.

3 Prerequisites

IMPORTANT: Before proceeding with the deployment, it's crucial to complete the following preparatory steps. You'll need to provide specific information to MyWave, and in return, the appropriate packages will be provided.

3.1 Install Docker Desktop:

1. Download: Go to the Docker Desktop download page and download the installer.
2. Install: Run the installer and follow the on-screen instructions.
3. Launch: Open Docker Desktop to ensure it is running correctly.

3.2 Ensure Java 17(17.0.9 or above) is the Default Version:

1. Check Version: Open a terminal or command prompt and type java -version. It should show Java 17.
2. Set Default Version: If it doesn't, you may need to set Java 17 as the default in your system settings.

3.3 Ports:

Avoid using the following ports:

- 80
- 8080
- 8081
- 5432
- 9092
- 9093

3.4 Azure Open AI subscription

An Azure Open AI subscription is required for Large Language Model (LLM) integration.

3.4.1 Setup instructions

- A corporate email address is required for setup; personal email addresses like Gmail or Outlook are not accepted.
- Ensure you have proper authorisation from your company and clear any setups with your IT department.

Where to find:

1. **Setting up:**
   - Ensure you have a company email address and authorisation.
   - Create an Azure account if you don't have one.
   - Azure does not permit free account tier access to OpenAI services.

2. **Obtain Azure subscription ID:**
   - Go to Azure Portal Page and sign in.
   - Access your subscription ID via the "Subscriptions" service.
   - If not visible, search for "Subscriptions" in the search bar.
   - Click your subscription name to view details.
   - Hover over the subscription ID to copy it to the clipboard, you will use this Subscription ID in subsequent steps, specifically when requesting access to Azure OpenAI Service for Modified Content Filters or Abuse Monitoring.

3. **Azure OpenAI Access Considerations:**

   - **For Standard Use:** You can proceed directly to creating an Azure OpenAI resource. The access request form is not required.

   - **For Modified Content Filters or Abuse Monitoring:**
     - If you need to modify content filters or abuse monitoring, please visit https://aka.ms/oai/access and select the appropriate link for registration.
     - Submit the form and await confirmation email from Microsoft.
     - Save the email for future reference.

4. **Azure OpenAI service setup:**
   - Return to Azure Portal and search for "Azure OpenAI".
   - Click the Azure OpenAI service and click **Create**.
   - Choose your subscription, resource group, region, and provide a resource name.
   - Select the pricing tier (usually a single option).
   - Proceed through the screens to configure network security and optional tags.
   - Review the entries and click **Create** to deploy your resource.
   - Wait a few minutes for the deployment process to complete.

5. **Access Azure OpenAI Studio:**
   - Once deployed, find your resource under Azure OpenAI services.
   - Click the resource name to view details.
   - Access Azure OpenAI Studio from the provided link.

6. **Alternatively:**
   - Directly access Azure OpenAI Studio at https://oai.azure.com/portal.
   - Log in and select your Azure OpenAI resource.

7. **Working with Azure OpenAI Portal:**
    - On the Portal page, deploy a new model to proceed.
    - Select the OpenAI model **"gpt-4o-mini".**
        - For detailed information on model availability in each region, refer to the Azure OpenAI model region availability page
    - Name the deployment and click **Deploy**.
    - Your new deployment will be listed under manage deployments.

For the Client Partners Currently Using the 0613 Model

We recommend deploying the gpt-4o-mini and replace the existing deployment(gpt-35-turbo 0125) with it if you already have it.

3.4.2 Information required by MyWave

- **Open AI Endpoint URL:** The URL endpoint provided by Open AI for integration.
    - **Where to find:** This value can be found in the Keys & Endpoint section when examining your resource from the Azure Portal. An example endpoint is: https://<area>docs-test-001.openai.azure.com/.
- **API Key (Key1):** The API key generated by Open AI for authentication.
    - **Where to find:** This value can be found in the Resource Management > Keys & Endpoint section when examining your resource from the Azure Portal. Ensure you use KEY 1.
- **Deployment Name:** The unique name assigned to the deployment of MyWave on Azure.
    - **Where to find:** This value can be found in the Azure OpenAI Portal under the "Deployments" section, in the "Deployment Name" column.

3.5 Azure AI Document Intelligence Service

An Azure AI Document Intelligence service is required for the text extraction(OCR) integration.

3.5.1 Setup instructions

- A corporate email address is required for setup; personal email addresses like Gmail or Outlook are not accepted.
- Ensure you have proper authorisation from your company and clear any setups with your IT department.

Where to find:

1. **Setting up:**
    - Ensure you have a company email address and authorisation.
    - Create an Azure account if you don't have one.
    - Azure does not permit free account tier access to OpenAI services.

2. **Azure AI Document Intelligence(form recognizer) Setup:**
    - Go to Azure Portal Page and sign in.
    - Click the **+ Create a resource** button and search for " Document intelligence(form recognizer)" in the search bar.
    - Click Document intelligence(form recognizer) Card to create.
    - Click the Create button for the Plan Document Intelligence (form recognizer).
    - Choose your subscription, resource group, region, and provide a resource name.
    - Select the pricing tier (usually a single option).
    - Click Next button and proceed through the screens to configure network security and optional tags.
    - Review the entries and click **Create** to deploy your resource.
    - Wait a few minutes for the deployment process to complete.

3. **Access Document Intelligence Details:**
    - Once deployed, find your resource under Azure AI services.
    - Click the resource name to view details.

3.5.2 Information required by MyWave

- **Azure AI Document Intelligence Endpoint URL:** The URL endpoint provided in the details page for integration.
    - **Where to find:** This value can be found in the Keys & Endpoint section under the Document Intelligence resource details page in Azure Portal. An example endpoint is https://<REGION_NAME>.api.cognitive.microsoft.com.
- **API Key (Key1):** The API key generated by Azure AI Document Intelligence for authentication.
    - **Where to find:** This value can be found in the Keys & Endpoint section under the Document Intelligence resource details page in Azure Portal. Ensure you use KEY 1.

3.6 MyWave Wave files

Wave files exported from the MyWave innovation hub is required to set up InfoPlugins and other configurations.

3.7 SAP Document Information Extraction (DocX) Subscription

Document Information Extraction subscription is required if you are using “Report my expenses” wave model. This service provides the capability to extract information from unstructured data like Invoices. SAP Business One Integration Service uses the Document Information Extraction Service to extract information from Invoices uploaded by users.

Instructions to set up a trial subscription for DocX can be found here. For Enterprise subscription, please reach out to your SAP Sales/Support Team. Please note that this service is only available in Europe Region.

A brief summary of the instructions is provided below for convenience. Please refer to the main document if you need further information.

3.7.1 Create a New DocX Service Instance

1) Login to SAP Cloud Platform Trial Cockpit.
2) Click the "Access Cloud Foundry Trial" link and select your region(make sure you have selected Europe (Frankfurt)).
3) Click the "dev" link in the Spaces section to access Cloud Foundry Trial dev Space.
4) Click the "Service Marketplace" link from the side menu under Services.
5) Search for "Document Information Extraction trial service" and click the tile.
   Note: If the service is not visible, configure the entitlement and quota as described in the SAP Community Blog.
6) Click "Instances" from the side menu drawer.
7) Click the "New Instance" button. Follow the create instance wizard:
   a. Click "Next" (no parameters required).
   b. Click "Next" again (no application deployed in Cloud Foundry).
8) Enter an instance name (e.g., "aiservices-dox ").
9) Once new instance successfully created, you will see the status Created in Last Operation column.

3.7.2 Create New Service Key

A service key is required to access the DocX Service instance.

1) Click on the instance e.g., "aiservices-dox ".
2) Click Service Keys from side menu and click Create Service Key button.
3) A create instance wizard pop-up will appears. Enter the service key name. Click Save to proceed.

Once the service key has been created, you will see JSON output containing API endpoint URL and User Account and Authentication (UAA) details. Please save this Service Key Json for your reference.

A sample Service Key JSON which has the required details is highlighted below. You will need these values to be used in the Sections 5.5.6, 5.6.3

```
{
"url": "https://docx-service.cfapps.eu10.hana.ondemand.com",
"html5-apps-repo": {
"app_host_id": "e32e0e0d-5d82-4cda-bffe-0a1c5e636d56"
},
"uaa": {
"tenantmode": "shared",
"sburl": "https://internal-xsuaa.authentication.eu10.hana.ondemand.com",
"subaccountid": "fc0ecd13-b2b1-4a31-860d-7121dacccf08",
"credential-type": "binding-secret",
"clientid": "sb-55fdfb7e-2bd6-4db1-840a-d6f90ba06eb5!b409078|na-05dd5c92-af7a-4df4-b4fc-e3a10b1394dc",
"xsappname": "583266dc-266d-42c4-bfe5-0cba78788764!b409078|na-410ef2b9-81bc-49c5-b7b4-e3d84759f838",
"clientsecret": "ec15b96d-45f4-4216-a3c3-25e35342d449=",
"serviceInstanceId": "51a66839-b05c-4b39-8d67-07a563b42190",
"url": "https://expense-report-d3t34d.authentication.eu10.hana.ondemand.com",
"uaadomain": "authentication.eu10.hana.ondemand.com",
"verificationkey": "-----BEGIN PUBLIC KEY-----\neyJhbGciOiJIUzI1NiJ9.eyJSb2xlIjoiQWRtaW4iLCJJc3N1ZXIiOiJJc3N1ZXIiL
\n-----END PUBLIC KEY-----",
"apiurl": "https://api.authentication.eu10.hana.ondemand.com",
"identityzone": "expense-report-d3t34d",
"identityzoneid": "795feaa0-8766-4a5f-9798-627fdd79c84d",
"tenantid": "795feaa0-8766-4a5f-9798-627fdd79c84d",
"zoneid": "795feaa0-8766-4a5f-9798-627fdd79c84d"
},
"swagger": "/document-information-extraction/v1/",
"saasregistryenabled": true,
"endpoints": {
"backend": {
"url": "https://docx-service.cfapps.eu10.hana.ondemand.com",
"timeout": 30000
}
},
"sap.cloud.service": "com.sap.apps.documentinformationextraction",
"tenantuiurl": "https://expense-report-d3t34d.ui.dox.aiservices.cfapps.eu10.hana.ondemand.com"
}
```


SAP_DOCX_API_HOST

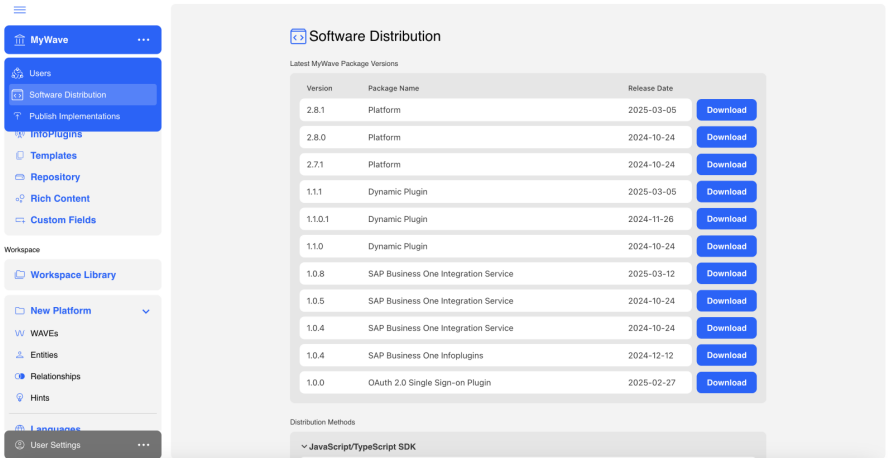
DOCUMENT_UPLOAD_CLIENT_ID

DOCUMENT_UPLOAD_CLIENT_SECRET

+ /oauth/token = DOCUMENT_UPLOAD_TOKEN_URL

3.8 Files Provided by MyWave

Once you have completed the Pre-Deployment Preparation, download the following files from the MyWave Innovation Hub. They can be accessed via the menu in the top left corner.



- **MyWave SAP B ONE Web Extension Package:** A React Single Page Application (SPA) serving as a web extension for SAP Business One.
- **mywave-ai-platform-2.8.2.zip:** The core runtime environment for the MyWave platform.
- **dynamic-config-mywave-ai-plugin-1.1.2-mywave-ai-platform-2.8.2.zip**
- **sap-business-one-integration-service-1.0.9.jar:** Integration service which provides SAP B1 SSO and LLM capabilities
- **sap-business-one-infoplugins-1.0.4.jar:** Java plugins for integration with SAP Business One.

Deleted: 1

Deleted: 1

Deleted: .1

Deleted: 1

4 Setting up SAP Business One (SAP B1)

The SAP B1 Web Extension Client is where the front end of the MyWave client is hosted. A React Web Extension will need to be setup, and a SAP B1 OIDC Provider URL will be needed for the single sign-on (SSO) via integration service.

4.1 Prerequisites

A Client ID and OIDC Provider URL are required for setting up the SAP B1 Web Extension Client. Follow these steps below to obtain them and make sure to record them for future reference.

Client ID: The ID registered on SAP B1.

- **Where to find:**
  - Log in to the SAP Business One Extension Single Sign-On Manager.
  - Click **Register** to begin registration.
  - In the Client Information section:
    - Name the client "SinglePageApp1".
    - Select **Single Page App** as the client type.
  - In the Redirect URIs section:
    - Enter your sap b1 web client URI, https://<sap-b1-webclient-url>/*.
    - Remember that wildcard URIs are accepted.
  - Proceed to the "Review" section in the wizard.
  - Submit the registration.
  - Copy the client ID provided and save it for your app's use.

SAP B1 OIDC Provider URL: Required for SSO via the integration service.

- **Where to find:** This URL can be found by logging onto the SAP B1 Control Center, navigating to the "Security" tab, and locating the "SLD Address" under “SAP Business One Authentication Service” section
- When setting up the OIDC URL, please ensure that the SLD Address is appended with /sld/sld0100.svc. For example, the SLD Address in the Control Center is "https://htpc213p01.sapb1.com:40000" then the SAP B1 OIDC URL should be https://htpc213p01.sapb1.com:40000/sld/sld0100.svc

4.2 Setup instructions

1. If you are using a **Windows machine**, please set up your local environment by following the instructions at this link.
2. Unzip the source code from the MyWave SAP B1 Web Extension Package.
3. Using a text editor of your preference, create a file at the root level of the folder you have unzipped in the previous step. Name the file .env
4. Set up the following configurations in the environment variable in .env.

```
VITE_CLIENT_ID=<your_extension_manager_client_id>
VITE_SLD_ADDRESS=<B1 SLD ADDRESS>/sld/sld0100.svc
VITE_API_URL=https://localhost/mywave
VITE_LOGIN_URL=https://localhost/integration/api/sso/login
VITE_REFRESH_TOKEN_URL=https://localhost/integration/api/ping
```

```
VITE_LLM_INTENT_RECOGNISE_ENDPOINT=https://localhost/integration/api/llm/intent
VITE_LLM_INTENT_RECOGNISE_API_KEY=<your_llm_intent_recognition_api_key>
```

Example Configurations

Configuration	Example	Description
VITE_CLIENT_ID	b1-ext-446456-7ac2-406b-b83d-5bde914e1a51	The Client ID you have obtained at the step 4.1 of this guide. Replace the string <your_extension_manager_client_id> with that value.
VITE_SLD_ADDRESS	https://your-oidc-provider-server:40000/sld/sld0100.svc	SAP B1 OIDC Provider URL you obtained at the step 4.1 of this guide.
VITE_API_URL	https://localhost/mywave	The URL of the Mywave Platform Runtime.
VITE_LOGIN_URL	https://localhost/integration/api/sso/login	SSO Endpoint provided by the Integration Service
VITE_REFRESH_TOKEN_URL	https://localhost/integration/api/ping	Endpoint to refresh authenticated token
VITE_LLM_INTENT_RECOGNISE_ENDPOINT	https://localhost/integration/api/llm/intent	LLM Intent recognition endpoint provided by the Integration service
VITE_LLM_INTENT_RECOGNISE_API_KEY		This will be required only if you are using your own custom intent recognition service instead of the one provided by MyWave Integration Service.

5. You need to ensure you have Node.js and pnpm installed locally as per the below versions:

```
node: 20.12.2
pnpm: 9.0.6
```

6. If you are unsure what versions you have installed, you can check it by running following commands:

```
node -v
pnpm -v
```

7. If you do not have these versions, please run the following command to set these up:

```
./scripts/preinstall.sh
```

8. To install all dependencies, run the following command:

```
pnpm install
```

9. Once all the dependencies are Installed, we should be able to build and deploy Web Extension. Build the extension for deployment by running the following command:

```
pnpm build:ext
```

10. Following prompt will ask for desired name and version and generate a <extension-name>_<version_number>.mtar file in the mta_archives directory at the root of the project.

```
./
├─ dist/
├─ mta_archives/
│  ├─ <extension-name>_<new_version_number>.mtar
├─ src/
```

11. This .mtar file can now be uploaded via your SAP B1 Extension Manager portal overriding any previous version of the deployed app. Please refer to the Deploying Web Client Extension Guide from SAP to understand how to deploy a mtar file to SAP B1.

4.3 Customise UI wordings

Wordings on SAP B1 Web Client can be customised with changing json files under /public/locales/en

There are 5 files under public/locales/en. Search and replace the text that is targeted to change

- mwAuthentication.json
- mwCommon.json
- mwConversation.json
- mwConversationHistory.json
- mwErrors.json

If you need to have your wordings in a different language, say Italian, you need to create a folder inside of /public/locales folder, name it *it* as abbreviation of the Italian language and copy all 5 files from *en* folder into *it* folder. Translate wordings into the language you have chosen. Do not update the names of the variables (on left-hand side).

For example, mwCommon.json looks like this:

```
{
  CANCEL: "Cancel",
  UPDATE: "Update",
  ...
}
```

Translate the wording placed on the right-hand side and save all modified files.

4.4 Configuring CSP Settings in SAP B1

1). Login to SAP B1 Web Client
2). Navigate to Settings in profile then click on General Settings
3). Update the Content Security Policy as below:

Please make sure that you are only updating <SLD_Address> , <mywave-runtime-host>, and data: to your existing connect-src and frame-src settings **and not replacing them.** Add blob: to default-src and 'unsafe-inline' to style-src:

```
connect-src 'self' <SLD_Address> <mywave-runtime-host> data: blob:; frame-src 'self'
<SLD_Address> <mywave-runtime-host> data: blob:; default-src 'self' <SLD_Address>
<mywave-runtime-host> blob: data:; style-src 'self' *.sap.com *.hana.ondemand.com
'unsafe-inline';
```

For example,

```
connect-src 'self' htpc21320p01.sapb1.com:* <mywave-runtime-host> data:; frame-src
'self' htpc21320p01.sapb1.com:* <mywave-runtime-host> data: blob:; default-src 'self'
htpc21320p01.sapb1.com:* <mywave-runtime-host> blob: data:; style-src 'self' *.sap.com
*.hana.ondemand.com 'unsafe-inline';
```

Note:

- <mywave-runtime-host> is the domain name of your MyWave Platform server runtime. Use https://localhost in your case as we are setting up mywave runtime for use on your localhost.
- In some cases, specifying the SLD_Address with wildcard for ports like htpc21320p01.sapb1.com:* might not work. In that case, you might have to make the port numbers explicit, like https://htpc21320p01.sapb1.com:40000 https://htpc21320p01.sapb1.com:40020.

5 Setting up local hosting

Once you have completed the above prerequisites, follow these steps to setup local hosting for MyWave.

5.1 Setup Nginx

5.1.1 Install Nginx

This command installs Nginx using the Homebrew package manager.

```
brew install nginx
```

5.1.2 Create a certificate

This command generates a self-signed SSL certificate for local development.

```
openssl req -x509 -out /opt/homebrew/etc/nginx/server.crt -keyout
/opt/homebrew/etc/nginx/server.key \
-newkey rsa:2048 -nodes -sha256 \
-subj
'/C=NZ/ST=Auckland/L=Auckland/O=MyWave/OU=IT/CN=localhost/emailAddress=abc@mywave.ai'
\
-extensions EXT -config <( \
printf "[dn]\nCN=localhost\n[req]\ndistinguished_name =
dn\n[EXT]\nsubjectAltName=DNS:localhost\nkeyUsage=digitalSignature\nextendedKeyUsage=s
erverAuth")
```

5.1.3 Edit nginx.conf file

This command navigates to the directory containing the Nginx configuration file and opens it for editing.

```
cd /opt/homebrew/etc/nginx
vi nginx.conf
```

5.1.4 Edit server in the nginx.conf

This configuration block defines how Nginx handles incoming requests and forwards them to the appropriate backend services.

```
server {
      listen       80;
      listen       443 ssl;
      server_name  localhost;
      ssl_certificate  /opt/homebrew/etc/nginx/server.crt;
      ssl_certificate_key /opt/homebrew/etc/nginx/server.key;
      client_max_body_size 10M;

      location /mywave {
        proxy_pass http://localhost:8080/mywave;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_cache_bypass $http_upgrade;
      }

      location /integration {
        proxy_pass http://localhost:8081/integration;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_cache_bypass $http_upgrade;
      }
}
```

5.1.5 Stop and start Nginx

This command checks the Nginx configuration for syntax errors, stops the Nginx server if it's running, and then starts it again.

```
nginx -t && nginx -s stop && nginx
```

5.1.6 Add certificate to local keystores

5.1.6.1 For Java 17:

This command imports the SSL certificate into the Java 17 keystore to establish trust.

```
sudo keytool -import -alias localhost -file /opt/homebrew/etc/nginx/server.crt -
keystore {YOUR_JAVA_17_HOME}/lib/security/cacerts
Password: {LOGIN_PASSWORD_OF_YOUR_LOCAL_MACHINE}
Enter keystore password: changeit
Trust this certificate? [no]: yes
```

5.2 Working directory and required files

5.2.1 Working directory

This sets the working directory to "~/Documents/mywave" for the subsequent steps.

```
WORKING_DIRECTORY=~/Documents/mywave
cd $WORKING_DIRECTORY
```

5.2.2 Required files

Copy the files provided earlier under $WORKING_DIRECTORY/tmp directory.

- mywave-ai-platform-2.8.2.zip
- dynamic-config-mywave-ai-plugin-1.1.2-mywave-ai-platform-2.8.2.zip
- sap-business-one-infoplugins-1.0.4.jar
- sap-business-one-integration-service-1.0.9.jar

Deleted: 1

Deleted: 1

Deleted: .1

Deleted: 1

5.3 Running Docker

5.3.1 Copy Docker Compose File

Copy the docker-compose.yaml file found here to your $WORKING_DIRECTORY

5.3.2 Run Zookeeper, Kafka, and Postgresql on Docker

This command initiates the Docker containers for Zookeeper, Kafka, and PostgreSQL.

```
docker-compose up
```

After running the command, you should observe the following Docker containers:

- mywave-zookeeper
- mywave-kafka
- mywave-postgres-multi

5.4 Check connectivity to MyWave Licence Server

The MyWave Platform needs to be able to access our licence server to verify your licence.

You can verify the connectivity by running the following command on the machine running the platform

```
curl -v https://licence.app.mywave.me/ping
```

You should receive a 200 OK response as below

```
$ curl -v https://licence.app.mywave.me/ping
*   Trying 130.211.45.214:443...
* TCP_NODELAY set
* Connected to licence.app.mywave.me (130.211.45.214) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
```

```
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN, server accepted to use h2
* Server certificate:
*  subject: CN=licence.app.mywave.me
*  start date: Jun  2 21:40:04 2024 GMT
*  expire date: Aug 31 22:33:57 2024 GMT
*  subjectAltName: host "licence.app.mywave.me" matched cert's "licence.app.mywave.me"
*  issuer: C=US; O=Google Trust Services LLC; CN=GTS CA 1D4
*  SSL certificate verify ok.
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* Using Stream ID: 1 (easy handle 0x55d5adaf00e0)
> GET /ping HTTP/2
> Host: licence.app.mywave.me
> user-agent: curl/7.68.0
> accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* Connection state changed (MAX_CONCURRENT_STREAMS == 100)!
< HTTP/2 200
< content-length: 0
< date: Fri, 19 Jul 2024 20:51:16 GMT
< via: 1.1 google
< alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
<
* Connection #0 to host licence.app.mywave.me left intact
```

If the command returns an error, that means our licence server is not accessible from the machine and needs to be fixed before moving further.

5.5 Setup the MyWave platform

5.5.1 Create necessary directories:

These commands create directories for storing logs and conversation configurations.

```
mkdir -p ./mywave-ai-platform/logs
mkdir -p ./mywave-ai-platform/conversations
```

5.5.2 Extract MyWave AI Server:

Use the following command to unzip the MyWave AI Server into the target directory:

```
unzip ./tmp/mywave-ai-platform-2.8.1.zip -d ./mywave-ai-platform
```

5.5.3 Extract Dynamic Config Conversation Plugin:

This command extracts the Dynamic Config Conversation Plugin files into the plugins directory of the MyWave AI Platform:

```
unzip ./tmp/dynamic-config-mywave-ai-plugin-1.1.2-mywave-ai-platform-2.8.2.zip -d
./mywave-ai-platform/plugins
```

Deleted: 1.1

Deleted: 1

5.5.4 Copy SAP Business One InfoPlugins:

This command copies the SAP Business One InfoPlugins JAR file to the plugins directory.

```
cp ./tmp/sap-business-one-infoplugins-1.0.4.jar ./mywave-ai-platform/plugins
```

5.5.5 Extract Wave Files :

These commands extract and move Wave files to the conversations directory.

```
unzip ./tmp/<wave-export-file-name>.zip -d ./mywave-ai-platform/conversations/
```

5.5.6 Enabling Debug Logs for Troubleshooting (Optional)

⚠ **Warning:** Enabling debug logs may result in Personally Identifiable Information (PII) being captured in the logs. This should only be enabled for debugging purposes and not recommended in production environments without proper precautions. Use this feature responsibly to protect sensitive data.

Enabling debug logs is not required for the server to start. Enable this only if you need detailed HTTP request and response logs. Follow these steps to enable debug logs:

1. **Locate the Configuration File**, navigate to the configuration file at: `$WORKING_DIREXTORY/mywave-ai-platform/config/rolling-file-log4j2.xml`
2. **Uncomment the Debug Logger Section** :

```
<!-- Uncomment the following section to turn on debugging logs for mywave code -->
<!--
<Logger name="mywave" level="DEBUG">
      <AppenderRef ref="System"/>
</Logger>
-->
```

3. **Restart the Server**
    a. Restart the server to apply the changes.
    b. Alternatively, if this is the first time starting the server, ensure the changes are made before starting.

5.5.6.1 Viewing Debug Logs

Once the debug logs are enabled, all HTTP request parameters and response details will be logged in the `server.log` file. You can find this file in the following directory:
`$WORKING_DIRECTORY/mywave-ai-platform/logs/`

5.5.6.2 Example of Request and Response Debug Logs

Below is an example of how a request and response would appear in the server.log file:

```
2024-10-14 17:12:22,101 DEBUG [INFO_PLUGINS-9-C-1]
Base URL: https://sapb1host:50000/b1s/v2
Path: /Users
HTTP Method: GET
Headers: {Content-Type=application/json, x-b1-companyid=0001, Accept=application/json,
Authorization=Bearer ...}
Query Parameters: {$select=UserCode,UserName,InternalKey}
Request Body:
{}

2024-10-14 17:12:24,701 DEBUG [INFO_PLUGINS-9-C-1] Status code: 200
Response Body: {
    "@odata.context":"https://sapb1host:50000/b1s/v2/$metadata#Users",
    "value" :
    [
        {
            "InternalKey" : 2,
            "UserCode" : "B1i",
            "UserName" : "B1i"
        },
        {
            "InternalKey" : 3,
            "UserCode" : "EDsUser",
            "UserName" : "EDsUser"
        }
    ]
}
Headers: {
```

```
    content-type=[application/json;odata.metadata=minimal;charset=utf-8],
    date=[Mon, 14 Oct 2024 04:12:23 GMT],
    odata-version=[4.0],
    preference-applied=[odata.maxpagesize=100],
    server=[Apache/2.4.56 (Unix)],
    set-cookie=[ROUTEID=.node4; path=/;Secure;SameSite=None,
clxservice=2335007333.1.372128608.4103876608; path=/; secure],
    transfer-encoding=[chunked],
    vary=[Accept-Encoding]
}
```

5.5.7 Run MyWave Platform:

This command runs the MyWave Platform with your configured settings. Adjust the values as needed for your setup and save a copy with your custom values for easier future updates.

```
SERVER_CONTEXT="/mywave" \
SERVER_PORT="8080" \
DEFAULT_LOCALE="en_AU" \
DB_USERNAME="mywavedev" \
DB_PASSWORD="mywavedev" \
DB_NAME="mywave_ai" \
DB_HOST="localhost" \
LICENCE_KEY="<LICENCE_KEY>" \
JWT_TOKEN_KEY="75khazLQmPDCUdZun9uvvEHGfmurZZnfdQi0gbSV1AWOXgm0qT3VxeZV1HSPVSV7ZcQaW0l
Tu2rnYsXESNw8sFlXH7C3rPwQl9hnJDTOIzAZVGoOgGmJTGA4YLBoSrUGCNbIDyqTVIBCOXptBJriPts2XUmNH
hkj6XetOSMYa8ne67nsOc7YShJfiCFWdLlAluYFpeicfg36cAAQoddgmRwoL85CS8V9CguBXcmmfqCgnpqqSOA
fqB8GjFDxW5WVrbmar6yB46aMPDkQo5wurOutptRgcgCH8xVyRXeNCWFdQ2ZE8eRubgRdDOqpV3Ux8KMwHqWBH
jQBqEbSydZvr" \
CLIENT_API_KEY="ATyc1PgFzb6Z5Ytd2fpQHnDgN6itaV2BrPULlui1INE1s6CYeRDABowYlyWf4Xe08hD1cB
zKMhi61O6aKdGnbvQn1R5GlmkEbJ7IrmJAorzoHlgMzqvTnPdLgFvdAuVL" \
KAFKA_BROKER_ADDRESSES="localhost:9092" \
DYNAMIC_CONVERSATION_CONFIGURATIONS_FOLDER="file:<WORKING_DIRECTORY>/mywave-ai-
platform/conversations" \
SAP_B1_INTEGRATION_SERVICE_BASE_URL="https://localhost/integration" \
SAP_B1_INTEGRATION_SERVICE_GET_ACCESS_TOKEN_URL_PATH="/api/sso/account/token" \
SAP_B1_BASE_URL="<sap-b1-service-layer-url>/b1s/v2" \
DOCUMENT_UPLOAD_TOKEN_URL=<DOCUMENT_UPLOAD_TOKEN_URL> \
DOCUMENT_UPLOAD_CLIENT_ID=<DOCUMENT_UPLOAD_CLIENT_ID> \
DOCUMENT_UPLOAD_CLIENT_SECRET=<DOCUMENT_UPLOAD_CLIENT_SECRET> \
DOCUMENT_UPLOAD_OPTIONS_JSON="{\"schemaId\":\"cf8cc8a9-1eee-42d9-9a3e-
507a61baac23\",\"clientId\":\"default\",\"documentType\":\"invoice\",\"receivedDate\":
\"2020-02-
17\",\"enrichment\":{\"sender\":{\"top\":5,\"type\":\"businessEntity\",\"subtype\":\"s
upplier\"},\"employee\":{\"type\":\"employee\"}}}" \
DOCUMENT_UPLOAD_OPTIONS_PO_JSON="{\"schemaId\":\"fbab052e-6f9b-4a5f-b42f-
29a8162eb1bf\",\"clientId\":\"default\",\"documentType\":\"purchaseOrder\",\"receivedD
ate\":\"2020-02-
17\",\"enrichment\":{\"sender\":{\"top\":5,\"type\":\"businessEntity\",\"subtype\":\"s
upplier\"},\"employee\":{\"type\":\"employee\"}}}" \
AZURE_OPENAI_API_KEY="<AZURE_OPENAI_API_KEY>" \
AZURE_OPENAI_CHAT_DEPLOYMENT="<AZURE_OPENAI_CHAT_DEPLOYMENT>" \
AZURE_OPENAI_ENDPOINT="<AZURE_OPENAI_ENDPOINT>" \
```

```
AZURE_OPENAI_TEMPERATURE=0 \
AZURE_OPENAI_CHAT_URL="<AZURE_OPENAI_CHAT_URL>" \
./mywave-ai-platform/bin/mywave-ai
```

Example Values for some of the configurations above

Configuration	Example Config	Description
LICENCE_KEY	**a3a1d699-9db2-45c2-a034-97789f4f91d4**	The licence key issued by MyWave
SAP_B1_BASE_URL	https://yourcompany-sapb1-server:50000/b1s/v2	SAP B1 Service Layer API base URL - **<SERVICE LAYER ADDRESS>/bis/v2** You can find <SERVICE LAYER ADDRESS> under B1 control centre > Services Layer. It has the format of **https://<hostname>:<port>**
DOCUMENT_UPLOAD_TOKEN_URL	https://docx-service.authentication.eu10.hana.ondemand.com/oauth/token	Required only if you are using "Report my expenses" wave model. The URL can be found from the service key JSON under the uaa section, in the url field. An example value might be something like `https://expense-demo-6nyshh3n.authentication.eu10.hana.ondemand.com` Please append ***/oauth/token*** suffix to the ***uaa*** URL. The final value should look something like below `https:// docx-service.authentication.eu10.hana.ondemand.com/oauth/token` Please refer to Section 3.6.2 for more details.
DOCUMENT_UPLOAD_CLIENT_ID	sb-5886639f-230f-49d0-8e09-e0c499d77136-1c9f35ecfb2d!b20821	Required only if you are using "Report my expenses" wave model. This can be found under the ***uaa*** section, in the ***clientid***

		field from the service key's JSON. Please refer to Section 3.6.2, to know how to create a Service Key Json.
DOCUMENT_UPLOAD_CLIENT_SECRET	d2d9b2c0-42014532-b8ee-65c37\$NwyZJtlZxB41c2cBH0lgk=	Required only if you are using "Report my expenses" wave model. This can be found under the ***uaa*** section, in the ***clientsecret*** field from the Service Key JSON. Please refer to Section 3.6.2, to know how to create a Service Key Json.
~~INNOVATION_OPEN_AI_BASE_URL~~ *(deprecated)*	~~https://trial-openai-dev.openai.azure.com~~	~~The LLM API Endpoint for backward compatibility and will be removed in a subsequent release.~~
~~INNOVATION_OPEN_AI_EXTRACT_JSON_URL~~ *(deprecated)*	~~/openai/deployments/gtp-4o-mini/chat/completions~~	~~The chat completion API URL path for backward compatibility and will be removed in a subsequent release.~~
~~WAVE_LLM_APIKEY~~ *(deprecated)*	~~Efc8f4d3727a465dbf80438d948b557c~~	~~The LLM API Secret Key for backward compatibility and will be removed in a subsequent release.~~
~~OPENAI_API_ENDPOINT~~ *(deprecated)*	~~https://trial-openai-dev.openai.azure.com~~	~~The LLM API Endpoint~~
~~OPENAI_API_CHAT_URL~~ *(deprecated)*	~~/openai/deployments/gtp-4o-mini/chat/completions~~	~~The chat completion API URL path~~
~~OPENAI_API_KEY~~ *(deprecated)*	~~Efc8f4d3727a465dbf80438d948b557c~~	~~The LLM API Key~~
AZURE_OPENAI_API_KEY	Efc8f4d3727a465dbf80438d948b557c	Key 1 under the Keys and Endpoint in Azure OpenAI
AZURE_OPENAI_CHAT_DEPLOYMENT	gpt-4o-mini	Name of the deployment
AZURE_OPENAI_ENDPOINT	https://trial-openai-dev.openai.azure.com	Azure OpenAI Endpoint under the Keys and Endpoint
AZURE_OPENAI_TEMPERATURE	0	Parameter that controls the randomness or creativity of the model's output. The temperature value typically ranges from 0 to 2.

AZURE_OPENAI_CHAT_URL	/openai/deployments/gpt-4o-mini/chat/completions?api-version=2024-08-01-preview	Specifies the Azure OpenAI API endpoint used for generating chat-based AI responses via the deployment model. It defines the deployment name (e.g. gpt-4o-mini), the chat completion endpoint, and the API version (e.g. 2024-08-01-preview)

5.5.8 Webhook Configuration for MyWave Platform (Optional)

Formatted: Font: (Default) Poppins Medium, Not Bold, Font color: Custom Color(RGB(26,26,56)), Complex Script Font: +Headings CS (Times New Roman), 14 pt, Not Bold

To enable webhook integration with external providers in the MyWave Platform, define the following properties in a `webhook.conf` configuration file, for example:

```
# List of enabled webhook providers (comma-separated)
webhook.providers=SAPB1,GOOGLE

# SAP Business One Webhook Configuration
webhook.SAPB1.apikey=SDF2345FGH-DB50-DF45-123F-DFGDFG567854
webhook.SAPB1.get_auth_reference_info_plugin_name=get_authentication_reference
webhook.SAPB1.start_conversation_notification_info_plugin_names=notification_email

# Google Webhook Configuration
webhook.GOOGLE.apikey=DFD4DXRGGH45-234B-345X-QWSDF345FG
```

Example Values for some of the configurations above:

Configuration	Example Config	Description
webhook.providers	SAPB1,GOOGLE	List of enabled webhook providers (comma-separated)
webhook.SAPB1.apikey	F62F86AB-CD78-4A5B-9A79-LKSDFJDHERF	Valid API key used to authenticate webhook requests from each provider
webhook.SAPB1.get_auth_reference_info_plugin_name	get_authentication_reference	InfoPlugin name used to get the authentication reference
webhook.SAPB1.start_conversation_notification_info_plugin_names	notification_email	InfoPlugin name used to
webhook.GOOGLE.apikey	ASD868690-KJHG-4A5B-9A79-9847592384579	Valid API key used to authenticate webhook requests from each provider

Notes:

- webhook.providers specifies the active webhook integrations. Each value must match a defined section in the config.
- apikey should be a valid API key used to authenticate webhook requests from each provider.

- InfoPlugin names (e.g., get_auth_reference_info_plugin_name) refer to registered infoplugin handlers used by the MyWave Platform to process specific webhook events.

5.6 Setup the SAP Business One Integration Service

5.6.1 Create necessary directories:

This command creates a directory for storing logs of the SAP Business One integration service.

```
mkdir -p ./business-one/service/logs
```

5.6.2 Copy SAP Business One integration service JAR file:

This command copies the SAP Business One integration service JAR file to the service directory.

```
cp ./tmp/sap-business-one-integration-service-1.0.9.jar ./business-one/service/
```

5.6.3 Run SAP Business One integration service:

This command runs the SAP Business One integration service with the specified configurations and environment variables. Adjust the values as needed for your setup.

Ensure to keep a copy of the following command once completed with your own variables, as it will simplify updating Waves in the future.

```
SERVER_SERVLET_CONTEXT_PATH="/integration" \
SERVER_PORT="8081" \
DB_USERNAME="mywavedev" \
DB_PASSWORD="mywavedev" \
DB_NAME="integrationservice" \
DB_HOST="localhost" \
DB_PORT="5432" \
PLATFORM_URL="https://localhost/mywave" \
PLATFORM_JWT_SECRET_TOKEN_KEY="75khazLQmPDCUdZun9uvvEHGfmurZZnfdQi0gbSV1AWOXgm0qT3VxeZ
V1HSPVSV7ZcQaW0lTu2rnYsXESNw8sFlXH7C3rPwQl9hnJDTOIzAZVGoOgGmJTGA4YLBoSrUGCNbIDyqTVIBCO
XptBJriPts2XUmNHhkj6XetOSMYa8ne67nsOc7YShJfiCFWdLlAluYFpeicfg36cAAQoddgmRwoL85CS8V9Cgu
BXcmmfqCgnpqqSOAfqB8GjFDxW5WVrbmar6yB46aMPDkQo5wurOutptRgcgCH8xVyRXeNCWFdQ2ZE8eRubgRdD
OqpV3Ux8KMwHqWBHjQBqEbSydZvr" \
PLATFORM_CLIENT_API_KEY="ATyc1PgFzb6Z5Ytd2fpQHnDgN6itaV2BrPULlui1INE1s6CYeRDABowYlyWf4
Xe08hD1cBzKMhi61O6aKdGnbvQn1R5GlmkEbJ7IrmJAorzoHlgMzqvTnPdLgFvdAuVL" \
CONVERSATION_CONFIGURATIONS_FOLDER="file:<WORKING_DIRECTORY>/mywave-ai-
platform/conversations" \
AZURE_OPENAI_ENDPOINT="<AZURE_OPENAI_ENDPOINT>" \
AZURE_OPENAI_API_KEY="<AZURE_OPENAI_API_KEY>" \
AZURE_OPENAI_CHAT_DEPLOYMENT="<AZURE_OPENAI_CHAT_DEPLOYMENT>" \
AZURE_OPENAI_TEXT_EMBEDDING_DEPLOYMENT="text-embedding-ada-002" \
SAP_B1_OIDC_PROVIDER_HOST=<SAP_B1_OIDC_PROVIDER_HOST> \
SAP_B1_CLIENT_ID="<SAP_B1_CLIENT_ID>" \
SAP_DOCX_API_HOST=<SAP_DOCX_API_HOST> \
AZURE_DOCUMENT_INTELLIGENCE_API_ENABLED="<AZURE_DOCUMENT_INTELLIGENCE_API_ENABLED>" \
AZURE_DOCUMENT_INTELLIGENCE_API_HOST="<AZURE_DOCUMENT_INTELLIGENCE_API_HOST>" \
```

```
AZURE_DOCUMENT_INTELLIGENCE_API_KEY="<AZURE_DOCUMENT_INTELLIGENCE_API_KEY>" \
LLM_PROXY_API_ENABLED="<LLM_PROXY_API_ENABLED>" \
LLM_PROXY_API_CHAT_URL="<LLM_PROXY_API_CHAT_URL>" \
KNOWLEDGE_SEARCH_TYPES="<KNOWLEDGE_SEARCH_TYPES>" \
java -jar ./business-one/service/sap-business-one-integration-service-1.0.9.jar
```

Example Values for some of the configurations above

Configuration	Example Config	Description
AZURE_OPENAI_ENDPOINT	https://trial-openai-dev.openai.azure.com	Azure OpenAI Endpoint under the Keys and Endpoint
AZURE_OPENAI_API_KEY	Efc8f4d3727a465dbf80438d948b557c	Key 1 under the Keys and Endpoint in Azure OpenAI
AZURE_OPENAI_CHAT_DEPLOYMENT	test-openai-deployment	Name of the deployment
SAP_B1_OIDC_PROVIDER_HOST	https://yourcompany-sapb1-oidc-server:40000	URL to the B1 OIDC provider. This URL can be found by logging onto the SAP B1 Control Center, navigating to the "Security" tab, and locating the "SLD Address".
SAP_B1_CLIENT_ID	b1-ext-3721c34a-9f2e-42f9-8f6c-1a9e2d5f7b83	The client ID registered in SAP B1 for single sign-on integration. This ID can be found by logging onto the SAP B1 Extension Single Sign-On Manager Refer to Section 4.1 for further details.
SAP_DOCX_API_HOST	your-docx-services-cfapps.eu10.hana.ondemand.com	Required only if you are using “Report my expenses” wave model. This value can be found in the “***url”***

		field in the root element of the JSON. Please refer to Section 3.6.2 for more details.
AZURE_DOCUMENT_INTELLIGENCE_API_ENABLED	true or false (default is false)	To enable the Azure AI Document Intelligence API for the Sales Order from Customer Purchase Order wave, set this value to true. The default is false, which disables the API
AZURE_DOCUMENT_INTELLIGENCE_API_HOST	https://trial-openai-dev.openai.azure.com	Azure AI Document Intelligence API Endpoint and **this should not have a trailing slash**
AZURE_DOCUMENT_INTELLIGENCE_API_KEY	Efc8f4d3727a465dbf80438d948b557c	Key 1 under the Keys and Endpoint in Azure AI Document Intelligence
LLM_PROXY_API_ENABLED	true or false (default is false)	Enables the LLM proxy feature for the Sales Order from Customer Purchase Order wave when set to true. This allows API calls to the LLM through a proxy, often used to avoid CORS issues. If set to false (default), the LLM proxy feature is disabled, and direct LLM calls will not be routed through the proxy.
LLM_PROXY_API_CHAT_URL	/openai/deployments/gpt-4o-mini/chat/completions?api-version=2024-08-01-preview	Specifies the Azure OpenAI API endpoint used for generating chat-based AI responses through a specific deployment model (e.g., gpt-4o-

		mini). This includes the chat completions path and API version (e.g., 2024-08-01-preview). To activate this chat URL, ensure that the LLM_PROXY_API_ENABLED flag is set to true. This proxy endpoint helps avoid CORS issues when calling the LLM from Custom Fields.
KNOWLEDGE_SEARCH_TYPES	ExtractPurchaseOrderDocumentAndModificationWithSizeInBusinessOne_item, CreateMultipleSalesOrdersFromPurchaseOrderGroupedByDeliveryDatesInBusinessOne_item,CreateMultipleSalesOrdersFromPurchaseOrderGroupedByDeliveryDatesInBusinessOne_businessPartner	Comma separated variables used in knowledge search as types. For example, usage in custom fields, prefer to use the following format: custom_field_name + '_' + specific_data_name
~~LLM_PROXY_API_HOST~~ *(deprecated)*	~~https://trial-openai-dev.openai.azure.com~~	~~The LLM(Azure OpenAI) API Endpoint and **this should not have a trailing slash**~~
~~LLM_PROXY_API_KEY~~ *(deprecated)*	~~Efc8f4d3727a465dbf80438d948b557c~~	~~Key 1 under the Keys and Endpoint in Azure OpenAI~~

5.7 Troubleshooting

5.7.1 Resetting your environment

If you want to reset your environment, follow these steps:

```
docker compose down
docker compose up
```

Restart Platform and Integration service using the start commands provided earlier.

5.7.2 SSL Handshake Error

MyWave Platform Runtime and Integration service might end up having an SSL Handshake error. You will see an exception in the server logs.

if you are using a CA certificate from a provider like GoDaddy or using a self-signed certificate then the CA certificate has to be imported in to the Java trust store.

1. Please download the X509 certificate from your browser by accessing the domain; for example, in Chrome Browser,
    - View the certificate and go into the details
    - On the Certificate Hierarchy, select the CA certificate (e.g., Go Daddy Secure Certificate Authority - G2)
    - Click the export button to download the CA certificate
    - Select "Base64-encoded ASCII, single certificate" format and click the save button
    - A PEM file will be downloaded
2. Follow the below steps to import the certificate to your trust store

```
sudo keytool -import -alias <yourdomain> -file <certificate>.pem -keystore
<JAVA_HOME>/lib/security/cacerts

Enter keystore password: changeit
Trust this certificate? [no]: yes
```

6 Appendix

6.1 File: docker-compose.yaml

Save the following content into a file called docker-compose.yaml using your preferred code or text editor.

```
version: '3.8'
services:
  postgres-multi-local-hosting:
    image: pgvector/pgvector:pg15
    container_name: mywave-postgres-multi-local-hosting
    ports:
      - "5432:5432"
    environment:
      POSTGRES_PASSWORD: "mywavedev"
      POSTGRES_USER: "mywavedev"
    tty: true
    stdin_open: true
    restart: always
    volumes:
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql

  zookeeper:
    image: zookeeper:3.9.2-jre-17
    container_name: mywave-zookeepr
    ports:
      - "2181:2181"

  kafka:
    image: bitnami/kafka:3.6.2
    container_name: mywave-kafka
    depends_on:
      - zookeeper
    ports:
      - "9092:9092"
    expose:
      - "9093"
    environment:
      KAFKA_ADVERTISED_LISTENERS: INSIDE://kafka:9093,OUTSIDE://localhost:9092
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INSIDE:PLAINTEXT,OUTSIDE:PLAINTEXT
      KAFKA_LISTENERS: INSIDE://0.0.0.0:9093,OUTSIDE://0.0.0.0:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: INSIDE
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_CREATE_TOPICS: "example-topic:1:1"
```

6.2 database creation script: init.sql

Save the following content into a file called **`init.sql`**. Ensure both docker-compose.yaml and init.sql in the same folder for proper database initialization.

```
-- init.sql

CREATE DATABASE mywave_ai WITH OWNER mywavedev;
CREATE DATABASE integrationservice WITH OWNER mywavedev;
```