MyWave — Hosting Guide

A DEVELOPER'S GUIDE FOR HOSTING MYWAVE |

MARCH 2025

Contents

1 Document Details 3
1.1 Document Control 3
2 Introduction 4
2.1 Audience 4
2.2 Purpose 4
2.3 Things to note 4
2.4 Infrastructure overview 5
3 Pre-Deployment Preparation 6
3.1 Azure Open AI subscription 6
3.2 Azure AI Document Intelligence Service 8
3.3 MyWave Wave files 9
3.4 SAP Document Information Extraction (DocX) Subscription 9
4 Files Provided by MyWave 11
5 Setting up SAP Business One (SAP B1) 11
5.1 Prerequisites 11
5.2 Setup instructions 12
5.3 Customise UI wordings 14
5.4 Configuring CSP Settings in SAP B1 15
6 MyWave AI Platform 15
6.1 RDS (Relational Database Service) 15
6.2 Managed Streaming Service (MSS) 16
6.3 Generate JWT key (JSON Web Token) 16
6.4 Generate Integration API key 16
6.5 Health check endpoint 16
7 SAP B1 Integration Service 17
7.1 Prerequisites 17
7.2 Health check endpoint 17
8 Deployment process 17
8.1 Install prerequisites 17
8.2 Copy files 17
8.3 MyWave Platform setup 18
8.4 SAP Business One Integration Service setup 21
8.5 Running the platform and service 22

8.6 Next steps 30
9 Troubleshooting 31
9.1 SSL Handshake Error 31
10 Appendix 31
10.1 AWS example 31
10.2 Glossary 36

1 Document Details

1.1 Document Control

Version	Date	Author	Sign-Off
1.0	20/05/2024	Ollie Hermans	Amy Johnson
1.1	15/07/2024	Vijay A	Jacky Cheung
1.1.1	05/08/2024	Art Pai	Jacky Cheung
1.2	09/09/2024	Vijay S	Jacky Cheung
1.3	14/10/2024	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

Hosters who wish to set up the runtime hosting environment for MyWave solutions.

2.2 Purpose

This document aims to outline the essential steps necessary for configuring and deploying MyWave on a hosting platform.

The guide offers general steps, leaving the choice of specific infrastructure to the host service provider. An example deployment using Amazon Web Services (AWS) is provided here.

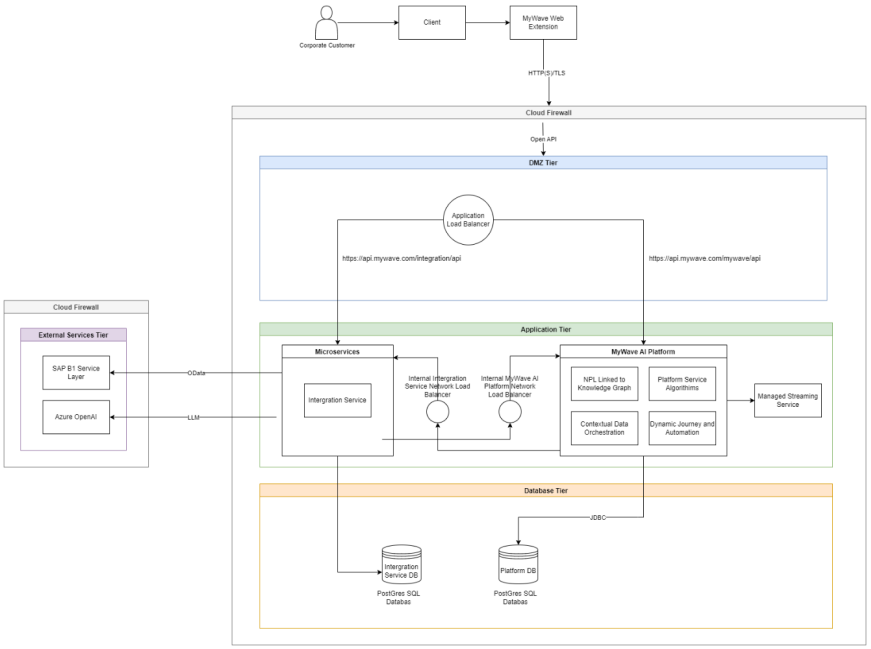
2.3 Things to note

This user guide:

- Is designed to provide generic instructions. However, depending on your choice of infrastructure, steps may differ. Please refer to your chosen infrastructure's documentation for specific instructions.
- Assumes you are using a Linux box. If you are using a different operating system, you will need to tweak the code inputs accordingly.

2.4 Infrastructure overview

This section provides an overview of the infrastructure required for deploying MyWave. It covers the architecture and components involved in the MyWave runtime environment.



An example of this architecture in action using AWS can be found here.

- **DMZ Tier**
  - **Application Load Balancer:** Responsible for routing incoming traffic to different destinations based on specified rules. In this context, it balances traffic between two domain names, directing requests to the MyWave AI platform server (`yourdomain.com/mywave/*`) and the Integration Service (`yourdomain.com/integration/*`).
- **Application Tier**
  - **Integration Service:** Manages the integration between MyWave and external systems or services, such as InfoPlugins and the Azure OpenAI.
  - **MyWave AI Platform:** Serves as the core of the MyWave solution.
  - **Internal Integration Service Load Balancer:** Distributes internal traffic among multiple instances of the Integration Service.
  - **Internal MyWave AI Platform Network Balancer:** Manages internal traffic flow across multiple instances of the MyWave AI Platform.
  - **Streaming Data Service:** Handles real-time data processing and analysis.

- **Database Tier**
  - **Integration DB:** An SQL (Structured query language) database used for storing and managing data related to integration processes within the MyWave ecosystem.
  - **MyWave AI Platform DB:** An SQL database used for storing and managing data related to the core functionalities of the MyWave AI Platform.
- **External Services Tier**
  - **SAP B1 Service Layer**: Provides the interface for integrating with SAP Business One, facilitating communication and data exchange between MyWave and SAP's enterprise resource planning (ERP) system.
  - **Azure OpenAI**: Utilises Microsoft's Azure-based Large Language Models for advanced AI functionalities, enabling natural language processing and other AI-driven services within the MyWave ecosystem.

3 Pre-Deployment Preparation

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 Azure Open AI subscription

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

3.1.1 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 gtp-4o-mini and replace the existing deployment(gtp-35-turbo 0124) with it if you already have it.

3.1.2 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.

3.2 Azure AI Document Intelligence Service

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

3.2.1 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.

3.2.2 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.

3.3 MyWave Wave files

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

3.3.1 Information required by MyWave

- **The Wave File:** This step should have been completed, and the Wave file should have been provided to you by the client.

3.4 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.

3.4.1 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.

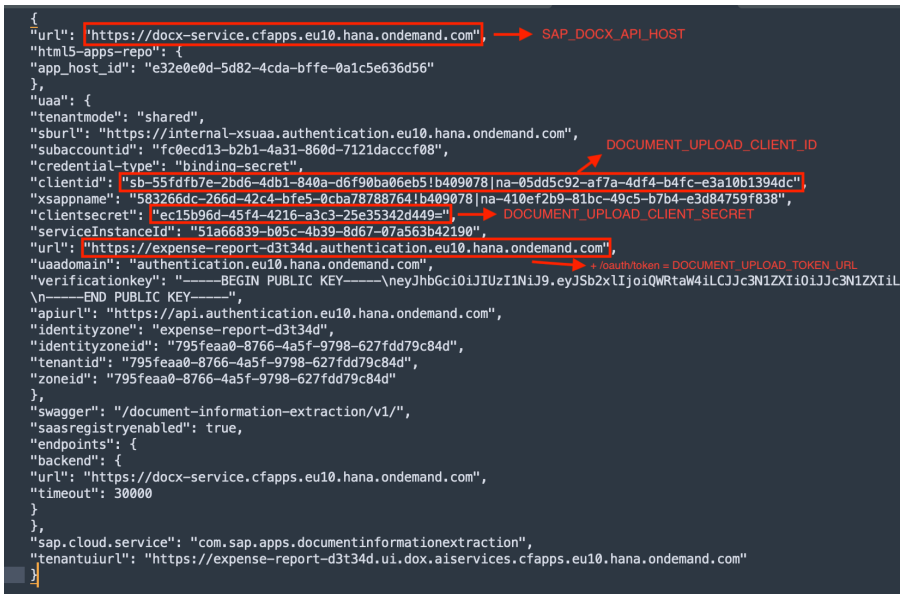
3.4.2 Create New Service Key

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.

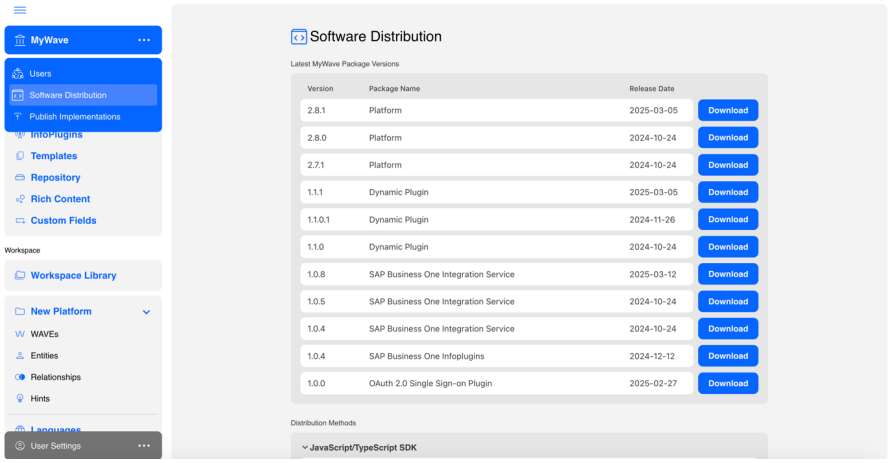
Once the service key has been created, you will see JSON output containing API endpoint URL and User Account and Authentication (UAA) details. Please save this Service Key Json for your reference.

A sample Service Key JSON which has the required details is highlighted below. You will need these values to be used in the Sections 8.5.1, 8.5.2



4 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-2.8.1.zip**
- **sap-business-one-infoplugins-1.0.4.jar:** Java plugins for integration with SAP Business One.
- **sap-business-one-integration-service -1.0.9.jar:** Integration service which provides SAP B1 SSO and LLM capabilities

Deleted: 1

Deleted: 1.1

5 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 an SAP B1 OIDC Provider URL will be needed for the single sign-on (SSO) via integration service.

5.1 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

5.2 Setup instructions

- If you are using a **Windows machine**, please set up your local environment by following the instructions at this link.
- Unzip the source code from the MyWave SAP B1 Web Extension Package (MyWave-SAP-B1-Web-Extension-<version_number>.zip).
- 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
- Set up the following configurations in the environment variable in .env. You can copy the text from .env.example

```
VITE_CLIENT_ID=<your_extension_manager_client_id>
VITE_SLD_ADDRESS=<B1 SLD ADDRESS>/sld/sld0100.svc
VITE_API_URL= https://<loadbalancer-hostname-and-port>/mywave
VITE_LOGIN_URL=https://<loadbalancer-hostname-and-port>/integration/api/sso/login
VITE_REFRESH_TOKEN_URL=https://<loadbalancer-hostname-and-port>/integration/api/ping
VITE_LLM_INTENT_RECOGNISE_ENDPOINT=https://<loadbalancer-hostename-and-
port>/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 5.1 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 5.1 of this guide.
VITE_API_URL	https://<loadbalancer-hostname-and-port>/mywave	The URL of the Mywave Platform Runtime.
VITE_LOGIN_URL	https://<loadbalancer-hostname-and-port>/integration/api/sso/login	SSO Endpoint provided by the Integration Service
VITE_REFRESH_TOKEN_URL	https://<loadbalancer-hostname-and-port>/integration/api/ping	Endpoint to refresh authenticated token
VITE_LLM_INTENT_RECOGNISE_ENDPOINT	https://<loadbalancer-hostename-and-port>/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.

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
```

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

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

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

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

- To install all dependencies, run the following command:

```
pnpm install
```

- 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
```

- This will generate a mywave-app-ext_1.4.0.mtar file in the mta_archives directory at the root of the project.

```
./
├─ dist/
├─ mta_archives/
│  ├─ mywave-app-ext_<version>.mtar
├─ src/
```

- This .mtar file can now be uploaded via your SAP B1 Extension Manager portal and assign it to companies if necessary.
- In case you need to build B1 web extensions for more than one B1 company, build with subtitle provided:

```
pnpm build:ext "my subtitle"
```

- This will generate a `mywave-web-ext-my-subtitle_my_subtitle.mtar` file.
  `my subtitle` will be displayed in SAP B1 Home's UI under extention's name, and mtar's ID will be `mywave-web-ext-my-subtitle`
- Please refer to the Deploying Web Client Extension Guide from SAP to understand how to deploy a mtar file to SAP B1.

5.3 Customise UI wordings

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

There are 5 files under public/locales/en :

- 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.

5.4 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 connect-src and frame-src settings **and not replacing any of your existing settings.** 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: blob:; 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';
```

- <mywave-runtime-host> is the domain name of your MyWave Platform server runtime.
- 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.

6 MyWave AI Platform

6.1 RDS (Relational Database Service)

Create a database and a user on an RDS instance. Use the following placeholders when running the MyWave platform:

- Database username: `<DB_USERNAME>`
- Database user password: `<DB_PASSWORD>`
- Database name: `<DB_NAME>`
- RDS endpoint: `<DB_HOSTNAME>`

6.2 Managed Streaming Service (MSS)

Set up an MSS cluster with 3 brokers across 3 different availability zones. If using Amazon Managed Streaming for Apache Kafka, instructions can be found here.

- Once the cluster is set up, configure topics with these settings:

Topic	Replication Factor	Partitions
DOMAIN_EVENTS	2	100
INFO_PLUGINS	2	100
COMMANDS	2	100

- The 3 topics created above should match the configuration environment variables TOPICS_DOMAIN_EVENTS, TOPICS_INFO_PLUGINS, TOPICS_COMMANDS when running the MyWave platform.
- Replace `<BROKER1>`, `<BROKER2>`, and `<BROKER3>` when running the MyWave platform with the 3 broker endpoints.

6.3 Generate JWT key (JSON Web Token)

- Use the command below to generate a key once for each environment. Reuse the same key for subsequent updates:

```
cat /dev/urandom | LC_ALL=C tr -dc 'a-zA-Z0-9' | fold -w 341 | head -n 1
```

- Replace <JWT_TOKEN_KEY> when running the MyWave platform with the generated key.

6.4 Generate Integration API key

- Use this command to generate a key once for each environment, and reuse it for subsequent updates:

```
cat /dev/urandom | LC_ALL=C tr -dc 'a-zA-Z0-9' | fold -w 128 | head -n 1
```

- Replace <CLIENT_API_KEY> when running the MyWave platform with the generated key.

6.5 Health check endpoint

- The root endpoint (https://yourdomain.com/mywave/api) responds with a JSON body and HTTP Status 200 OK when the server is running smoothly.

6.5.1 Verification

- Use the following command to confirm a HTTP 200 OK response:

```
curl -v https://yourdomain.com/mywave/ping
```

7 SAP B1 Integration Service

The SAP B1 Integration Service will listen on port 8080.

7.1 Prerequisites

7.1.1 RDS (Relational Database Service)

- Create a database and a user on an RDS instance.
- Use the following placeholders when running the integration service:
  - Database username: `<DB_USERNAME>`
  - Database user password: `<DB_PASSWORD>`
  - Database name: `<DB_NAME>`
  - RDS endpoint: `<DB_HOSTNAME>`

7.2 Health check endpoint

- The health check endpoint (https://yourdomain.com/integration/api/health) responds with a JSON body and HTTP Status 200 OK when the server is running smoothly.

7.2.1 Verification

- Use the following command to confirm a successful response:

  ```
  curl -v https://yourdomain.com/integration/api/health
  ```

8 Deployment process

Once you have received the required packages from MyWave and completed all the above prerequisites, you can continue with the deployment.

IMPORTANT: These commands are provided as examples for setting up on RHEL (Red Hat Enterprise Linux) for a customer in Australia.

8.1 Install prerequisites

Run the following commands to install prerequisites:

```
sudo yum -y update
sudo yum -y install tmux
sudo yum -y install zip unzip
sudo yum -y install java-17
```

8.2 Copy files

Copy the following files to the /tmp folder:

- mywave-ai-platform-2.8.1.zip
- dynamic-config-mywave-ai-plugin-1.1.1.1-mywave-ai-platform-2.8.1.zip

8.3 MyWave Platform setup

8.3.1 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.

8.3.2 Installing Platform and dynamic plugin

Run the following commands to install the platform and dynamic plugin:

```
sudo groupadd mywave
sudo useradd -M -s /bin/bash -g mywave -d /home/mywave mywave
sudo mkdir -p /home/mywave/mywave-ai-platform/plugins
sudo unzip /tmp/mywave-ai-platform-2.8.2.zip -d /home/mywave/mywave-ai-platform
sudo unzip /tmp/dynamic-config-mywave-ai-plugin-1.1.2-mywave-ai-platform-2.8.2.zip -d
/home/mywave/mywave-ai-platform/plugins
```

Deleted: 1

Deleted: 1.1

Deleted: 1

8.3.3 Update permissions and ownership

Set permissions and ownership for the platform:

```
sudo chown -R mywave:mywave /home/mywave
sudo chmod g+x /home/mywave/mywave-ai-platform/bin/mywave-ai
sudo chown -R mywave:mywave /home/mywave/mywave-ai-platform/
```

8.3.4 Installing InfoPlugin

Copy sap-business-one-infoplugins-1.0.4.jar to the /tmp/plugins folder.

Execute the following commands:

```
sudo mv /tmp/plugins/sap-business-one-infoplugins-1.0.4.jar /home/mywave/mywave-ai-
platform/plugins
```

8.3.5 Wave deployment

Copy the wave export <wave-export-file-name>.zip file under /tmp folder:

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

8.3.6 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: `/home/mywave/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.

8.3.6.1 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: `/home/mywave/mywave-ai-platform/logs/`

8.3.6.2 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]
}
```

8.4 SAP Business One Integration Service setup

8.4.1 SAP Business One Integration Service

Assuming this will be set up on a different box.

Copy sap-business-one-integration-service-1.0.9.jar file to /tmp/service folder.

Install prerequisites:

```
sudo yum -y update
sudo yum -y install tmux
sudo yum -y install zip unzip
sudo yum -y install java-17
```

Create group and user:

```
sudo groupadd mywave
sudo useradd -M -s /bin/bash -g mywave -d /home/mywave mywave
```

Install SAP Business One Integration Service:

```
sudo mkdir -p /home/mywave/business-one/service
sudo mkdir -p /home/mywave/business-one/service/logs
sudo mv /tmp/service/sap-business-one-integration-service-1.0.9.jar
/home/mywave/business-one/service
sudo mv /tmp/service/conversations /home/mywave/business-one/service
sudo chgrp -R mywave /home/mywave
sudo chown -R mywave /home/mywave/business-one/
sudo chown -R mywave /home/mywave/business-one/service/
sudo chown -R mywave /home/mywave/business-one/service/logs/
```

Copy wave export <wave-export-file-name>.zip file to /tmp folder.

```
sudo unzip /tmp/<wave-export-file-name>.zip -d /home/mywave/business-
one/service/conversations/
```

8.5 Running the platform and service

8.5.1 Running the MyWave Platform

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

Execute the following command (make sure to replace all of the variables):

```
sudo -u mywave \
SERVER_CONTEXT=/mywave \
DEFAULT_LOCALE=en_AU \
DB_USERNAME=<DB_USERNAME> \
DB_PASSWORD=<DB_PASSWORD> \
DB_NAME=<DB_NAME> \
DB_HOST=<DB_HOSTNAME> \
LICENCE_KEY=<LICENCE_KEY> \
JWT_TOKEN_KEY=<JWT_TOKEN_KEY> \
CLIENT_API_KEY=<CLIENT_API_KEY> \
KAFKA_BROKER_ADDRESSES=<BROKER1>,<BROKER2>,<BROKER3> \
TOPICS_DOMAIN_EVENTS=<TOPICS_DOMAIN_EVENTS> \
TOPICS_INFO_PLUGINS=<TOPICS_INFO_PLUGINS > \
TOPICS_COMMANDS=<TOPICS_COMMANDS> \
DYNAMIC_CONVERSATION_CONFIGURATIONS_FOLDER="file:/home/mywave/mywave-ai-
platform/conversations" \
SAP_B1_INTEGRATION_SERVICE_BASE_URL=<SAP_B1_INTEGRATION_SERVICE_BASE_URL> \
SAP_B1_INTEGRATION_SERVICE_GET_ACCESS_TOKEN_URL_PATH="/api/sso/account/token" \
SAP_B1_BASE_URL=<SAP_B1_BASE_URL>/bls/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>" \
tmux new-session -d -s platform /home/mywave/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
KAFKA_BROKER_ADDRESSES	b-1.trial.pucuri.c5.kafka.eu-west-1.amazonaws.com,b-2.trial.pucuri.c5.kafka.eu-west-1.amazonaws.com,b-3.trial.pucuri.c5.kafka.eu-west-1.amazonaws.com	Comma separated Kafka broker addresses.
TOPICS_DOMAIN_EVENTS	DOMAIN_EVENTS	Kafka topic name created for Domain Events
TOPICS_INFO_PLUGINS	INFO_PLUGINS	Kafka topic name created for Info plugin events
TOPICS_COMMANDS	COMMANDS	Kafka topic name created for commands
CLIENT_API_KEY	c6SIrD41F3G39xgICg88wiOTwT3LgWm5PiTqjtIwRF58k6XcbhroKCNqXvqO4pACdKNbdZUMQOfY8um87oqAno0jBWO7dDWm8G4wo5I6Rtz5eoMvKeE3v6Q4x2AVE4Bz	Please use the key generated in **Section 6.4** The CLIENT_API_KEY is shared between Mywave AI Platform and SAP Business One Integration Service.
JWT_TOKEN_KEY	75khazLQmPDCUdZun9uvvEHGfmurZZnfdQi0gbSV1AWOXgm0qT3VxeZV1HSPVSV7ZcQaW0lTu2rnYsXESNw8sFlXH7C3rPwQl9hnJDTOlzAZVGoOgGmJTGA4YLBoSrUGCNblDyqTVIBCOXptBJriPts2XUmNHhkj6XetOSMYa8ne67nsOc7YShJfiCFWdLlAluYFpeicfg36cAAQoddgmRwoL85CS8V9CguBXcmmfqCgnpqqSOAfqB8GjFDxW5WVrbmar6yB46aMPDkQo5wurOutptRgcgCH8xVyRXeNCWFdQ2ZE8eRubgRdDOqpV3Ux8KMwHqWBHjQBqEbSydZvr	Please use the JWT Token generated in **Section 6.3** The JWT_TOKEN_KEY is shared between Mywave AI Platform and SAP Business One Integration Service.
SAP_B1_INTEGRATION_SERVICE_BASE_URL	https://yourdomain/integration	This URL is used to route requests to the SAP B1 integration service. https://<loadbalancer-hostname-and-port>/integration
SAP_B1_BASE_URL	https://your.sapb1.com: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://yourdocx.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://yourdocx-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://yourdocx-demo-6nyshh3n.authentication.eu10.hana.ondemand.com/oauth/token` Please refer to Section 3.3.2 for more details.
DOCUMENT_UPLOAD_CLIENT_ID	sb-5886639f-230f-49d0-8e09-e0c499d77136-1c9f3ecfb2d!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 3.3.2, to know how to create a Service Key Json.
DOCUMENT_UPLOAD_CLIENT_SECRET	d2d9b2c0-4201-4e53-b8ee-65c37\$NwyZJtlZx41c2cBH0lgk=	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 3.3.2, 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/gpt-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)

8.5.2 Webhook Configuration for MyWave Platform (Optional)

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.

8.5.3 Running the Integration Service

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

Execute the following command (make sure to replace all of the variables):

```
sudo -u mywave \
```

```
DB_USERNAME=<DB_USERNAME> \
DB_PASSWORD=<DB_PASSWORD> \
DB_NAME=<DB_NAME> \
DB_HOST=<DB_HOSTNAME> \
DB_PORT=<DB_PORT> \
LOGS_LEVEL= info \
SERVER_PORT=8080 \
SERVER_SERVLET_CONTEXT_PATH=/integration \
PLATFORM_URL=<PLATFORM_URL> \
PLATFORM_JWT_SECRET_TOKEN_KEY=<JWT_TOKEN_KEY> \
PLATFORM_CLIENT_API_KEY=<CLIENT_API_KEY> \
CONVERSATION_CONFIGURATIONS_FOLDER="file:/home/mywave/business-
one/service/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>" \
tmux new-session -d -s business-one-service \
java -jar /home/mywave/business-one/service/sap-business-one-integration-service-
1.0.9.jar
```

Deleted: KNOWLEDGE_SEARCH_TYPES="<KNOWLEDGE_SEARCH_TYPES>" \¶

Example Values for some of the configurations above

Configuration	Example Config	Description
PLATFORM_URL	https://mywave.lb.com/mywave	The Base URL to the platform - https://<mywave-lb-hostname>/mywave
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
PLATFORM_CLIENT_API_KEY	c6SIrD41F3G39xglCg88wiOTwT3LgWm5PiTqjtlwRF58k6XcbhroKCNqXvqO4pACdKNbdZUMQOfY8um87o	Please use the key generated in **Section 6.4**

	qAno0jBWO7dDWm8G4wo5I6Rtz5eoMvKeE3v6Q4x2AVE4Bz	The CLIENT_API_KEY is shared between Mywave AI Platform and SAP Business One Integration Service.
PLATFORM_JWT_SECRET_TOKEN_KEY	75khazLQmPDCUdZun9uvvEHGfmurZZnfdQ1AWOXgm0qT3VxeZVaW0lTu2rnYsXESNw8sFlXH7C9hnJDTOlzAZVGoOgGmNbIDyqTJriPts2XUmNHhkj6Xet7YShJfiCFWdLlAluYFpeicfg36cAAQo9CguBXcmmfqCgGjFDxW5WVrbmar6yB46aMPDCH8xVyRXeNCWFdQ2ZE8eRubgRdDOqpV3Ux8KMwHqWBHjQBqEbSydZvr	Please use the JWT Token generated in **Section 6.3** The JWT_TOKEN_KEY is shared between Mywave AI Platform and SAP Business One Integration Service.
SAP_B1_OIDC_PROVIDER_HOST	https://your.sapb1.oidc: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-84f5-4f8b-9350-892c690546fa	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 5.1 for further details.
SAP_DOCX_API_HOST	your.dox.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 3.3.2 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~~

8.6 Next steps

Once you've completed all the steps outlined above, you can inform the client that their MyWave hosting is now set up and operational.

9 Troubleshooting

9.1 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
```

10 Appendix

10.1 AWS example

The following is an example setup using AWS. Please note that your experience may differ if you are using different infrastructure.

10.1.1 AWS Web Application Firewall (WAF)

AWS WAF serves as the entry point for incoming requests to all components of the MyWave AI Application. It aids in protecting against common application layer vulnerabilities such as SQL injection and XSS attacks. Placing WAF in front of the Application Load Balancer, as proposed in this design, can enhance security. Below is a list of example configurations that may be useful for protecting against common attacks:

- Configuring cross-site scripting match conditions to block traffic.
- Configuring SQL injection match conditions to block SQL attacks.
- Configuring IP match conditions to block requests from known bad IP addresses.
- Configuring Rate-based rules to prevent DoS attacks.

10.1.2 Route53

The primary entry points for deployments are the two public-facing Application Load Balancers backing the MyWave AI Platform Server and Integration Service. The proposed domain names for the production system are as follows:

- MyWave AI Platform: yourdomain.com/mywave/*
- Integration Service: yourdomain.com/integration/*

10.1.3 Network Requirements

The following technical requirements need to be addressed in the network design:

- Ensuring that both the MyWave AI Platform Server and Integration Service can send requests to each other.
- The MyWave AI Platform uses information in the HTTP request to generate URLs in the response. Therefore, requests received by the MyWave AI Platform server must include uniform information about its host and protocol, even when the request is from the internal network. In other words, both the front-end client and Integration service need to use the same hostname and protocol when sending requests to the MyWave AI Platform.

As depicted in the Infrastructure Network Diagram, both the MyWave AI Platform and Integration Service will be fronted by an additional set of internal Network Load Balancers. These load balancers will use the same hostname as the public-facing load balancers.

This can be achieved using AWS Route53, which needs to be configured with split-view DNS, also known as split-horizon DNS. In AWS, this is accomplished by creating a private hosted zone with the same name as the public hosted zone and associating the VPC with the private hosted zone.

This way, DNS queries respond with answers based on the source of the request. From within the VPC, answers come from the private hosted zone, while public queries return answers from the public hosted zone.

10.1.4 VPC

The MyWave AI Platform is deployed in a VPC in the Australia Region, consisting of 1 public subnet and 2 private subnets.

Subnet	Type
DMZ Tier	Public
App Tier	Private
Database Tier	Private

For high availability, it's recommended to mirror these subnets in other Availability Zones.

The VPC will require the following additional configurations:

1. Association with a Private Hosted Zone in Route53.
2. DNS resolution enabled.
3. DNS hostname enabled.

10.1.5 DMZ Tier - Public Subnets

This subnet comprises 2 Application Load Balancers for the MyWave AI Platform and the Integration Service. The load balancers will be configured using host names of both the MyWave AI Platform and Integration Service with HTTPS listeners and EV Certificates procured from an authorized CA.

10.1.5.1 Application Load Balancer (ALB) for MyWave AI Platform

The ALB will listen on the HTTPS port. The listener will be configured with a Forward Action rule with one target group attached to the Platform Auto Scaling group. The forwarding rule will be very specific, ensuring that requests are forwarded only when rules match. This prevents attacks like DoS. Since MyWave AI deployment does not require the ALB to use any AWS services, no Identity and access management (IAM) roles are required to be assigned to it.

The security group for the MyWave AI Platform ALB:

Type	Port Range	Source
HTTPS	443	AWS WAF IP range for ap-southeast-1

It's essential that the ALB only allows external inbound traffic from AWS WAF.

10.1.5.2 ALB for Integration Service

The ALB will also listen on the HTTPS port and be configured with a Forward Action rule. This rule will have one target group attached to the Integration Service Auto Scaling group. Like the MyWave AI Platform ALB, this configuration aims to ignore any requests that do not match the rules, preventing attacks like DoS. No IAM roles are required for the Integration Service deployment.

The security group for the Integration Service ALB:

Type	Port Range	Source
HTTPS	443	AWS WAF IP range for ap-southeast-1

It's essential that the ALB only allows external inbound traffic from AWS WAF.

The ALB will also listen on the HTTPS port and be configured with a Forward Action rule. This rule will have one target group attached to the Integration Service Auto Scaling group. Like the MyWave AI Platform ALB, this configuration aims to ignore any requests that do not match the rules, preventing attacks like DOS. No IAM roles are required for Integration Service deployment.

The security group for the Integration Service ALB:

Type	Port Range	Source
HTTPS	443	AWS WAF IP range for ap-southeast-1

It's essential that ALB only allows external inbound traffic from AWS WAF.

10.1.6 Application Tier – Private Subnet

This subnet will contain 3 classes of servers:

- Auto Scaling Group of MyWave AI Platform Server
- Amazon Managed Streaming for Apache Kafka (MSK)
- Auto Scaling Group of Integration Server instances

10.1.6.1 MyWave AI Platform Server

MyWave AI Platform Server is a Java-based application that can run standalone without the need for installing a traditional servlet container.

Required Software Components:

Requirement	Version
OpenJDK	17.0.9 or above

MyWave AI Platform Server Security Group Inbound Rules:

Type	Port Range	Source
HTTPS	8080	CIDR of the VPC

MyWave AI Platform Server does not require access to any AWS Services and therefore does not require any IAM role for its functioning. However, depending on the deployment, there might be a need to assign IAM roles for operational and automation needs, such as read access to S3 for server configurations.

10.1.6.2 Integration Service

Integration Service is a Java-based Spring Boot application that can run standalone without the need for installing a traditional servlet container.

Required Software Components:

Requirement	Version
OpenJDK	17.0.9 or above

Integration Service Security Group Inbound Rules:

Type	Port Range	Source

HTTPS	8080	CIDR of the VPC

10.1.7 Application internal messaging

MyWave AI Platform server relies on Apache Kafka for messaging, compatible with both self-managed and Amazon managed Kafka models.

Self-Managed Kafka Cluster vs Amazon MSK:

It is recommended to use Amazon's MSK instead of a self-managed Kafka cluster due to the managed features provided by MSK, such as easier scaling and automatic recovery and patching.

The minimum Amazon MSK cluster size is 2; however, it requires at least 3 nodes to ensure message delivery during a single node failure event.

Kafka Security Group Inbound Rules:

Type	Port Range	Source
Custom TCP	9092	MyWave AI Platform Security Group
Custom TCP	2181	CIDR of the App Tier Subnet

The Kafka broker does not require access to any AWS Services and therefore does not require any IAM role for its functioning. However, depending on the deployment, there might be a need to assign IAM roles for operational and automation needs.

10.1.8 Database Tier – Private Subnet

This subnet is used to hold 1 class of data store:

- RDS instance for both MyWave Platform Server and Integration Service

RDS Postgres:

A single RDS instance will suffice for hosting databases for both MyWave AI Platform Server and Integration Service.

The supported version of Postgres is 15.

RDS will not be available to the internet and will require a private subnet group consisting of the DB tier subnet in at least 2 availability zones for production.

Once the database is created, MyWave AI Platform Server and Integration server can manage their respective schema on Postgres. A User will have to be created with permission to create tables, and the credentials need to be provided to MyWave AI Platform server and Integration server as part of their configuration.

Security group for RDS:

Type	Port Range	Source
Custom Port	5432	Security Group of Platform, Security group of Integration Server, and Security group of Jumphost (for management)

10.2 Glossary

AWS: Amazon Web Services

IAM: Identity and access management

JWT Key: JSON Web Token Key

LLM: Large Language Models

MSK: Amazon Managed Streaming for Apache Kafka

MSS: Managed Streaming Service

OIDC: OpenID Connect

RDS: Relational Database Service

RHEL: Red Hat Enterprise Linux

SAP B1: SAP Business One

SSO: Single Sign-on

SQL: Structured query language