ExxonMobil

EMC2PDFA V3

PIPELINE DUCTILE FRACTURE ANALYSIS

Administrator Manual

v1.1

Table of Content

Important Notes

Licensing

Please make sure that you have the proper license to deploy EMC2PDFA V3 in your environment.

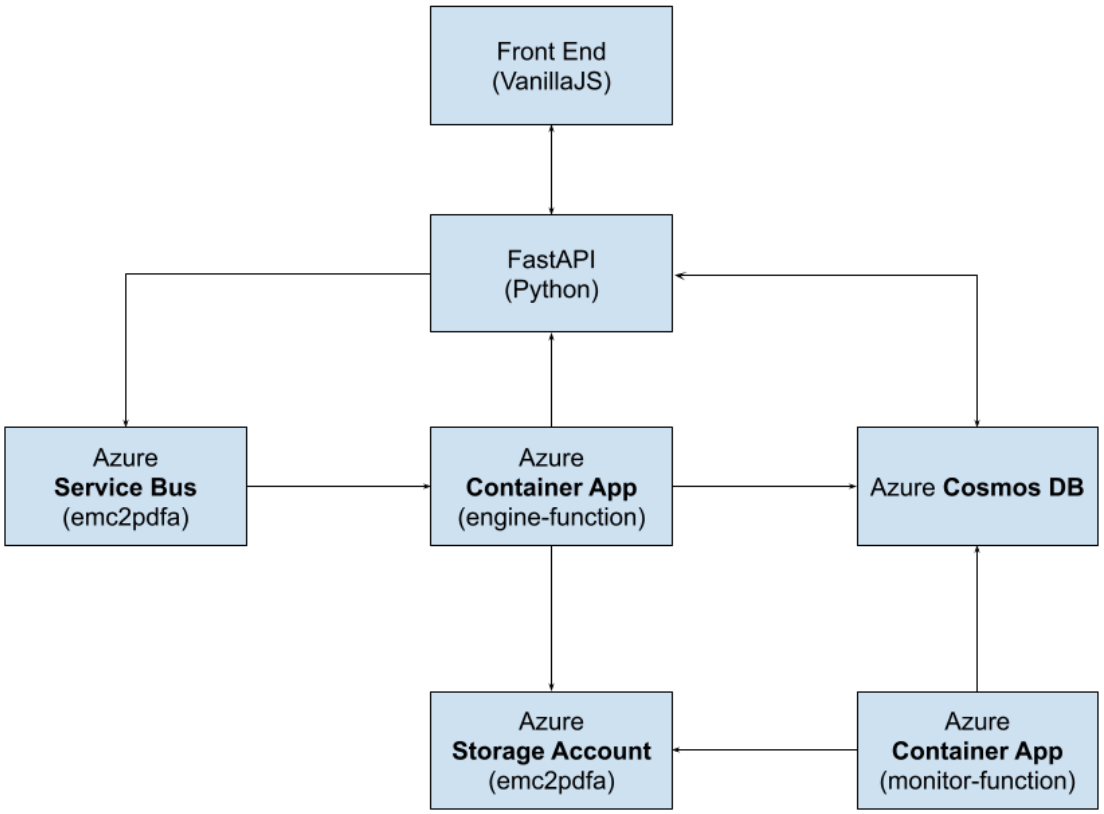
Target Audience

The following document was created for an **Azure Admin** which is about to deploy **EMC2PDFA V3** on their **Azure Environment**.

As such, you are expected to have a basic understanding of Azure Native Services, and possess the min required permissions to set up such services.

High Level Architecture Diagram

The following diagram shows the required network connectivity between the user and the relevant services. Any user/consumer will need to be able to access the **Front End**, and **FastAPI** services. Currently the user/consumer will access these either via the front-end, and/or directly, as these services are REST API which allows for great flexibility when interacting with them.



Services and Ports Mappings

#	From	To	Port(s)
1	Any	Frontend	HTTPS/443
2	Any	FastAPI	HTTPS/443
4	FastAPI	Azure Service Bus	Azure Service (Encrypted)
5	FastAPI	Azure CosmosDB	Azure Service (Encrypted)
6	Engine Function/Monitor	Azure CosmosDB	Azure Service (Encrypted)
7	Azure Container App	Azure Storage Account	Azure Service (Encrypted)
8	Engine Function/Monitor	FastAPI	HTTPS/443
9			

Deployment Methods

The EMC2PDFA V3 can be deployed in multiple forms. Since you'll be provided with Docker containers (AMD64) you can use Azure VM, Azure Container Services, Azure App Services, etc. As long as the environment variable and networking is configured properly, EMC2PDFA V3 will function as intended.

Docker Images

The different services are available as Docker Images, and are being hosted by EMC2 in their Azure Container Registry. You can either copy the images to your own registry and deploy from there, or use them directly from EMC2 Container Registry.

A special account (identity) was created to allow users to pull the images, these credentials will be provided to your team, please contact support and they'll be provided to you.

Microservices Features and Functionalities

Frontend (React/NextJS/VanillaJS)

Our frontend website has 4 main functionalities

- **Listing Batches** and related Tasks
- **Creating New Batches** (Data entry)
- Documentation, Support & Debugging
- User Authentication

The frontend is executing HTTP Calls (client/server side) to our main REST API Service.

FastAPI Service (Python / FastAPI)

Our fastapi-service is exposing several endpoints (available via the Swagger UI), mostly to communicate with the DB (list/create batches/tasks), and add a Batch/Task to the Queue.

Another important functionality is **queue management** (via Azure Service Bus)

Refprop Engine (Python / Container App)

This is our main entry to the REFPROPdll, which is developed by NIST. Our refprop-engine is written in Python and in its heart it is utilizing REFPROPdll for Entropy and Density related

calculations. Additional EMC2 proprietary methods / functions are being executed which holds 85% of the code. Our code is being triggered by Azure Service Bus Queue. Azure Service Bus will trigger this Container App Function.

Azure Service Bus

This is our Service Bus / Message Queue. Tasks are being pushed to this queue here by the fastapi-service , which will trigger the refprop-engine (function) for processing.

CosmosDB

Our only database, it holds the **Batch**, **Tasks, Configuration, Monitoring, PreSets Collections**. A Batch will have a list of tasks IDs, and a **Task** will have a batch_id, this is how they are connected.

Microservices Data Flow

Please see the User Manual for additional details on INPUTS and COMPOSITIONS.

Creating a New Batch

1. A User creates a New Batch by providing **INPUTS** and **COMPOSITIONS** in the **Create Batch Form**.
    a. A few utility endpoints are being used in order to populate compositions and tasks (data), these utilities are being served by the fastapi-services.
2. Once the form is submitted, the final request payload (batch info + related tasks) will be posted to fastapi-services
    a. A few new sets of records (batch/tasks) will be created in the DB
3. The refprop-service will get triggered by the fastapi-service, which will pull the batch details from the DB
    a. It'll then verify the batch exists, and push all of its tasks to the queue
4. The refprop-engine (serviced by Dramatiq) will pick up the tasks from the queue and will process them. (max of 5 tasks at the same time)
    a. The batch/task(s) status will get updated as they are being processed.

Listing Batches

Our **Home Page** (aka Dashboard) among other links to useful resources, will be listing all of the Batches in the DB, and it'll be utilizing the fastapi-services to do so.

Microservices Specific (minimum) Requirements

FRONT-END

- **Environment:** ReactJS based web application
- **OS:** rtsp/lighttpd based Docker Image
- **Size:** Standard B2s, 2 vCPU, 4GB RAM

FASTPI-SERVICES

- **Environment:** FastAPI based REST API
- **OS:** python:3.11-slim based Docker Image
- **Size:** 2 vCPU, 4GB RAM

REFPROP-ENGINE-FUNCTION

This is an Azure Function (container App). The more resources that will be available to this function, the more tasks it can run at-a-time. For now, it'll process 4 tasks at-a-time. In ExxonMobile env, it's currently processing ~40 tasks every 5 seconds (ExxonMobil have autoscalling enabled).

- **Environment:** Python based Processing Engine
- **OS:** ubuntu:22.04 based Docker Image
- **Size:** 4 vCPU, 4GB RAM

Troubleshooting

The flow of data and the interaction between the services is pretty simple.

Issue #01 - Batch is NOT being processed

Based on our testing, if and when things do not go/work as expected, it's either the refprop-service and/or the refprop-engine which are not running, and even then, most of the time it's the refprop-engine which at times get overloaded if under provisioned.

Solution: The easiest solution is to restart the service, it'll recover and restart from the last point.

Issue #02 - Batch is RUNNING, tasks are FAILING

In such a scenario, it's probably an "edge use-case" and/or an issue with parsing the provided inputs. These 2 scenarios will lead to calculations being "off" (division by zero, etc) and will error out in the code.

Solution: Contact support, and provide them with the inputs/compositions being used, and logs (if possible). The Support team will try to reproduce the issue locally.

In most cases, a new version of the code will be provided to you which includes a fix/update to the reported issue.

Debugging

In general only the main/complex components are writing to logs. The refprop-engine, which is our main and most complex component, writes to logs. **Currently logs are written to the console**, which means you need access to the host/service in order to view the logs.

Appendix

Environment Variables (WIP)

Please see attached txt files for a complete list.

Azure Container Registry

- An Azure identity needs to be used as part of the authentication. Please contact support for credentials.
- With the credentials, you'll receive the names of the containers to be used.