Hardware and Software Requirements for DMFAS 7

DMFAS Programme

As of May 2024

Non official document

Table of Contents

1 Introduction

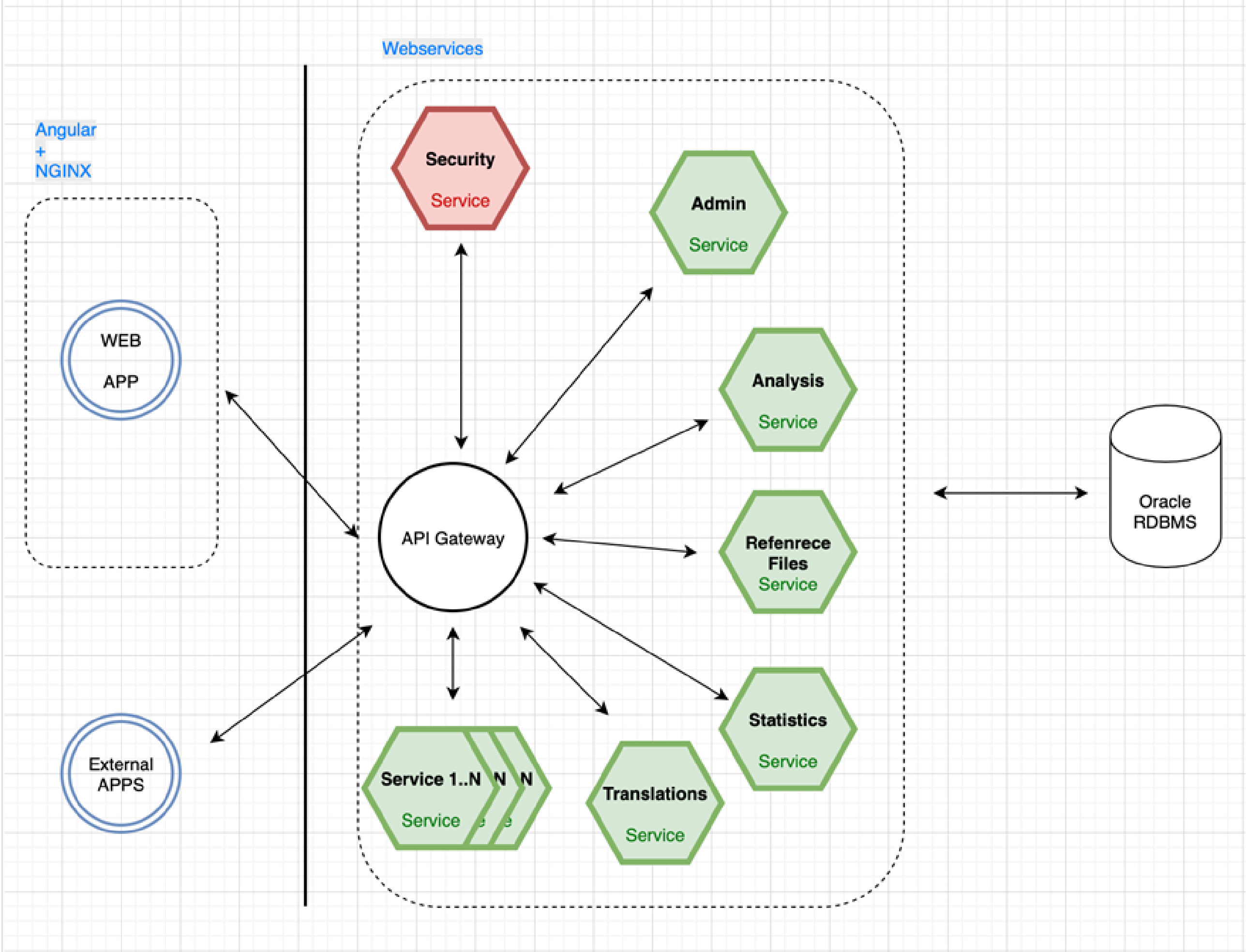
The purpose of this document is to describe installation options for DMFAS 7 and the minimum recommended hardware and software requirements to run the application.

This document will be updated whenever necessary to consider changes and developments in information technology. Considering that DMFAS 7 is continuously evolving, some hardware or software requirements may change over time. Nevertheless, the DMFAS Programme will do as much as possible to maintain the foreseen hardware and software platform.

The precise requirements for each institution to run DMFAS 7 will depend on the specific situation of the institution (number of users, number of debt instruments, available connections etc.). Therefore, it is important to contact the DMFAS Programme before ordering equipment or software to ensure that the latest and most appropriate specifications are met.

2 Architecture Overview

The application is divided into three component groups, the persistence layer, comprised of the Database, the business layer or Backend composed of a collection of Java web services and the Frontend (presentation layer) which is an Angular Web Application.The following diagram depicts the high-level architecture of the application and its components.



2.1 Backend

The Backend architecture is based on the "separation of concerns" principle, which states that every component should address one concern, so it is implemented by a collection of microservices. All of these microservices are independent of each other and communication between them happens using REST services. This has the following advantages:

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities

Since all services are independent the actual microservices can be written in any programming language, it was decided that Java would be the Development Language of choice for the first round of services using Spring Boot (https://spring.io/projects/spring-boot+) as the development stack.

2.2 API Gateway

The API gateway is the main glue that connects all the microservices together, all the traffic, either internal (from microservice to microservice) or external (from the web app or other applications to the microservices) will flow through the API, this allows for an extra layer of security being that the only exposed service is the API service. The gateway will publish the available routes and it will validate the required security attributes before the request reaches the backend service.

2.3 Frontend

The Frontend is developed using Angular (http://angular.io[1]) as the default presentation language. Furthermore, it was originally decided that Primefaces would be the component library of choice (https://www.primefaces.org/primeng/), this component library has more than 80 native widgets and more than 50 user components that will facilitate the development of the frontend. The Angular application is divided in 3 main modules:

Module Name	Description
core	holds core interceptors, services (http to the api-gateway, and other core local services), guards (authorization and authentication, etc.), constants, entity models and DTOs and some utility functions
shared	holds all shared components, directives, guards and common validators. These components are used on all the different modules, simplify development and help maintain a single look & feel across the whole application
main application	The main application holding the login page, the main layout component, and all the modules that together make the application. These are lazy loading modules.

[1] http://angular.io/

3 Software Requirements

3.1 Database Server

In general, the DMFAS system runs on every Operating System platform compatible with the latest Oracle release:

Oracle Release	**Operating System**	
19c	MS WS 2019 or later	Linux/Unix/Solaris/AIX

3.2 Services

All services, including the Backend services, the Frontend service and the API Gateway, are deployed as Containers, the deployment will need an internet connection to pull the containers from the UNCTAD container registry that holds all versions of the services.

3.2.1 Supported Installation Options

There are mainly two installation options, one simply uses Podman and Podman-Compose to install all the services in any machine running Podman, the other uses any Container Orchestration Tool, the sizing in this guide is based on Kubernetes and K3s because they are the recommended tools, but any orchestration tool will work (e.g. RedHat OpenShift, Nomad, etc.).

3.2.1.1 Podman Compose

This is the smallest option in terms of sizing and only requires one Server to run all the backend and frontend services, keep in mind that the database should always have it's on server. Installing using Podman is recommended only for Windows OS installations.

Podman Version	**Operating System**	
>= 4.8.0	MS WS 2019 or later	Linux/Unix/Solaris/AIX

The following distributions and architectures are currently supported for Podman installation

Platform	**x86_64 / amd64**	**arm64 / aarch64**	**arm (32-bit)**	**s390x**
CentOS[2]	✓	✓		
Debian[3]	✓	✓	✓	

[2] https://docs.docker.com/engine/install/centos/
[3] https://docs.docker.com/engine/install/debian/

Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	s390x
Fedora[4]	✓	✓		
Raspbian[5]			✓	
RHEL[6]				✓
SLES[7]				✓
Ubuntu[8]	✓	✓	✓	✓
Windows[9]	✓			

Additional Storage

Some additional storage might be required in order to store persistent data, such as attachments. This storage will be mapped using an external Volume to Podman and must be on a shared drive.

High Availability

There is no way of achieving high availability with Podman on Windows. In order to achieve high availability Docker should be used along with Docker Swarm that must be installed and running on at least two servers, this would increase the required number of server to at least 2, it is possible to use the Database Server as part of the Docker Swarm but this will require an additional 8GB of RAM and an additional vCPU on the Database server in order to accommodate the new workload. This setup will need to be licensed.

Software Requirements

The following table lists the required software that must be installed in order to be able to complete the Application installation. Source

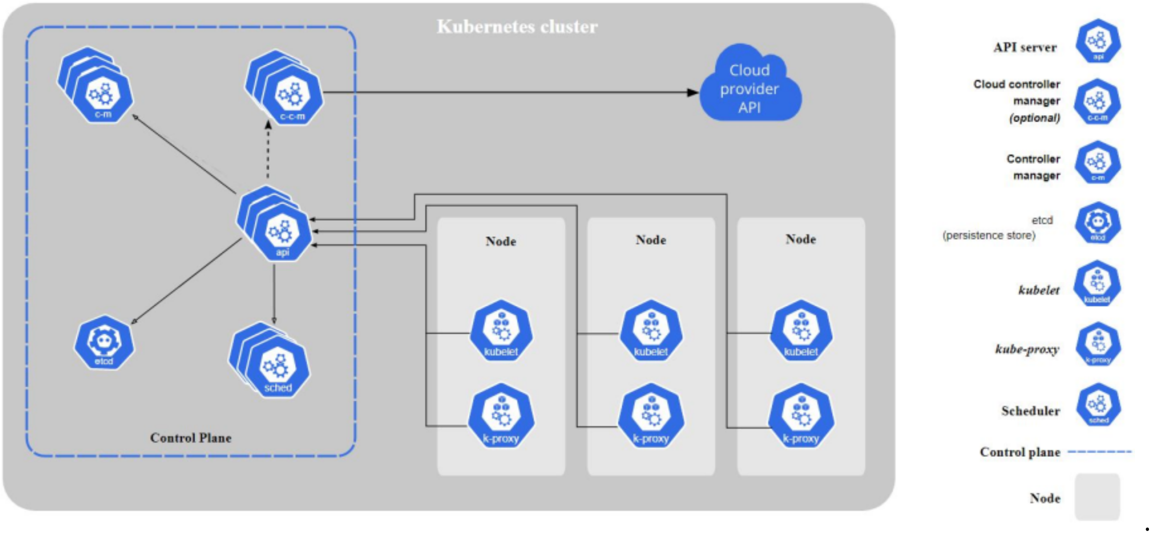
Software	Version	Description	
Podman	>= 4.8	Podman engine	https://podman.io/docs/installation

[4] https://docs.docker.com/engine/install/fedora/
[5] https://docs.docker.com/engine/install/debian/
[6] https://docs.docker.com/engine/install/rhel/
[7] https://docs.docker.com/engine/install/sles/
[8] https://docs.docker.com/engine/install/ubuntu/
[9] https://docs.docker.com/engine/install/binaries/

Software	Version	Description	
git	latest		https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

3.2.1.2 Kubernetes

For all Linux installations it is recommended that Kubernetes be used to deploy all the services, Kubernetes[10] is an open-source container orchestration platform for managing, automating, and scaling containerized applications, Kubernetes is the de facto standard for container orchestration because of its greater flexibility and scaling capabilities
The following diagram depicts the basic architecture of a Kubernetes cluster.



source: https://kubernetes.io/docs/concepts/overview/components/

Each cluster is formed by one or more servers that hold the Control Plane Components (api server / etcd / scheduler / ….) and one or more servers that hold the running pods (Node Servers).
Each node server can hold a given number of pods, this number is determined by the amount of available Memory and Cores of each individual server. There are also some hard limits to the amount of pods that a server can run imposed by the Kubernetes architecture, but these numbers are high enough (no more than 110 pods per node, no more than 5000 nodes per cluster, etc.), so that they will not impact on the architecture of DMFAS 7.
If high availability is a requirement there should be at least two Control Plane servers and at least two Nodes for running pods in different zones, this ensures that on failure the pods from the other nodes will take the load.

Kubernetes Alternative Installation

K3s[11] is a Lightweight Kubernetes distribution that is configured to run on a single server. This is the recommended distribution for all smaller installations that would not require a full Kubernetes installation (2+servers), this distribution is capable of supporting all the same components as the Kubernetes installation but it can run on a single server. High availability and scaling can be achieved by simply adding nodes to the K3s cluster.

[10] https://www.dynatrace.com/news/blog/what-is-kubernetes-2/
[11] https://k3s.io

Additional Storage

There is a number of options in terms of provisioning Storage Classes supported by Kubernetes, the selection of the provisioner is based on local expertise and preference, here[12] (https://kubernetes.io/docs/concepts/storage/storage-classes/#nfs) is a list of supported adapters.
When using K3s in a single instance K3s "Local Storage" should be used along with a backup strategy.

Software

The following table lists the required software that must be installed in order to be able to complete the Application installation.
Podman Installations:

Software	Version	Description	
Podman	>= 4.8	Podman Engine	https://podman.io/docs/installation

Kubernetes / K3S installation

Software	Version	Description	
kubectl		Kubernetes administrator	https://kubernetes.io/releases/download/
Kubernetes	>= 1.27	At least one master and one worker node should be installed and configured	
K3s	>= v1.27		https://k3s.io/
cert-manager	latest	Kubernetes Certificate manager used to manage and update digital certificates. This should be installed in the Kubernetes container and is only required if automated certificate management is a requirement.	https://cert-manager.io/docs/installation/
helm	>= 3.9.x	Helm charts are used to install all DMFAS components to the Kubernetes cluster.	https://helm.sh/docs/intro/install/

[12] https://kubernetes.io/docs/concepts/storage/storage-classes/#nfs

3.3 Other Requirements

3.3.1 Internet Access

While having internet access from the application servers is not a must it is highly recommended that the Server has either direct access or access via a Proxy to the internet in order to be able to fully install the servers with the required software without the need to download everything to a local registry in order to mirror the public and private registries used to install the software.

3.3.2 Reverse Proxy / Load Balancer

There are a few ways that the network topology can be configured in order to enforce security and it is up to the installing country to determine the best practices, but, if the application is to be accessed from the Internet we recommend the use of a Reverse Proxy or Load Balancer in the DMZ to receive all connections to the domain and route accordingly, this proxy should receive all incoming request and forward them to the application Ingress controller which in turn will serve the content using the correct service.

3.3.3 HTTPS

To safeguard the confidentiality and integrity of sensitive data exchanged between clients and servers, the implementation of HTTPS protocol is strongly recommended. This measure is essential to protect sensitive information, including passwords and confidential debt-related data, from potential vulnerabilities.

To enable HTTPS, a valid SSL certificate is required. This certificate can be obtained through two primary methods:

1. Provided by the installing Country:
    - The installing country may furnish a pre-existing certificate for utilization.
2. Generated and configured during Installation:
    - The certificate can be generated and configured during the installation process, ensuring seamless integration. Depending on the installation platform this can even be handled automatically by the platform using a plugin.

Certificate Options:

- Global Catch-All Certificate: This certificate covers all subdomains within a domain, offering a comprehensive solution.
- App-Specific Certificate: This certificate safeguards a specific application url, providing a more granular approach to security.

Certificate Authority Selection:

- Preferred Authority: Installing countries have the discretion to utilize their preferred certificate authority.
- Let's Encrypt: In the absence of a specified authority, Let's Encrypt (https://letsencrypt.org[13]) offers a reputable and accessible option. The application's "Cert Manager" plugin can facilitate the creation and management of certificates through Let's Encrypt.

DNS Access requirement:

- While having access to modify the DNS server is not mandatory for the application to work it is necessary to acquire an SSL certificate. This process verifies ownership of the public IP address.

[13] https://letsencrypt.org/

Self-Signed Certificates:

- While self-signed certificates can function and give some security, their use is generally discouraged due to compatibility concerns with modern browsers. The potential for future browser rejection highlights the importance of utilizing valid certificates.

Reverse proxy:

- Adding a reverse proxy between clients and servers, providing additional security features, caching, and SSL termination, in general a generic DNS (*.domain.xxx) entry pointing to this reverse proxy will be enough to create the SSL certificates needed.

DMFAS prioritizes unwavering data protection and strongly advocates for the adoption of HTTPS with valid SSL certificates to ensure the highest level of security for sensitive information.

3.3.4 SMTP

All crucial notifications and password management functionalities rely on email as the primary communication channel. Therefore, it's essential for all users to have a valid and accessible email address.

For installations within closed intranets lacking access to an external SMTP server, a dedicated private SMTP server will be deployed during setup. This internal server's functionality will be limited to facilitating login configuration and notifications within the private domain, restricting external email sending and receiving. This email server will be installed in the server machine in one of 3 ways:

- If installing the application Podman or Docker then the smtp server will be installed as a Pod on that same server.
- If installing the application in a multi node Kubernetes platform then it is recommended to use Podman as the installation method in any of the platform servers, or on a dedicated server since installing as a deployment in the cluster would require specifying a specific worker node in whitch to deploy the application (because of port mapping issues) and this is not good practice.
- If installing in a single node Kubernetes platform (e.g.: K3s) it can be installed in this application and use a special type of service to map the SMTP and IMAP ports to give access to external users to the application.

Since the provided SMTP server is used as a workaround in closed intranets it will not have SSL security in place nor password management, so it is highly encouraged for the countries to provide a valid email server.

3.4 Hardware Requirements

3.4.1 Database Server

Additional memory, processor speed and disk space might be needed if other software is installed. Hardware specifications for servers running another network operating system should have equal performance and capacity and must be compatible with Oracle RDBMS. It Is recommended to have an automatic detection software for monitoring updates and proactive support notifications.

Component	Specifications
CPU[1]	Intel® / Core i(7/9) 9th to 11th Gen, 4 or more GHz Series Processor

Component	Specifications
Hard disk	3 x 480 GB or more SATA, NVMe, SSD or SAS 15K rpm with data striping through RAID X (Mixed drive types allowed matching type/speed/ capacity)
DVD-ROM	DVD+/-RW
RAID/Internal Controller	PERC Hx
Memory	From 32 GB up to 128 GB DIMM/LRDIMM/RDIMM
Screen/Video adapter	17" Flat Panel with integrated video card with 1GB or more
Backup streamer	Digital tape streamer with the same capacity as the total disk space
NIC	1 or more network card(s) supported by the network installed with capacity of100/1000 Mbps for best performance
Computer in general	If the server is a stand-alone computer, it could be a tower model or suitable for placement in a rack. Both, with a scalable internal capacity and flexibility to adapt to changing workload conditions. Local technical support is strongly recommended.

[1] Any 100% Intel compatible processor, such as AMD can be used as well, given that they deliver equal or better performance.

3.4.2 Podman Installation

The following table represents the minimum requirements for a Production Server, note that it is recommended that the Database be in a separate server.

Component	Specifications
CPU Count	Minimum: 4; Recommended 6+. CPU Architecture should be chosen according to the table above.
Hard disk	1. 3 x 100MB or more SSD or SAS SSD with data striping through RAID Xa
Memory	From 16 GB up to 128 GB DIMM/LRDIMM/RDIMM

Component	Specifications
NIC	1 or more network card(s) supported by the network installed with capacity of100/1000 Mbps for best performance
Computer in general	If the server is a stand-alone computer, it could be a tower model or suitable for placement in a rack. Both, with a scalable internal capacity and flexibility to adapt to changing workload conditions. Local technical support is strongly recommended.

3.4.3 Kubernetes Cluster

Control Plane

Component	Specifications
CPU Count	Minimum: 4+. CPU Architecture should be chosen according to the table above.
Hard disk	1. 3 x 200 MB or more SSD or SAS SSD with data striping through RAID Xa
Memory	From 16 GB up to 128 GB DIMM/LRDIMM/RDIMM
NIC	1 or more network card(s) supported by the network installed with capacity of100/1000 Mbps for best performance
Computer in general	If the server is a stand-alone computer, it could be a tower model or suitable for placement in a rack. Both, with a scalable internal capacity and flexibility to adapt to changing workload conditions. Local technical support is strongly recommended.

Node Server

Component	Specifications
CPU Count	Minimum: 4+. CPU Architecture should be chosen according to the table above.
Hard disk	3 x 200 MB or more SSD or SAS SSD with data striping through RAID Xa.
Memory	From 16 GB up to 128 GB DIMM/LRDIMM/RDIMM

Component	Specifications
NIC	1 or more network card(s) supported by the network installed with capacity of100/1000 Mbps for best performance
Computer in general	If the server is a stand-alone computer, it could be a tower model or suitable for placement in a rack. Both, with a scalable internal capacity and flexibility to adapt to changing workload conditions. Local technical support is strongly recommended.

3.4.4 **Single Instance Server**

This is used in small instances where the application is installed on a single server

Component	Specifications
CPU Count	Minimum: 8+. CPU Architecture should be chosen according to the table above.
Hard disk	3 x 300 MB or more SSD or SAS SSD with data striping through RAID Xa.
Memory	From 24 GB up to 128 GB DIMM/LRDIMM/RDIMM
NIC	1 or more network card(s) supported by the network installed with capacity of100/1000 Mbps for best performance
Computer in general	If the server is a stand-alone computer, it could be a tower model or suitable for placement in a rack. Both, with a scalable internal capacity and flexibility to adapt to changing workload conditions. Local technical support is strongly recommended.