Deployment Guide

PREPARED FOR

Table of Contents

[Description 3](#_Toc46308876)

[Istishara Software Architecture 4](#_Toc46308877)

[Software Requirements 5](#_Toc46308878)

[1. Infrastructure 5](#_Toc46308879)

[a. Docker 5](#_Toc46308880)

[b. Docker-compose 5](#_Toc46308881)

[2. Web Application 5](#_Toc46308882)

[a. Angular 5](#_Toc46308883)

[b. Nginx 6](#_Toc46308884)

[3. Authentication and Authorization 6](#_Toc46308885)

[a. Keycloak 6](#_Toc46308886)

[b. LDAP System (Optional, not required at this stage) 6](#_Toc46308887)

[4. API Services 6](#_Toc46308888)

[a. Python 6](#_Toc46308889)

[b. FastAPI 6](#_Toc46308890)

[5. Storage 7](#_Toc46308891)

[a. MongoDB 7](#_Toc46308892)

[b. PostgreSQL 7](#_Toc46308893)

[6. Licenses 7](#_Toc46308894)

[Hardware Requirements 8](#_Toc46308895)

[1. Operating System 8](#_Toc46308896)

[Istishara Deployment 9](#_Toc46308897)

[1. Install Docker and Docker-Compose 9](#_Toc46308898)

[2. Build the Solution 9](#_Toc46308899)

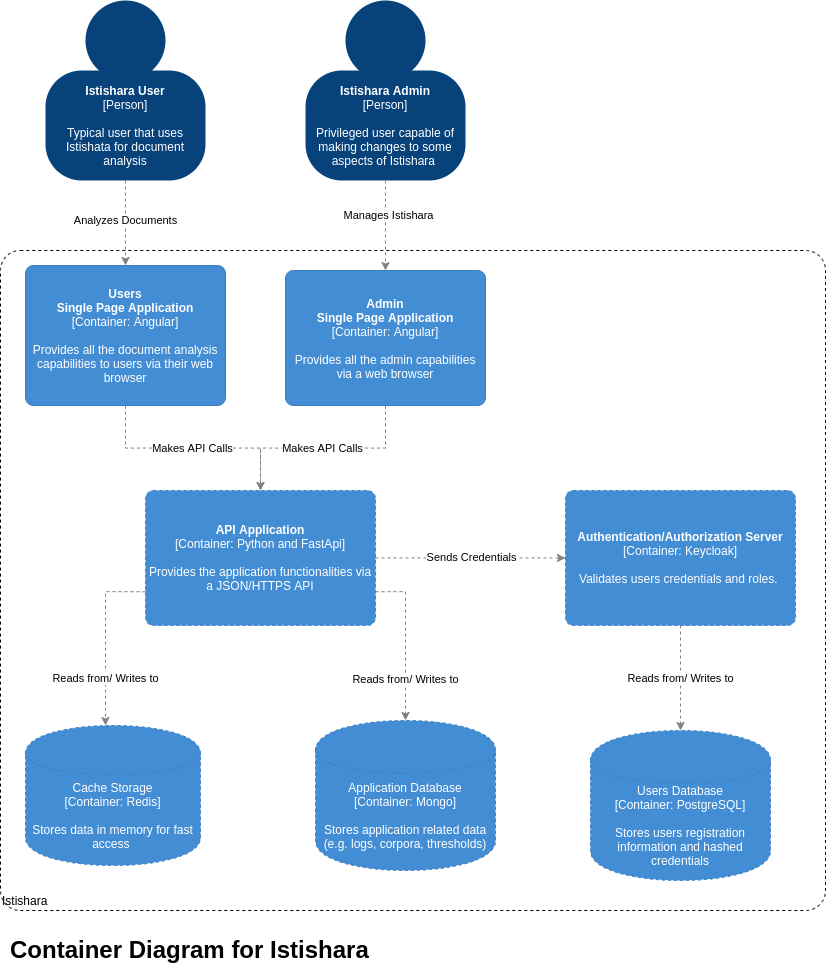
Description

Istishara is a software tool aiming to achieve Aim 1) automated document classification and relevance ranking with respect to (w.r.t.) the set of 17 SDGs defined in the UN 2030 Agenda, and Aim 2) graph-based documents visualization highlighting their relationships and connectivity with the SDGs.

Istishara allows a UN or government employee to Objective 1) easily map and evaluate the relevance of a given input document (e.g., a national development plan, or a resolution proposal) w.r.t. each of the 17 SDGs, and eventually Objective 2) visualize the set of input documents in a graph-based structure, highlighting the co-occurrence relationships between the documents themselves, as well as their connectivity (e.g., relevance) with the SDGs.

The purpose of this document is to present the technical requirements (i.e., software and hardware) needed for the deployment of Istishara. In addition, a deployment tutorial is provided to ensure a smooth installation process.

Istishara Software Architecture



Software Requirements

## Infrastructure

The following software are required to deploy and manage the application components

## [Docker](https://www.docker.com/)

Docker is a set of platform as a service (PaaS) products that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files.

Docker was used in the project to provide an ease of deployment and maintenance. Also, updating dependencies of a specific part of the application becomes much easier without breaking other parts.

## [Docker-compose](https://docs.docker.com/compose/)

Docker-compose is a tool for defining and running multi-container Docker applications. This makes the communication between the different Docker containers easier. It also helps manage the containers in a more efficient way.

## Web Application

The following software are required to serve the web applications

## [Angular](https://angular.io/)

Angular is an open source web development framework developed by Google. It provides advanced javascript support and makes the development of web applications easier and better structured. It also provides libraries for out of the box components

## [Nginx](https://www.nginx.com/)

Nginx is a well-known web server. It is used in Istishara to serve the Angular applications in production.

## Authentication and Authorization

The following are required to ensure resources protection

## [Keycloak](https://www.keycloak.org/)

Keycloak is an open source identity and access management provider. It is used for user management as well as role based authentication.

## [LDAP System](https://ldap.com/) (Optional, not required at this stage)

LDAP, the Lightweight Directory Access Protocol, is a mature, flexible, and well supported standards-based mechanism for interacting with directory servers.

LDAP Integration is required if ESCWA wants their employees to login to Istishara using their ESCWA credentials.

## API Services

The following are required to ensure resources protection:

## [Python](https://www.python.org/)

Python is an interpreted, high-level, general-purpose programming language. It is the language of choice for the API because it has a great support for text analysis and classification which are the core components of Istishara.

## [FastAPI](https://fastapi.tiangolo.com/)

FastAPI is a modern, fast web framework for building APIs with Python 3.6+ based on standard Python type hints. The main purpose of using FastAPI is to wrap our python code and make it available through HTTPS so it can be accessible from the angular app running on the client’s web browser.

## Storage

The following are required for data storage:

## [MongoDB](https://www.mongodb.com/)

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program. It is used to store terms vectors and scores.

## [PostgreSQL](https://www.postgresql.org/)

PostgreSQL is a free and open-source relational database management system. It is used to store users’ data and hashed credentials.

## Licenses

The following is a table detailing the licenses of each technology used:

|  |  |
| --- | --- |
| Technology | License |
| Docker | Apache 2.0 |
| Docker-compose | Apache 2.0 |
| Angular | MIT |
| Nginx | 2 clause BSD (FreeBSD) |
| Keycloak | Apache 2.0 |
| Python | Python Software Foundation License |
| FastAPI | MIT |
| MongoDB | MongoDb Server Side Public License |
| PostgreSQL | PostgreSQL License |

Hardware Requirements

## Operating System

Since the solution relies heavily on Docker, we use [Linux Ubuntu 18.04](https://releases.ubuntu.com/18.04/) as operating system.

Hardware

|  |  |
| --- | --- |
| CPU | 6-8 cores |
| Memory:   * Redis: rec. 6GB * MongoDb: rec. 2GB * Application: rec. 8GB | 16-24GB |
| Storage | 256 GB (SSD) |

Istishara Deployment

## Install Docker and Docker-Compose

The following links give detailed tutorials on installing Docker, making it available to non-root users and installing Docker compose:

[Install Docker Engine on Ubuntu](https://docs.docker.com/engine/install/ubuntu/)

[Manage Docker as a non-root user](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user)

[Install Docker Compose](https://docs.docker.com/compose/install/)

The install\_docker\_requirements.sh script has been provided, for convenience, to perform the aforementioned steps. It can be found in the Istishara folder.

1. Navigate to the folder of Istishara:

|  |
| --- |
| cd <path\_to\_istishara\_folder> |

1. Grant the script execution rights and run it by executing the following command in the terminal:

|  |
| --- |
| chmod +x install\_docker\_requirements.sh && source install\_docker\_requirements.sh |

## 

## Build the Solution

Since everything is wrapped in Docker containers, there is no need to install dependencies by hand. A Docker command will handle this for us. However, there are still some changes that must be made to allow the solution to function properly in your environment.

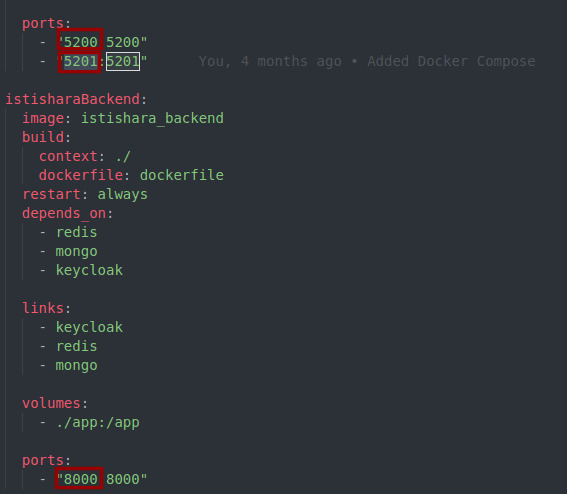
1. **Change the exposed ports (optional)**

Docker relies on port forwarding to ensure communication with the external world. That means that a port on the host machine will be mapped to a port inside of the Docker container.

The ports on the host machine can be changed by the user deploying the app.

To check which ports will be used and to change them if needed, go to Istishara’s folder and open the **docker-compose.yml** file.

The picture below represents a snippet from the docker-compose.yml file. The numbers marked with red squares are the ports that will be used by the app on the host machine and thus can be changed.



1. **Change the IP address of the backend**

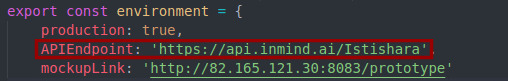
In order to ensure proper communication between the web app running on the browser and the API, the correct IP address must be provided.

To do so, follow the steps below:

1. Navigate to the environments folder in the frontend, which can be found in the following path.

|  |
| --- |
| <path\_to\_istishara\_frontend>/src/environments |

1. Inside the environments folder, open the **environment.uat.ts** file and change the **APIEndpoint** field to match the IP address and the port on which the API is running. Below is a snippet of the aforementioned file, with the field in question marked with a red square:



1. Save the changes and close the file.
2. **Run the Docker build command**

After performing the aforementioned steps, navigate to Istishara’s folder:

|  |
| --- |
| cd <path\_to\_istishara\_folder> |

Then, build the solution by running the build command:

|  |
| --- |
| docker-compose up --build |