bc2ip Tenant App Setup Guide

Version: 1.1.0

Table of Contents

1. Introduction

This document contains a detailed guide on how a technician can configure and start the BC2IP platform in 10 minutes. *After reading this document.*

2. Requirements

- OS: Linux (bash, curl, jq, wget)
- Docker
- Docker Compose
- System administrator knowledge. Especially Docker and Docker Compose, data management, backup solutions, DNS and SSL techniques.

3. Technical Overview

3.1. The BC2IP Platform

The BC2IP platform consists of multiple components. We reference them with their technical names through this document.

- **backend**: the core of the BC2IP platform. This service provides the API that is used by the webapp (see below). This API handles all the requests that are required to store files, seal versions or use any of the BC2IP platform's features. This component is the one which connects to the BC2IP Console via an API Key.
- **webapp**: the UI that runs in the browser and uses the backend service as a data provider. This is the interaction layer between the users/admins and the BC2IP backend itself.
- **db**: this is a PostgreSQL database, that holds the state of the BC2IP platform. Audit logs, user data, project data, etc. Note: the uploaded files are not stored in the database. See in the later sections.
- **tools**: this is a simple UI that runs in the browser and can even be deployed separately from the rest of the platform and provides these features:
    - Create a BC2IP Proof from a BC2IP Certificate.
    - Inspect a BC2IP Proof.
- **sdk-webservice**: this is a very simple webservice that helps the tools service by running some cryptographic calculations which cannot be done in a browser.

3.2. Configuration and Usage

3.2.1. Configuration

1. Open the **master-configuration.json** file.
2. Update the **value** fields for properties you need to set. Do NOT edit other properties.

3. Save the file.
4. Stop the application.
5. Start the application.

You need not to touch any other files rather than the master-configuration.json.
BC2IP does not have any responsibility if the new configuration fails.

3.2.2. Start, Stop

- Start: run the **start-app.sh** script and follow its instructions.
- Stop: run the **stop-app.sh** script.

See later sections what these scripts do.

3.3. The ZIP's Content

The zip contains the files below, and some of those files have copies of them ending with .tpl. See the explanation in the **master-configuration.json** and **start-app.sh** sections. We here explain the files not having .tpl extension.

3.3.1. master-configuration.json

System administrators have to touch only this file for configuring the BC2IP stack.
This configuration file holds all the configuration that can be customized and needed for the BC2IP stack. See also the **start-app.sh** section.

3.3.2. docker-compose.yml

The BC2IP platform is based on Docker containers. These containers are managed by this Docker Compose descriptor file. See the previous section for which services the BC2IP platform consists of.

3.3.3. start-app.sh

This script does the following:

1. Downloads all migration scripts from https://github.com/iop-global/bc2ip-tenant-config-migrations.The scripts are responsible to keep your installation package up to date as configurations may change and may require some changes on your local filesystem.
2. Downloads all tenant settings from the console (thus these settings must be updated in the BC2IP console):
   - administrator emails
   - the tenant's brand name
   - the tenant's brand color
   - the tenant's logos

3. Downloads BC2IP's Docker registry's credentials to be able to update the BC2IP Docker images.
4. Reads the **master-configuration.json**.
5. Displays an overview of the BC2IP stack's current configuration.
6. Update all the files based on the downloaded data and the **master-configuration.json**.
7. Downloads all updates for the BC2IP stack.
8. Restarts components if any of those get updated.
9. Logs out from the BC2IP's Docker registry.

The script collects all of the updates from everywhere, it uses the .tpl files to create new configuration files. Thus, if you edit any of the configuration files, the script will always overwrite them. Please only edit the **master-configuration.json** and/or the settings on the BC2IP Console.

3.3.4. stop-app.sh

Stops the BC2IP platform.

3.3.5. BC2IP_Tenant_Setup_Guide-1.1.0.pdf

This guide, you're reading.

3.3.6. Config

You need not touch any of these files.

This directory contains all the configurations the Docker containers use. If any of these values changed, the platform must be restarted.

backend

Contains all the configuration the backend service needs.

config.json

- **mode**: describes if the backend service runs in a production or development mode. Can be *prod* or *development*. In development mode for example, the JWT token does not expire.
- **auth.admins**: a list of email addresses who must have an admin role to the BC2IP platform.
- **auth.encryptionKey**: the key which is used for encrypting the uploaded files.
- **auth.jwt**: the security configuration for the JWT token generator.
- **auth.magicLink.secret**: the key which is used to generate secure sign in and sign up links.
- **console**: the credentials that the backend service uses to connect to the BC2IP Console.
- **frontEndUrl**: the URL where the BC2IP platform runs.

- **mailer**: configuration for mailing. The app uses NestJS's Mailer module. See this documentation for further details.

ormconfig.json

This contains the configuration for the database. Usually you should not modify it. The only thing must be noted, that few of these properties defined in this file are referenced in the **docker-compose.yml** file:

- username
- password
- database
- host
- port

webapp

Contains all the configuration files that the webapp service needs.

nginx/conf/app.conf

This is the configuration file for the webapp's Nginx container.

nginx/conf/tenant

Contains the tenant's theme configuration that was provided when the whitelabel solution was set up.

tools

nginx/conf/app.conf

This is the configuration file for the tools's Nginx container.

nginx/conf/tenant

Contains the tenant's theme configuration that was provided when the whitelabel solution was set up.

4. Data Handling

The BC2IP platform manages the following data.

4.1. Uploaded Files

These files are the files that were uploaded to a project version. *All the files are encrypted at rest* by the key you set in the configuration.

4.1.1. Preparing to Backup

The files are stored inside the backend service's Docker container. To be able to access these files and backup them, you must introduce a mount point to its container.
To be able to do that, you have to edit the **master-configuration.json** file's **backupLocations.encryptedFilesPath.value** property.

4.2. User and Project Data

BC2IP manages metadata of the users and metadata of projects and all the relations between them. BC2IP also manages audit logs, and relations between the Blockchain's state and its local state.

4.2.1. Preparing to Backup

As mentioned in the beginning, a PostgreSQL database stores all the users' and projects' data. To be able to backup these, you have to edit the **master-configuration.json** file's **backupLocations.databaseDataPath.value** property.