



ELIXIR Beacon network

Administration Guide

Project Title:	<i>ELIXIR Beacon Infrastructure</i>
Deliverable title:	<i>Service administration guide document</i>
Deliverable number:	<i>D2.1.</i>
Partner(s) contributing to this deliverable:	<i>BSC</i>
Document Author(s):	<i>Sergi Aguiló-Castillo, Jordi Rambla</i>
Document version	<i>1.0</i>

Table of Contents

Executive Summary.....	3
Purpose and Scope.....	4
Introduction.....	5
Operations for EBN Maintainers.....	5
Specification from ELIXIR Beacon Network.....	5
Load-Balance.....	5
Product updates and maintenance.....	5
Install and Deploy ELIXIR Beacon Network.....	6
Full Docker Installation (Backend and Frontend).....	6
Backend installation.....	6
Frontend installation.....	7
Update the EBN.....	7
Add a Beacon to the Network.....	8
Remove a Beacon from the Network.....	8
Update registry.....	9
Logging.....	9
Monitoring.....	9
Beacons in the EBN with problems.....	10
ELIXIR Beacon Network malfunctioning.....	10
Documentation.....	10
Operations for Beacon Maintainers.....	12
Validate the Beacon before add it to the Network.....	12
Specification from ELIXIR Beacon Network.....	13

Executive Summary

The Administration Guide for the ELIXIR Beacon Network (EBN) provides instructions and guidelines for administrators to manage and maintain the network effectively. It includes essential information on installation, configuration, beacon management, troubleshooting, and best practices to ensure the smooth operation and security of the EBN.

Purpose and Scope

The purpose of the Administration Guide for the ELIXIR Beacon Network (EBN) is to provide comprehensive and detailed instructions to network administrators for the effective management, maintenance, and operation of the EBN. The guide aims to ensure that administrators have the necessary information, procedures, and best practices.

Introduction

This guide serves as a comprehensive resource mainly for ELIXIR Beacon Network (EBN) maintainers, but also it is useful for beacon maintainers, and other stakeholders involved in the management and operation of the EBN. Thus, the document is divided by two parts, the main one for EBN Maintainers and the section for Beacon Maintainers.

The administration guide provides instructions, procedures, and best practices to ensure smooth functioning of the EBN. It covers various aspects, including beacon management, deploying technical infrastructure, communication and collaboration, and documentation.

Operations for EBN Maintainers

Specification from ELIXIR Beacon Network

The ELIXIR Beacon Network Maintainer must verify the beacons that are part of the network, with additional properties from the original standard, including:

- `'isAggregated'` Property: Set to `'true'`, indicating aggregation functionality to query multiple beacons simultaneously. The absence of this property means that the given beacon does not have aggregation functionalities.
- Unique `'beaconId'`: Each beacon within the network, including the EBN, is assigned a unique identifier enabling individual identification and preventing duplicates.

Load-Balance

To ensure continuous operation of the EBN, a load-balancing approach can be implemented. The primary objective is to maintain a main functioning EBN server as the primary point of query submission. Additionally, a secondary server, located in a separate location, serves as a backup to assist in scenarios where the primary server experiences downtime or becomes overloaded with queries. This setup enhances the reliability and resilience of the network, ensuring uninterrupted access to beacon data for users.

Product updates and maintenance

For the maintenance and updates of the ELIXIR Beacon Network (EBN) code base, it is recommended to utilize GitHub's tools, specifically Issues and Pull Requests, within the desired repositories. These tools facilitate efficient communication and collaboration among contributors and maintainers.

Notifications regarding updates to beacon maintainers and EBN maintainers could be communicated via a focused email list, as outlined in the [ELIXIR Beacon Network Customer Support Model](#). This ensures timely dissemination of information and enables swift action to address any maintenance or update requirements.

Install and Deploy ELIXIR Beacon Network

Note!

The following instructions are up to date at the time of writing. As each component and corresponding repository evolves, our recommendation is to check and follow the instructions provided in each repository, using the text below as supplementary.

Full Docker Installation (Backend and Frontend)

To download the code from the repository and set up the ELIXIR Beacon Network Frontend and Backend, follow these steps:

1. Clone the [repository](#) and navigate to the project directory:

```
git clone https://github.com/elixir-europe/beacon-network-docker.git
cd beacon-network-docker
```

2. Edit the `config.json` file located in the `frontend` folder to point to the URLs where you will be making the queries. Here's an example of how the `config.json` file should look:

```
{
  "API_URL": "http://localhost:8080/beacon-network/v2.0.0",
  "REDIRECT_URL": "https://yourUIDomain.com",
  "KEYCLOAK_URL": "https://yourKEYCLOAKdomain.com"
}
```

Then, replace "<http://localhost:8080/beacon-network/v2.0.0>" and the other URLs with the appropriate URL where your backend is hosted.

3. Run [Docker Compose](#) to build and start the containers:

```
docker compose up -d
```

With these instructions, the Beacon Network FrontEnd will be accessible at <http://localhost:8080/> and the backend at <http://localhost:8080/beacon-network/v2.0.0/>. Adjust the URLs as needed based on your setup.

Backend installation

To install the backend of the EBN, follow these steps:

1. Download the code from this [repository](#) and go to the correct folder:

```
git clone https://github.com/elixir-europe/beacon-network-backend.git
cd beacon-network-backend/docker
```

2. Run [Docker Compose](#) and start the containers:

```
docker compose up -d
```

With these instructions, the backend of the ELIXIR Beacon Network will be created and accessible at <http://localhost:8080/beacon-network/v2.0.0/>. You can now view and query the backend as needed. Adjust the URLs as necessary based on your setup.

Frontend installation

To set up the ELIXIR Beacon Network Frontend, follow these steps:

1. Download the code from the [repository](#) and navigate to the project directory:

```
git clone https://github.com/elixir-europe/beacon-network-ui.git
cd beacon-network-ui
```

2. Create a `.env` file inside the folder and modify the following variables with the appropriate values:

```
REACT_APP_CLIENT_ID="ID of your LS Login"
REACT_APP_CLIENT_SECRET="password of your LS Login"
```

3. Run [Docker Compose](#) and start the containers:

```
docker-compose up -d -build
```

4. Edit the `config.json` file located in the `frontend/src` folder to point to the URLs where you will be making the queries. Here's an example of how the `config.json` file should look:

```
{
  "API_URL": "https://yourAPIdomain.com/beacon-network/v2.0.0",
  "REDIRECT_URL": "https://yourUIDomain.com",
  "KEYCLOAK_URL": "https://yourKEYCLOAKdomain.com"
}
```

Access the Beacon Network Frontend at <http://localhost:3000>. Adjust the URLs as needed based on your setup.

Update the EBN

To update the ELIXIR Beacon Network (EBN) with a new image, follow these steps:

1. Pull the latest image:

```
docker-compose pull
```

2. Recreate and rebuild the containers with the new image:

```
docker-compose up --force-recreate --build -d
```

These commands will download the latest image for the EBN and recreate the containers using the new image. The `--force-recreate` flag ensures that the containers are

recreated even if they are already running, and the `--build` flag rebuilds the containers if necessary. The `-d` flag runs the containers in detached mode.

Add a Beacon to the Network

To add a beacon to the ELIXIR Beacon Network (EBN), the beacon maintainer must first send an email to the EBN maintainer containing the URL of the beacon's API. Before sending the URL, the beacon maintainer must ensure that the beacon passes the [validator](#), which verifies that the beacon meets the necessary specifications for integration into the network.

The ELIXIR Beacon Network Maintainer is responsible for requesting the validator output file to confirm that any Beacon meets validation criteria. Subsequently, the Beacon owner should forward the validated Beacon's URL to the EBN Maintainer.

Once validated, the EBN Maintainer will incorporate the URL of the Beacon into the ELIXIR Beacon v2 Network. This involves adding the Beacon's URL to the ``wildfly/BEACON-INF/beacon-network.json`` file. The structure of this JSON file should be an array containing the URL of the validated Beacon, like this:

```
[
  "your.beacon.url/api"
]
```

After adding the URL to the ``beacon-network.json`` file, the Beacon Network will automatically fetch the new Beacon, ensuring its integration into the network. This process facilitates the incorporation of Beacons into the ELIXIR Beacon Network.

Remove a Beacon from the Network

To remove a beacon from the ELIXIR Beacon Network, the beacon maintainer needs to send an email to the EBN maintainer requesting the removal of the beacon. This email should include relevant details such as the beacon's identifier and a justification for its removal.

To remove a Beacon from the ELIXIR Beacon Network, follow these steps:

1. Access the ``beacon-network.json`` file located in the ``wildfly/BEACON-INF/`` directory.
2. Remove the URL of the Beacon you want to remove from the ``beacon-network.json`` file.
3. Save the changes to the ``beacon-network.json`` file.
4. Once the URL of the Beacon is removed from the ``beacon-network.json`` file, the Beacon will no longer be part of the network.

By following these steps, you can effectively remove a Beacon from the ELIXIR Beacon Network.

Update registry

The EBN routinely checks for changes in the metadata of beacons on an hourly basis. It accomplishes this by storing metadata hashes, enabling the detection of changes without the need to parse responses. If there is an update in a beacon and it experiences changes making it going down or malfunctioning, the EBN continues to display its metadata along with the corresponding flag indicating the beacon's status.

To update the registry of the ELIXIR Beacon Network (EBN) after a metadata update in any of the Beacons, you can follow these two methods:

1. Automatic update: The EBN automatically checks for metadata updates every hour. You can simply wait for this automatic update process to occur, and the registry will be refreshed accordingly.
2. Command Line Refresh: If you require an immediate refresh of the metadata, you can use the `curl` command to hard-refresh the metadata of the Beacon. Here's the command:

```
curl -H "Accept: application/json" -H "Cache-Control: no-cache"
{url_of_your_ebn}/info
```

Replace `{url_of_your_ebn}` with the URL of your ELIXIR Beacon Network. By executing this `curl` command, you will trigger a manual refresh of the metadata, ensuring that any recent updates in the Beacons' metadata are reflected in the registry of the ELIXIR Beacon Network.

Logging

The application provides simple SQL logging, which can be configured using the `BEACON_NETWORK_LOG_LEVEL` environment variable. The different logging levels and their functionalities are as follows:

- "NONE": Only the informational responses (`/info`) from beacons are logged.
- "METADATA": Metadata from the beacons in the network is logged. This includes information from the `/info`, `/configuration`, and `/map` endpoints.
- "REQUESTS": Metadata from the beacons and the requests are logged. Additionally, HTTP response status codes (e.g., 200, 404) are logged, but not the responses themselves.
- "RESPONSES": Metadata from the beacons, the requests, and the responses are stored.
- "ALL": This option is equivalent to the "responses" level of storage. It is included for future-proofing, in case additional options are introduced later.

Monitoring

The monitoring of the EBN must have at least two sections: network performance and traffic. When reporting the monitoring to the Steering Committee, the EBN Maintainer must have data of the performance of the website, such as average time response, downtime in a certain period of time, etc. Regarding the traffic, it must report how many people use the

application daily, monthly and yearly with the number of queries done per user and total number of queries. These details can provide information on how to improve the EBN in the future.

Beacons in the EBN with problems

When a beacon in the ELIXIR Beacon Network (EBN) encounters problems, the following steps should be taken:

1. *Beacon is down:* If a beacon in the network is not responding, contact the owner of the Beacon to address the issue. Even if the Beacon is currently inactive, it will still be displayed in the /info page of the EBN if metadata was previously available. Additionally, restarting Docker will gather the metadata of the Beacon from the EBN's cache. If you remove the SQL database where the cache is stored, the Beacon won't be able to return the metadata information.
2. *Beacon is up but malfunctioning:* If a beacon is operational but fails to return any queries or exhibits malfunctioning behavior, contact the owner of the Beacon to report the issue. Request them to use the validator tool to identify any errors. If the problem persists, contact the EBN Operations Team for assistance.

By following these steps, you can effectively address issues with beacons in the ELIXIR Beacon Network, ensuring the smooth operation of the network for users and stakeholders.

ELIXIR Beacon Network malfunctioning

In the event of malfunction in the EBN, follow these steps to address the issue:

1. *Identify Errors:* Determine the specific errors or issues occurring with the EBN.
2. *Gather logs:* Collect logs from Docker related to the malfunction. These logs can provide valuable information for diagnosing and resolving the issue.
3. *Contact the Operations Team:* Reach out to the team responsible for the EBN. Provide them with details of the errors observed and attach the logs from Docker.
4. *Describe the issue:* Explain the nature of the malfunction, including any error messages or unusual behavior encountered.
5. *Collaborate for resolution:* Work with the Operations Team to diagnose and resolve the issue. Provide any additional information or assistance they may require to speed the process.

By following these steps and collaborating with the Operations Team, you can help ensure the timely resolution of any malfunctioning issues with the ELIXIR Beacon Network, minimizing disruptions to its operation.

Documentation

Maintaining a comprehensive documentation for beacon operators and users is an ongoing process, essential for ensuring clarity and usability of the ELIXIR Beacon Network (EBN).

This documentation should be regularly updated to incorporate new features and enhancements, driven by feedback from users. By continuously improving the documentation, operators and users can stay informed about the latest functionalities and best practices.

For documentation-related issues with the ELIXIR Beacon Network (EBN), follow these steps:

1. *Identify Documentation Errors:* Determine any inaccuracies, inconsistencies, or missing information in the documentation related to the EBN.
2. *Gather Information:* Collect details about the specific documentation issues, including the affected sections or topics.
3. *Provide Examples:* If possible, provide examples or instances where the documentation is unclear or insufficient. This can help the documentation contributors better understand the issue and make appropriate revisions.
4. *Collaborate for Improvement:* Help to write, address the identified issues and improve the quality of the documentation. Offer any additional insights or suggestions that may help enhance the clarity and usefulness of the documentation.
5. *Follow-Up:* Stay engaged with other documentation contributors to track progress on resolving the identified issues. Provide feedback and clarification as needed to ensure that the documentation meets the needs of users effectively.

By following these steps and collaborating with other documentation contributors, you can help ensure that the documentation for the ELIXIR Beacon Network is accurate, comprehensive, and user-friendly.

Furthermore, implementing training programs is crucial for educating data providers, beacon operators, and users on the best practices and guidelines associated with the EBN. These training programs should cover a range of topics, including data submission protocols, technical specifications, ethical considerations, and compliance with standards. By providing comprehensive training, the EBN can empower stakeholders to effectively utilize the network and contribute to its success.

Operations for Beacon Maintainers

Validate the Beacon before add it to the Network

This validation process can be initiated either through the FrontEnd page or the [GitHub repository](#). The validator tool assesses the beacon's compliance with the schema outlined in the "/map" endpoint.

To validate a Beacon before adding it to the ELIXIR Beacon Network, follow these steps:

1. Ensure you have [Apache Maven](#) installed.
2. Download the validation tool from the [repository](#) and navigate to the `tool` folder:

```
git clone
https://github.com/elixir-europe/neat-beacon-v2-validator/tree/master
cd neat-beacon-v2-validator
```

3. Build the code using Maven:

```
mvn install
```

4. After building, you'll find a `target` directory within the `neat-beacon-v2-validator-tool` folder. This directory contains the validator Java application.
5. Run the validator application with the Beacon's API endpoint to be validated. You can use the following command:

```
java -jar neat-beacon-v2-validator.jar -f https://beacons.bsc.es/beacon/v2.0.0/
```

This command will validate the beacon instance located at the specified API endpoint.

Optionally, you can specify an output file for the validation report using the `-o` or `--output` flag. For example:

```
java -jar neat-beacon-v2-validator.jar -f https://beacons.bsc.es/beacon/v2.0.0/
-o report.json
```

Available parameters for the validator application:

```
-h (--help)           - help message
-f (--framework)      - location of the beacon
-o (--output)         - report output file
```

The beacon owner is responsible for addressing any errors reported by the validator. While the beacon may still function within the Beacon Network despite reported errors, critical errors may prevent it from returning data. Network maintainers may request the output of the validator before adding the instance to the network.

Specification from ELIXIR Beacon Network

The Beacon Owner must have the following property in their beacons:

- Unique `beaconId`: Unique identifier enabling individual identification and preventing duplicates.