

Where is the power

Technical Installation Manual (v1.0)

Introduction	3
Manual Overview	3
Section 1: System Architecture	4
1.1 Introduction to System Architecture	4
1.1.1 Scope	4
1.1.2 Purpose	4
1.2 High-Level Overview	4
1.2.1 System Goals and Objectives	4
1.2.2 System Components	4
1.2.3 Technology Stack	4
1.3 Scalability and Redundancy	5
1.3.1 Scalability Strategies	5
1.3.2 Redundancy and High Availability	5
Section 2: Amazon ECR Configuration	6
2.1 Introduction to Amazon ECR Configuration	6
2.1.1 Scope	6
2.1.2 Purpose	6
2.2 Repository Configuration	6
2.3.1 Repository Creation	6
2.3.2 Image Lifecycle Policies	6
2.4 Docker Image Manipulation	6
2.4.1 Pushing and Pulling images.	6
Section 3: Amazon ECS Setup	8
3.1 Introduction to Amazon ECS	8
3.1.1 Scope	8
3.1.2 Purpose	8
3.2 Task Definitions	8
3.2.1 Task Definition Creation	8
3.2.2 Versioning and Updates	8
3.3 Logging and Monitoring	8
3.3.1 Viewing Logs	8
3.3.2 Deployment Alerts	8
Section 4: Database Setup	9
4.1 Introduction to MongoDB	9
4.1.1 Scope	9
4.1.2 Purpose	9
4.2 Amazon EC2 Setup	9
4.2.1 Instance creation	9
4.2.2 Instance setup	9
4.3 Network Setup	9
4.3.1 Firewall Configuration	9
Section 5: Local Development Environment Rust Rocket API	10

5.1 Environment Variables	10
5.1.1 Database Environment Variables	10
5.1.2 Mapbox API Key	10
5.2 Building and serving	10
5.2.1 Building	10
5.2.2 Serving	10
Section 6: Local Frontend Development environment	11
6.1 Environment setup	11
6.2 Dev setup	11
Final Words	12

Title	Back-End Technical Manual
Project	Where is the Power
Version	1.0

Introduction

This manual serves as an indispensable resource for software engineers, developers, and system administrators responsible for maintaining and enhancing the back-end infrastructure of the "Where is the Power" ecosystem. It provides a detailed roadmap for understanding the intricacies of the server-side components that drive our platform, ensuring its reliability, scalability, and optimal performance.

Manual Overview

This manual is organized into sections, each dealing with different aspects of the back-end system. Here is a brief overview of what you can expect to find in each section.

1. **System Architecture:** This section outlines the high-level architecture of the back-end system, describing the various components and their interactions. It provides an overview of the technology stack used to build our platform.
2. **Amazon ECR Configuration:** Upload and label Elastic Container Registry images, setup lifecycle policies and more.
3. **Amazon ECS Setup:** Create your containerized environment and start an instance.
4. **Database setup:** Configure and deploy MongoDB on AWS EC2 with firewall rules to prevent unauthorized access.

By the time you've finished reading this manual, you'll have a comprehensive understanding of the installation and maintenance of "Where is the Power".

Section 1: System Architecture

1.1 Introduction to System Architecture

1.1.1 Scope

This section covers the high-level architecture of the system, emphasizing AWS services like ECR and ECS and the MongoDB database.

1.1.2 Purpose

Understanding the system architecture is essential for developers and administrators to comprehend how "Where is the Power" functions and to maintain its robustness and scalability.

1.2 High-Level Overview

1.2.1 System Goals and Objectives

The primary goal of "Where is the Power" is to provide users with real-time energy consumption data, historical analysis and predictive insights promoting sustainability and efficiency.

1.2.2 System Components

- **Rust Web Server:** Serves as the entry point for user requests and handles HTTP traffic.
- **Application Server:** Manages business logic, data processing, and interactions with MongoDB.
- **Database Management System (DBMS):** MongoDB, a NoSQL database, stores data, especially GeoJSON data, very efficiently and provides the ability to create indices on geo-spatial data.

1.2.3 Technology Stack

- **Web Server:** Rocket v0.5-rc3
- **Application Server:** Tokio
- **Database:** MongoDB
- **Containerization:** Docker
- **Orchestration:** AWS ECR, ECS

1.3 Scalability and Redundancy

1.3.1 Scalability Strategies

- Amazon ECS allows for horizontal scaling by launching additional container instances as traffic increases.
- MongoDB can be horizontally scaled using replica sets or sharding for data distribution.
- Rust, Rocket, and Tokio provide a great layer of isolation

1.3.2 Redundancy and High Availability

- Amazon ECS ensures high availability by distributing containers across multiple Availability Zones.
- MongoDB replica sets maintain data redundancy and fault tolerance.

Section 2: Amazon ECR Configuration

2.1 Introduction to Amazon ECR Configuration

2.1.1 Scope

This section covers the setup and maintenance of Amazon ECR repositories.

2.1.2 Purpose

Understanding the configuration of Amazon ECR is essential for system administrators and developers to effectively manage container images within the “Where is the Power” application.

2.2 Repository Configuration

2.3.1 Repository Creation

1. The first step to create a repository is to log in to Amazon ECR and navigate to your private repositories.
2. Click on **Create Repository** and fill out the resulting form with all the required fields¹.

2.3.2 Image Lifecycle Policies

After creation of a repository, it is possible to set up “Lifecycle Policies.” These policies describe how docker container images should be stored within your repository. Specifically, they allow you to automatically delete old images by tag. See the [Amazon ECR documentation](#) for more information.

2.4 Docker Image Manipulation



2.4.1 Pushing and Pulling images.

Logging in to Amazon ECR is as simple as installing the Amazon AWS CLI (Command-Line Interface), running the following command, and filling out your specific details.

```
aws ecr get-login-password --region region | docker login --username AWS
--password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

¹ Ensure that the repository’s visibility is set to *Private*.

To push a docker image to ECR the following commands are recommended by the official documentation.

1. `docker tag  e9ae3c220b23
<aws_account_id>.dkr.ecr.<region>.amazonaws.com/my-repository:tag`
2. `docker push 
<aws_account_id>.dkr.ecr.<region>.amazonaws.com/my-repository:tag`

Section 3: Amazon ECS Setup

3.1 Introduction to Amazon ECS

3.1.1 Scope

This section covers deploying a cluster to Amazon ECS.

3.1.2 Purpose

Understanding The Amazon ECS setup process is crucial to deploying, managing, and orchestrating containerized applications within the “Where is the Power” system.

3.2 Task Definitions

3.2.1 Task Definition Creation

1. After logging in to Amazon ECS, navigate to Task Definitions and click on **Create New Task Definition**.
2. Fill out the form with your specific requirements, ensuring that container port 443 is open to the outside world.

3.2.2 Versioning and Updates

Versioning and updates are taken care of by the CI/CD pipeline. Each new version is sequentially numbered, starting at rev. 1. Newly built images are tagged with their SHA256 hash to ensure that duplicate images are not uploaded. Maintaining a list of artifact hashes also makes caching easier.

3.3 Logging and Monitoring

3.3.1 Viewing Logs

Container logs can be viewed on a “per service” basis by accessing the cluster configuration menu and picking the service in question. By navigating to the **Logs** tab, system administrators may view the latest logs generated by the Docker container.

3.3.2 Deployment Alerts

GitHub automatically notifies code owners of deployment failures via email. Detailed logs of deployments can be viewed under the **Actions** tab of the [public GitHub repository](#).

Section 4: Database Setup

4.1 Introduction to MongoDB

4.1.1 Scope

This section deals with installation and configuration of MongoDB on Amazon EC2.

4.1.2 Purpose

Database setup is the last step in deploying the “Where is the Power” back-end system. It provides a place to store and retrieve data in a secure and efficient manner. Knowing how to perform this basic setup is an essential part of any database administrator’s skillset.

4.2 Amazon EC2 Setup

4.2.1 Instance creation

Creation of an Amazon EC2 instance entails navigating to the **Instances** page and clicking on **Launch instances**. Fill out a name, choose the number of instances to create, and configure the specifications of those instances.

4.2.2 Instance setup

Once the EC2 instance is running, log in via SSH and install MongoDB. Refer to the [official documentation](#) for details on MongoDB installation. Create a database user and grant it permission to modify the “wip” database.

4.3 Network Setup

4.3.1 Firewall Configuration

By default, the newly created EC2 instance cannot be interacted with by the outside world. This includes our Amazon ECS cluster. To rectify this situation, add the following firewall rule to the instance’s attached security group.

```
ALLOW    27017    TCP    <IP>
```

Where “<IP>” is the ECS Cluster’s allocated IP block in CIDR notation, e.g. 172.143.20.0/16.

Section 5: Local Development Environment Rust Rocket API

5.1 Environment Variables

5.1.1 Database Environment Variables

Before running the local development server, it is important to first fill out the “.env” file with your local configuration. The file should contain the key **DATABASE_URI** with the MongoDB connection string.

5.1.2 Mapbox API Key

The mapbox API key for the frontend should be added to the “.env” file as well with the key **MAPBOX_API_KEY**

5.2 Building and serving

5.2.1 Building

The backend server can be built by navigating to the “api” source code directory and executing **cargo build**. This will generate the executables require for testing and debugging code without going through the expensive process of building a docker container.

5.2.2 Serving

The backend server can be deployed locally by running **cargo run** or **bacon run** (for hot reloading).

Section 6: Local Frontend Development environment

6.1 Environment setup

6.1.1 locate environment.ts in the app directory. Replace HelloAPIKey with a Mapbox key.

6.2 Dev setup

- Node - Version 16
- Ionic - Version 7.1.1 (npm install -g @ionic/cli)
- Angular - version 15

In app/WhereIsThePower execute **npm install**. Once done, run **ionic server** to serve the frontend locally.

Final Words

This concludes the overview of the technical installation of “Where is the Power.” In the previous sections, all critical knowledge of the System Architecture, setup, and maintenance was discussed in detail. We extend our gratitude to each member of the technical team for their diligence and dedication throughout the installation process.

Good Luck

