Welcome to the Private Beta of DKG Edge node!

Hey there, awesome early adopter! 👋

We're thrilled to have you on board as one of the first users to try out **our first Edge node**. This is a **private beta** release, which means you're among the exclusive few testing our app before the full release. Your feedback will be key in shaping the future of the **Edge node**, so don't hesitate to share your thoughts, suggestions, or even those pesky bugs 🐛 (it's still a work in progress, after all!).

In this guide, we'll walk you through how to set up and get **Edge node** running on your machine. It's not perfect yet, but with your help, we're getting closer to launching something amazing. So let's dive in and get things up and running! 🚀

Below you will find instructions on how to:

- **Option 1: Setup Edge node services in your local development environment** in order to be able to develop your own processing (knowledge mining) pipeline's and DRAGs
    - **Option 1.1 (recommended) -** Local Edge node services setup with pre-deployed v8 testnet Core node
    - **Option 1.2:** Local Edge node services setup with **local DKG network**
- **Option 2:** Deploy Edge node with all components on Linux server

Option 1: Manual Installation on MAC

Prerequisites

1. System Requirements

Before installing **Edge node services**, make sure your system meets the following requirements:

- **Operating System**: macOS
- **RAM**: At least 8 GB
- **CPU**: 4

- **Storage**: At least 2 GB available space
- **Network**: Stable internet connection
- **V8 DKG Core node** up and running (setup instructions)

2. Software Dependencies

Ensure the following services are installed:

- **Git**
- **MySQL 8**
- **Redis**
- **Node.js v22.4.0**

3. Installation steps

1. Preparation
    a. **Option 1.1 (recommended) -** Local Edge node services setup with deployed v8 DKG Core node
        i. **Whitelist your local IP** on the v8 DKG Core node - establishing communication between **local services** and **v8 Core node**

```
ssh into your V8 DKG Core node node server
nano .origintrail_noderc
in auth-> ipWhitelist-> add your IP address
restart the node
```

    b. **Option 1.2:** Local Edge node services setup with **local DKG network**
        i. **Choose one node** from local network and use it in all configuration parameters across services

2. Installation
    a. **Clone repositories** in your local environment (git clone)
        i. Authentication service
        ii. Edge node API
        iii. Edge node interface
        iv. Knowledge mining API
        v. DRAG
    b. Setup process:
        i. Create paranet:
            1. Prepare you **wallet** with funds - you should have TRAC and ETH on **Base Sepolia**

2. **Create Paranet -** you can **follow instructions on the link**. Paranet UAL should be added on following places:
    a. V8 DKG Core node config (.origintrail_noderc)
    b. Edge node Auth service -> **userConfig** table, option: ***edge_node_paranet_ual*** *(this will be generated automatically inside of Authentication service setup and it needs to be replaced with real Paranet UAL value)*

ii. Edge node **Authentication** service
1. Follow **readme** file for instructions on how to install it locally - this service is being used between all other services
2. Server should be running on http://localhost:3001

iii. Edge node **API**
1. Follow **readme** file for instructions on how to install it locally
NOTE: Keep in mind that installation of dependencies can take some time because we are not installing officially released version of our clients
2. Server should be running on http://localhost:3002

iv. Edge node **UI**
1. Follow **readme** file for instruction on how to install it locally
2. Server should be running on http://localhost:5173

v. Edge node **Knowledge mining**
1. Follow **readme** file for instructions on how to install it locally
2. Server should be running on http://localhost:5005

vi. Edge node **dRAG**
1. Follow **readme** file for instructions on how to install it locally
2. Server should be running on http://localhost:5002

c. You are ready to start using the Edge node!

Option 2: Automated Installation on Linux

Github repository: https://github.com/OriginTrail/edge-node-installer

1. System Requirements

- **OS**: Linux
- **RAM**: At least 8 GB
- **CPU**: 4
- **Storage**: At least 60 GB available space

- **Network**: Stable internet connection

2. Software Dependencies:

Ensure the following services are installed:

- **Git**

Step 1: Clone the installer repository

Use the following commands to clone the beta repository:

```
Unset
git clone https://github.com/OriginTrail/edge-node-installer.git
cd edge-node-installer
chmod +x edge-node-installer.sh
```

Step 2: Execute installer

Note: With the current installer, you will get 2 knowledge mining pipeline examples and 1 SPARQL based dRAG example (as presented in the demo).

2.1 Run the installer with:

```
Unset
./edge-node-installer.sh
```

2.2 What will installer provide you with:

- Cloning official Edge node **service** repositories
    - V8 DKG Core node
    - Edge node interface
    - Edge node API
    - Knowledge mining API
    - DRAG
    - Authentication service

- Setup runtime environments for all Edge node components (Node.js, Python, MySQL, Redis, Apache airflow and deploy systemctl services for each Edge node component)

2.3 Configure Edge node services:

- **Setting up V8 DKG Core node:**
    - You will be provided with ot-node directory on the server
    - Enter ot-node directory and update .origintrail_noderc file (custom configuration) where placeholders will already be in place:
        - `<SERVER_PUBLIC_IP>` (for NAT)
        - `<your_sharesTokenSymbol>`
        - `<your_sharesTokenName>`
        - `<RPC_ENDPOINT>` (Base Sepolia)
        - Whitelist a public IP of the server on the node (required for Edge node services to be able to communicate with V8 DKG Core node)

```
Unset
  "auth": {
    "ipWhitelist": [
      "::1",
      "127.0.0.1",
      "<server_public_ip>"
    ]
  }
```

    - Start V8 DKG Core node (commands)

Create paranet:
There are 2 ways of creating a paranet:

- Using Client (DKG.js or DKG.py) - Instructions
- Run **create-paranet.js** inside of **Edge node API** repository
    - Provide the script with the wallet details and run it with `node create-paranet.js`
- Update your V8 DKG Core node to sync your paranet (.origintrail_noderc instructions)
- **Second place** where the paranet UAL should be added is Edge node Authentication service -> UserConfig table (during configuration of this service).

- Setting up **environment variables** for each service
NOTE: *Installer will set up generic values for main parameters, but depending on the needs of the project, those parameters **should be manually defined by the user**. This will be explained in next chapters.*

Step 3: Update configuration files for Edge node services (.env):

Each Edge node service has an instruction (README) to help you configure them properly.

- V8 DKG Core node
- Edge node interface
- Edge node API
- Knowledge mining API
- dRAG
- Authentication service

Step 4 (optional): Configure installer with your repositories and branches

Note: This step is required in case you wish installer to use your forked repositories, branches and features.

Open the installer with nano or any other file editor, and replace the following variables:

```
Unset
edge_node_knowledge_mining="<github_token_and_URL>"
edge_node_auth_service="<github_token_and_URL>"
edge_node_drag="<github_token_and_URL>"
edge_node_api="<github_token_and_URL>"
edge_node_interface="<github_token_and_URL>"
```

The installer will automatically checkout `/main` branch so make sure the latest code you wish to use is merged there.