Set up and Run your own Gaia hub

INDEX

1. Preface

Aladin Network stores application data using a storage system called Gaia. Transactional metadata is stored on the Aladin blockchain and user application data is stored in Gaia storage. Storing data off of the blockchain ensures that Aladin applications can provide users with high performance and high availability for data reads and writes without introducing central trust parties.

A Gaia system consists of a hub service and storage resource on a cloud software provider such as Azure, Amazon EC2. Gaia currently has driver support for S3 and Azure Blob Storage.

A Gaia hub runs as a service which writes to data storage. The hub service writes to data storage by requiring a valid authentication token from a requestor. Typically, the hub service runs on a compute resource and the storage itself on separate, dedicated storage. A Gaia hub stores the written data in the encrypted manner.

2. Prerequisites you need

➔ **Aws/Azure account.**

➔ **Ec2 instance(t2.micro, 30GB storage). For setting hub.**

➔ **S3 bucket or Azure blob storage for storage purpose.**

Below are the steps to follow:-

- Create EC2 instance from ubuntu 16.04 Aws AMI. https://medium.com/@jeevananandanne/guide-to-set-up-ubuntu-16-04-on-aws-ec2-instance-745f3433f16
- Aws S3 bucket/Azure blob storage for storing data. https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-bucket.html
- Configure Domain with SSL enabled.(You can purchase free domain from Freenom.com and paid from name.com, godaddy or any other) https://my.freenom.com/clientarea.php

3. Running the Gaia Hub

I. To get started running a gaia hub from source, execute the following:

$ git clone **https://github.com/dev-aladin/gaia**

$ cd gaia/hub

$ npm install

$ npm run build

$ cp config.sample.json config.json

$ npm run start

Here's a sample config file(config.json) :

```
{
  "servername": "myhubname", //  example:hub
  "port": "3000",
  "driver": "",//  exp:aws/azure
  "bucket": "name-of- the-bucket", //example:to-do
  "readURL": "",//  you have to write here read url of aws s3 bucket/azure blob storage
                    https://to-do.s3.ap-south-1.amazonaws.com/
  "cacheControl": "public, max-age=1",
  "pageSize": 20,
  "diskSettings": {
    "storageRootDirectory": "/tmp/gaia-disk"
  },
  "awsCredentials": {
    "accessKeyId": "",//  you get these credentials from your AWS account
    "secretAccessKey": ""//  you get these credentials from your AWS account
  },

  "proofsConfig": {
    "proofsRequired" : 0
  },

  "azCredentials": {
    "accountName": "",//  you get these credentials from your Azure account
```

```
    "accountKey": ""    //  you get these credentials from your Azure account
  },

  "argsTransport": {
    "level": "debug",
    "handleExceptions": true,
    "stringify": true,
    "timestamp": true,
    "colorize": true,
    "json": true
  }
}
```

➔ The following parameters of config.json file are explained as under.

Configuring the hub

Driver Selection

- The Gaia hub currently supports the following drivers:

'aws' == Amazon S3
'azure' == Azure Blob Storage

The readURL parameter

- By default, the gaia hub drivers will return read URLs which point directly at the written content. For example, an S3 driver would return the URL directly to the S3 file. However, if you configure a CDN or domain to point at that same bucket, you can use the readURL parameter to tell the gaia hub that files can be read from the given URL.
- If you have configured your hub on the aws s3 bucket then, you will get read url after creating your bucket on aws s3 console. You have to set the required

permission and bucket policy for your bucket. (This is the same applicable for your azure blob).

Minimum Proofs Requirement

- The gaia hub can also be configured to require a minimum number of social proof in a user's profile to accept writes from that user. This can be used as a kind of spam-control mechanism. However, we recommend for the smoothest operation of your gaia hub, to set the proofsConfig.proofsRequired configuration key to 0.

II. To get started running a gaia hub with docker and nginx. (This is the best and simplest way to set gaia hub)

After cloning the code, First have docker,nginx and certbot installed on a server with a FQDN pointed to it. The following guides should help you get this setup. (It is not necessary that you have to purchase SSL from Let's Encrypt only. You can purchase it from any site.)

- Install nginx
- Install docker
- Install certbot (do not run the setup yet!, for now don't follow step2)

Then follow the following steps :

- Make a folder $HOME/hub and copy the configuration file config.sample.json to $HOME/hub/config.json and add in your desired configuration.
- Copy nginx.conf.sample to $HOME/hub/nginx.conf and replace hub.example.com with your FQDN.
- sudo rm /etc/nginx/sites-enabled/default && sudo ln $HOME/hub/nginx.conf /etc/nginx/sites-enabled/default

- Finish certbot setup and cert generation.(now, go to step 2)
- Pull the docker image and start an instance of the container:

  $ docker pull aladinnetwork/gaia-hub

  $ docker run -d --restart=always -v $HOME/hub/config.json:/src/hub/config.json -p 3000:3000 -e CONFIG_PATH=/src/hub/config.json aladinnetwork/gaia-hub

➔ Now you can test the hub! The exact output will depend on your configuration.

$ curl https://hub.example.com/hub_info | jq

```
{
      "challenge_text":   "[\"gaiahub\",\"2017-09-19\",\"{{   (.serverName }}\",\"aladin_storage_please_sign\"]",
   "read_url_prefix": "https://{{ bucketName }}.{{ (.storageProviderUrl }}/"
}
```

➔ If you are getting output like above, then you have successfully set up your hub!!!!!

4. How to connect your own gaia hub through ALADIN platform.

- ➔ At the end, you have to put the write url and read url in the API setting page of Aladin browser. (if you are already signing into Aladin browser then follow below instructions.)
- ➔ Go to **Settings →Api setting**
- ➔ URL for Gaia Hub Config - replace **https://gaia.aladinnetwork.org/** your read url which is s3 bucket endpoint/your azure blob storage endpoint.

  example:-*https://to-do.s3.ap-south-1.amazonaws.com/*

- ➔ URL for Gaia Hub - replace https://hub.aladinnetwork.org to your write url

  example :https://hub.example.com

- ➔ After this, you have to click on the SAVE button.
- ➔ Congratulations!!! Now, You will be connected to your own gaia hub.
- ➔ For, cross verification You can see your data in your storage of s3 bucket and Azure blob storage.

5. Aladin File-Manager

- ➔ If you want to upload the files in your gaia hub, Go to **Profile →File Manage**r
- ➔ Now, whatever files you are uploading will go into your own gaia hub in the encrypted manner. From there, You can also download the files, which will be in decrypted manner.
- ➔ Note that, If you didn't set up your own gaia hub, then all the files will be stored in the default gaia hub of Aladin Network.

Note:-

- ➔ We *strongly* recommend that you deploy your Gaia hub with SSL enabled. Otherwise, the tokens used to authenticate with the Gaia hub may be stolen by attackers, which could allow them to execute writes on your behalf.
- ➔ Keep your config.json file confidential. Otherwise, your Aws and Azure account credentials will be stolen by Attackers.
- ➔ For your gaia hub, You want to upload large files more than 1MB, then you have to set client_max_body_size xM;(where x= 10,20 etc.) in nginx.conf file.
- ➔ This guide is for aws and azure blob storage driver.

Reference : https://github.com/dev-aladin/gaia/blob/master/hub/README.md