Платформа автоматизации процессов управления корпоративными топливными картами на базе технологий распределенного реестра и смарт-контрактов

Инструкция по установке

Листов 9

2023

Содержание

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Назначение программы

Платформа автоматизации процессов управления корпоративными топливными картами на основе технологии распределенного реестра и смарт-контрактов (далее – Платформа, Платформа «CardLedger») предназначена для обработки, учета и хранения операций заправки по корпоративным топливным картам. Данные операций заправки записываются и хранятся в системе распределенного реестра в виде транзакций.

1.2 Требования к техническим средствам автоматизации

Платформа должна быть развернута на аппаратном обеспечении, имеющим следующие характеристики:

1. процессор:
   − Intel не ниже 5 поколения (или схожий по характеристикам),
   − не менее 8 потоков,
   − частота не менее 3.6 ГГц;
2. видеокарта:
   − не менее 256 Мб видеопамяти;
3. ОЗУ:
   − не менее 32 Гб;
4. жесткий диск:
   − не менее 50 Гб.

1.3 Требования к общесистемному программному обеспечению

Для работы развертывания Платформы на серверной части должно использоваться следующее программное обеспечение:

1. ОС Ubuntu версии не ниже 21;
2. Ansible актуальной версии;
3. базовые драйверы серверного оборудования.

2 СТРУКТУРА ПРОГРАММЫ И ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Платформа состоит из следующих функциональных компонент:

1) Система распределенного реестра – подсистема предназначена для хранения данных (транзакций заправки) с высоким уровнем конфиденциальности и целостности данных, а также возможности создания и выполнения смарт-контрактов.
2) Смарт-контракт «Заправка картой топливного оператора» – программное обеспечение, предназначенное для обработки операций по КТК ТО. Смарт-контракт «Заправка картой топливного оператора» осуществляет формирование и запись транзакции заправки в Систему распределенного реестра.
3) Смарт-контракт «Заправка картой АЗС другой сети» – программное обеспечение, предназначенное для обработки операций по КТК сетей АЗС. Смарт-контракт «Заправка картой АЗС другой сети» осуществляет формирование и запись транзакции заправки в Систему распределенного реестра.
4) Пользовательское веб-приложение – программное обеспечение, предназначенное для предоставления доступа пользователей к работе с Платформой.
5) Средства интеграции с внешними информационными системами – библиотека, предназначенная для возможности обмена данными с другими программами. Данная библиотека предоставляет возможность взаимодействия внешних информационных систем с Системой распределенного реестра и смарт-контрактами Платформы.

3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1 Система распределенного реестра

В качестве системы распределенного реестра на Платформе используется система распределенного реестра «InnoChain» (далее – Система «InnoChain»).

3.1.1 Установка и запуск системы распределенного реестра

Экземпляр системы «InnoChain» доступен по ссылке: [*по запросу*].

Установка осуществляется на операционную систему Ubuntu версии не ниже 21.

Для начала работы необходимо установить утилиту docker на все сервера сети, используя следующие команды:

```
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
$ sudo apt update
$ apt-cache policy docker-ce
$ sudo apt install docker-ce
```

Чтобы не использовать постоянно sudo для работы с докером рекомендуется выполнить следующую команду:

```
$ sudo usermod -aG docker ${USER}
```

Далее необходимо установить docker-compose:

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.29.
```

```
1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

Далее необходимо распаковать архив дистрибутива и перейти в его корневую директорию. В корневой директории должны располагаться docker-compose.yml файл и директория node.

Для сборов логов используется связка сервисов loki и grafana, настройки которых находятся в директории node (local-config.yaml и grafana.ini соответственно). Также необходимо установить плагин, используя следующую команду:

```
$ sudo docker plugin install grafana/loki-docker-driver:latest --alias loki --grant-all-permissions
```

Для запуска системы необходимо перейти в директорию с файлом docker-compose.yml и выполнить команду:

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.29.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

После запуска контейнеров на каждой машине в сети станет доступен веб-интерфейс для просмотра журнала работ — Grafana. Для открытия интерфейса необходимо открыть веб-браузер и ввести https://:3000, где IP — это адрес машины. Далее для получения доступа к логам необходимо слева внизу нажать Configuration > Data Sources, в появившемся меню выбрать Loki, затем необходимо вписать строку http://{local_IP}:3100 (вместо IP необходимо указать ip-адрес сервера на котором разворачивается система) в секции HTTP в поле URL и сохранить. Теперь можно отправлять запросы через опцию Explorer, которая находится в меню слева.

Сверху будет выпадающее меню в нём нужно выбрать Loki как источник данных. Далее начните вводить {} и вам будут предложены подсказки, выберите

job и дальше подсказка выдаст названия сервисов, выберите тот, журнал работ которого необходимо просмотреть (например, сервис узла - node).

3.1.2 Проверка работоспособности Системы «InnoChain»

После развёртывания узла, первые пять секунд запросы через API могут не проходить, потому что узлу и API нужно время для запуска и установления связи. Для отслеживания состояния системы узел и API рекомендуется использовать curl со следующим набором данных:

```
curl -X POST --max-time 10 url -d '{"jsonrpc":"2.0","method":"getTransaction","params":{"hash":"0x0000000000000000000000000000000000000000000000000000000000000000","id":1},"id":1}'
```

Где:

- url - http адрес сервера, где развёрнуто API с указанием порта.
- "jsonrpc":"2.0" - версия спецификации jsonrpc.
- "method":"getTransaction" - функция узла, в данном случае это геттер, возвращающий информацию о транзакции блокчейна по её хэшу.
- params - параметры функции узла.
- hash - хэш транзакции, в данном случае он нулевой т. к. необходимо проверить в рабочем ли состоянии система узел и API. Поскольку транзакции с таким хэшем не существует узел сообщит, что информации о коммите такой транзакции нет.
- id - первое поле id вспомогательное и исользуется API для отслеживания сообщений, а второе поле id это идентификатор jsonrpc запроса.

Пример ответа, если узел и API в рабочем состоянии представлен на рисунке 1.

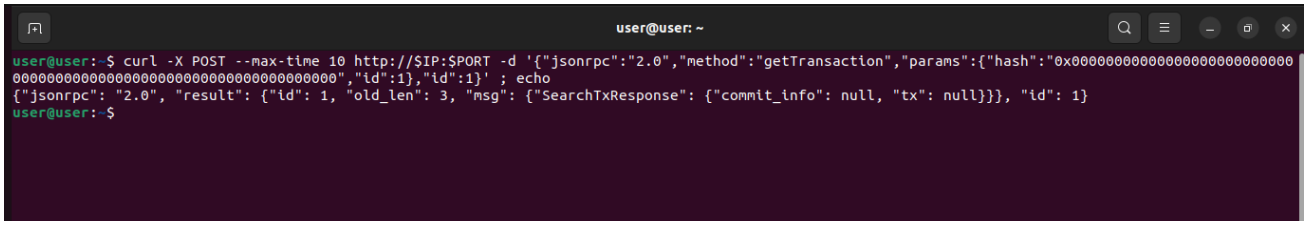

Рисунок 1. Пример положительного ответа о состоянии узла и API

Где:

– SearchTxResponce - один из 9 возможных типов сообщений.

– commit_info - подтип с информацией о коммите транзакции, в данном случае он всегда равен null т. к. транзакции с нулевым хэшем нет в блокчейне.

– tx - сериализованный при помощи scalecodec тип CommitInfo.

Пример ответа при остановленном узле представлен на рисунке 2.

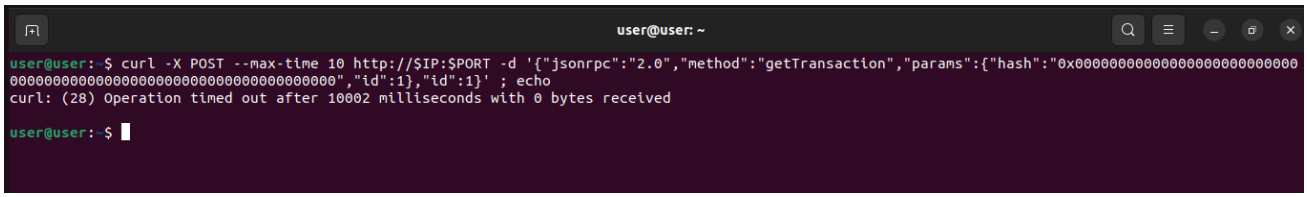

Рисунок 2. Пример отрицательного ответа о состоянии узла и API, при остановленом узле

При этом вместе с узлом необходимо перезагрузить и API т.к. дальнейшие запросы обработаны не будут.

Пример ответа при остановленном API представлен на рисунке 3.

Рисунок 3. Пример отрицательного ответа о состоянии узла и API, при остановленном API

3.1.3 Остановка работы

Для полной остановки и удаления контейнеров необходимо выполните команду:

```
$ docker-compose -f docker-compose.yml -p node down
```

3.2 Веб-приложение

3.2.1 Развертывание веб-приложения

Предварительно на сервере должен быть установлен docker и docker-compose.

Для развертывания веб-приложения на сервере необходимо:

1. Подключится и скачать дистрибутив с репозитория, на котором он размещен Экземпляр Системы «InnoChain» доступен по ссылке: [*по запросу*].
2. Перейти в папку репозитория ./backoffice. Открыть файлы в режиме редактирования docker-compose.dev.yml docker-compose.prod.yml в разделах api:environment: в поле CHAIN_IP=XX.XX.XX.XX заменить XX.XX.XX.XX на свой ip адрес узла InnoChain. Сохранить файл.
3. Запустить веб-приложение с использованием скрипта ./dev.
4. После запуска убедится, что следующие контейнеры «подняты» и находятся в статусе «up»:

- backoffice-api-db;
- backoffice-api;
- backoffice-frontend;
- backoffice-nginx.

Используя команду:

```
docker ps
```

Далее можно приступить к работе с Платформой, порядок действий пользователя Платформы представлен в Руководстве пользователя Платформы «CardLedger».