

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «АЙ-НОВУС»
(ООО «АЙ-НОВУС»)

**СЕРВИС УПРАВЛЕНИЯ НАСТРОЙКАМИ
«АЙГАЧ»**

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

На 9 листах

АННОТАЦИЯ

Настоящий документ представляет собой инструкцию по установке экземпляра программного обеспечения, предоставленного для проведения экспертной проверки.

СОДЕРЖАНИЕ

ТЕРМИНЫ И СОКРАЩЕНИЯ	4
1 ОБЩИЕ СВЕДЕНИЯ	5
1.1 Наименование программного обеспечения	5
1.2 Краткое описание возможностей программного обеспечения	5
2 МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ.....	6
3 ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	7
3.1 Особенность конфигурации IP адресов Сервиса.....	7
3.2 Запуск в режиме разработки.....	8
3.3 Запуск агента Сервиса в режиме Клиента.....	8
3.4 Запуск агента Сервиса в режиме Сервера	9

ТЕРМИНЫ И СОКРАЩЕНИЯ

Перечень принятых терминов и сокращений приведен в таблице 1.

Таблица 1 – Перечень принятых терминов и сокращений

Термин/ Сокращение	Определение/Расшифровка
HTTP-запрос	(англ. HyperText Transfer Protocol – протокол гипертекстовой передачи) – сообщения, которые отправляются клиентом на сервер, чтобы вызвать выполнение действий
HTTP API	(англ. Hypertext Transfer Protocol Application Programming Interface) – интерфейс программирования приложений, который позволяет программам взаимодействовать между собой через протокол HTTP
DNS-запрос	(англ. Domain Name System) – запрос от клиента (или сервера) серверу
IP-адрес	(англ. Internet Protocol - интернет-протокол) – уникальный числовой идентификатор устройства в компьютерной сети
JSON	(англ. JavaScript Object Notation - обозначение объектов JavaScript) – текстовый формат обмена данными, основанный на JavaScript
Агент	Фоновый процесс, запущенный на каждом узле кластера Сервера
Гб	(англ. Gbyte - Гбайт) – гигабайт, единица измерения количества информации
Клиент	Агент, который перенаправляет все вызовы RPC на сервер
ОЗУ	Оперативная память, энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код, а также входные, выходные и промежуточные данные, обрабатываемые процессором
Сервер	Агент с расширенным набором обязанностей, включая участие в распределении лидера и ведомых серверов, поддержание состояния кластера, ответ на запросы RPC
Сервис, Айгач	Сервис управления настройками «Айгач»
Токен	Уникальная последовательность символов (букв, цифр и символов), содержащая исчерпывающие данные для идентификации конкретного пользователя или системы
Хост	Любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определенное на этих интерфейсах
Центр обработки данных	Сетевая среда, которая является частной, с низким временем отклика и высокой пропускной способностью

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программного обеспечения

Полное наименование системы: Сервис управления настройками «Айгач».

Краткое наименование системы: Сервис.

1.2 Краткое описание возможностей программного обеспечения

Сервис управления настройками предоставляет следующие функциональные возможности:

- хранение древовидной базы данных настроек в формате «ключ-значение»;
- управление базой данных настроек;
- взаимодействие с базой данных настроек через командную строку для выполнения команд верхнего уровня;
- взаимодействие с базой данных настроек через пользовательский интерфейс;
- взаимодействие с базой данных настроек через HTTP API;
- создание токена с привилегиями для доступа к базе данных настроек.

2 МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ

Для обеспечения применения Сервиса необходимо наличие следующих минимальных технических средств:

- процессор с количеством ядер в диапазоне от 2 до 4;
- ОЗУ не менее 8 Гб;
- диск объемом не менее 100 Гб;
- для диска не менее 3000 операций ввода-вывода в секунду;
- пропускная способность диска не менее 75 МБ/с.

3 ИНСТРУКЦИЯ ПО УСТАНОВКЕ И НАСТРОЙКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Рекомендуется устанавливать Сервис используя Docker-образ. Docker-образ поставляется в виде архива. Далее архив загружается с помощью «dockerimageload».

В качестве базового образа используется образ на пакетной базе РЕД ОС.

На каждом хосте в кластере Сервиса работает агент Сервиса, который можно запустить в режиме Клиента или Сервера. На одном хосте, на котором запущен Docker, запускается один контейнер агента Сервиса. Каждый кластер имеет как минимум один Агент в режиме Сервера, обычно 3 или 5 для обеспечения высокой доступности. Сервис всегда следует запускать с параметром «--net=host» в Docker.

Запуск контейнера Сервиса без аргументов предоставит Сервер Сервиса в режиме разработки.

Контейнер предоставляет том «/consul/data», где будет храниться служебная информация агентов Клиентов и агентов Серверов и так же данные хранилища.

В контейнере есть каталог конфигурации Сервиса, настроенный по адресу «/consul/config», и агент будет загружать любые файлы конфигурации, размещенные здесь, путем привязки тома или путем создания нового образа и добавления файлов. Альтернативно, конфигурацию можно добавить, передав конфигурационный JSON через env-переменную «CONSUL_LOCAL_CONFIG».

3.1 Особенность конфигурации IP адресов Сервиса

У Сервиса существует адрес кластера, а также адрес клиента.

Адрес кластера – это адрес, по которому другие агенты Сервиса могут связаться с данным агентом.

Адрес клиента – это адрес, по которому другие процессы на хосте связываются с Сервисом для выполнения HTTP- или DNS-запросов.

Необходимо сообщить Сервису адрес кластера при запуске. Для этого используется аргумент «-bind=<externalip>».

Для адреса клиента используется аргумент «-client=<interfaceip>».

Задать адрес клиента и адрес кластера можно и по имени сетевого интерфейса, для этого используются переменные окружения CONSUL_CLIENT_INTERFACE и CONSUL_BIND_INTERFACE. При запуске Сервис автоматически определит IP-адрес этого интерфейса и передаст его как аргументы командной строки: «-client=<interfaceip>» и «-bind=<interfaceip>».

3.2 Запуск в режиме разработки

```
$ docker run -d --name=dev-igach -e CONSUL_BIND_INTERFACE=eth0 igach:1.0.0
```

В данном режиме Сервис хранит данные в памяти. Такой режим не подходит для запуска в продуктивной среде.

Рассмотрим пример, где Сервер работает по внутреннему адресу 172.17.0.2. Можно запустить кластер из трех узлов для разработки, запустив еще два экземпляра, указав им необходимость присоединиться к первому узлу.

```
$ docker run -d -e CONSUL_BIND_INTERFACE=eth0 igach:1.0.0 agent
-dev -join=172.17.0.2
... server 2 starts
$ docker run -d -e CONSUL_BIND_INTERFACE=eth0 igach:1.0.0 agent
-dev -join=172.17.0.2
... server 3 starts
```

Как только контейнер остановится, все состояние будет потеряно. Данный режим может быть использован для тестирования нескольких экземпляров Сервиса на одной машине, что обычно сложно сделать из-за конфликтов портов.

В режиме разработки также запускается веб-интерфейс Сервиса на порту 8500. В таком случае его необходимо добавить к другим конфигурациям Сервиса, указав для Сервиса параметр «-ui» в командной строке. Веб-ресурсы упакованы в двоичный файл Сервиса в контейнере.

3.3 Запуск агента Сервиса в режиме Клиента

```
$ docker run -d --net=host igach:1.0.0 agent -bind=<external ip> -
retry-join=<root agent ip>
```

Значением «Адрес клиента» будет 127.0.0.1. Это конфигурация используется если другие контейнеры на хосте используют «--net=host», и для того, чтобы процессы, работающие на хосте вне контейнеров, могли взаимодействовать с Сервисом.

Параметр «-retry-join» указывает внешний IP-адрес еще одного агента в кластере, который будет использоваться для присоединения при запуске. При запуске агент будет читать файлы конфигурации JSON из «/consul/config». Данные будут сохраняться в томе «/consul/data».

Чтобы предоставить доступы для других контейнеров, которые работают в другой сети, например, в сети «bridge» («bridgenetwork» – это сетевой режим по умолчанию в Docker).

```
docker run -d --net=host igach:1.0.0 agent -bind=<external ip> -
client=<bridge ip> -retry-join=<root agent ip>
```

В данной конфигурации клиентские интерфейсы Сервиса будут привязаны к

«<bridgeip>» и доступны другим контейнерам в этой сети, но не в хост-сети. Следует отметить, что адрес кластера в сети хоста для повышения производительности по-прежнему сохраняется. Сервис также может принимать опцию «-client=0.0.0.0», что означает привязку ко всем сетевым интерфейсам. Это делает Сервис доступным для всех сетей, включая сеть хоста и сеть «bridge».

3.4 Запуск агента Сервиса в режиме Сервера

```
$ docker run -d --net=host igach:1.0.0 agent -server -bind=<external ip> -retry-join=<root agent ip> -bootstrap-expect=<кол-во агентов в кластере>
```

Запускается агент Сервера Сервиса, совместно использующий сеть хоста. Все сетевые соображения и поведение, которые рассмотрены выше для клиентского агента, также применимы и к серверному агенту. Один сервер сам по себе не сможет сформировать кворум и будет ждать подключения других серверов.

Как и в случае с клиентским агентом, параметр «-retry-join» указывает внешний IP-адрес еще одного агента в кластере, который будет использоваться для присоединения при запуске. Агент Сервера также использует параметр «-bootstrap-expect» для указания ожидаемого количества серверных агентов в кластере. Данный параметр помогает Сервису определить, когда кластер считается готовым к работе и может начать выполнение своих функций.

При запуске агент будет читать файлы конфигурации JSON из «/consul/config». Данные будут сохраняться в томе «/consul/data».