GS LABS
GS GROUP

Публичный - L0

Сервис авторизации и проверки доступа

Руководство по установке

Индекс	Shield.Int-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

Содержание

1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой **Сервиса авторизации и проверки доступа** (далее в документе используется условное наименование данного сервиса - **Shield**). Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области программного обеспечения.

⚠ **Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиториев ООО "Цифра", доступ к которым предоставляется заказчику по запросу.**

2. Минимальные системные требования

Для установки сервиса необходимо наличие не менее 3 серверов с разными именами (hostname): master, worker1, worker2. Общее количество серверов должно быть нечетным.

Серверы должны удовлетворять следующим требованиям:

1. Операционная система ubuntu-20.04-server-amd64 (с установленным пакетом sudo).
2. Многоядерный центральный процессор с тактовой частотой каждого ядра 2 ГГц (минимум 20 ядер).
3. Объем оперативной памяти 64 ГБ.
4. Не менее 2 жестких дисков емкостью 500 ГБ или больше. Рекомендуется наличие на каждой ноде помимо основного дискового пространства с ОС 1-го диска SSD или NVMe и 9-ти дисков HDD (SATA, SAS), не собранных в RAID и не форматированных.
5. Два интерфейса Ethernet 100 и 1000 Base-T с поддерживаемой пропускной способностью 100 и 1000 Мбит/сек соответственно. Один предназначен для сети поддержки, второй используется для вывода генерируемого транспортного потока.

Установка должна производиться с дополнительного Ubuntu-сервера, не имеющего отношения к будущему кластеру. Требования к объему ресурсов дополнительного сервера отсутствуют.

Корректная работа сервиса гарантируется на версиях ОС Ubuntu 20.04

3. Установка и настройка баз данных

Установка и настройка баз данных описана в документе Установка и настройка баз данных (предоставляется по запросу заказчика).

4. Развертывание kubernetes кластера

Кластер развёртывается по официальной инструкции kubernetes.

Для установки сервиса в имеющийся настроенный кластер Kubernetes используется процесс CI/CD, настраиваемый с помощью GitLab.

Все действия возможно производить на локальной машине или на любом Ubuntu-сервере с доступом через консоль от имени любого пользователя.

4.1. Требования к окружению

Для развёртывания сервиса нужно соблюсти следующие требования:

1. Развернуть высокодоступный Kubernetes кластер
2. Развернуть высокодоступный PostgreSQL
3. Развернуть кластер NATS с использованием nats-operator
4. Развернуть etcd кластер

4.1.1. Распределение узлов

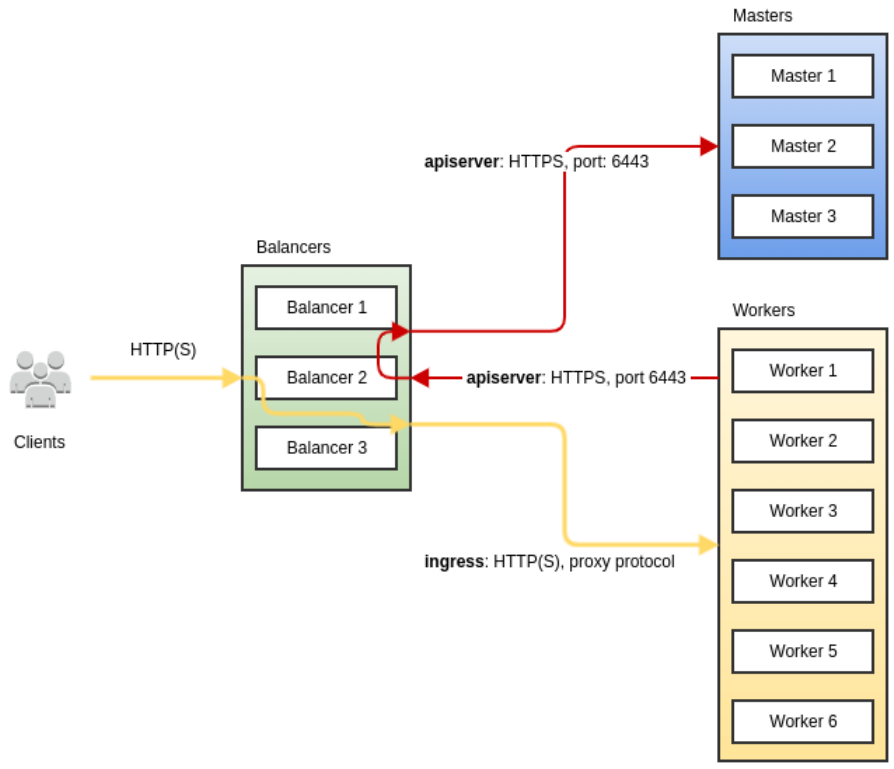
Необходимо разделить узлы логически на две роли:

Роль	Компоненты	Пример распределения ресурсов	Требования к количеству узлов	Требования к физическому местоположению	Резервное копирование
Master	• apiserver • control plane • etcd	• 2 CPU cores • 2GB RAM	1. Не менее трёх 2. Нечётное количество	Распределить по физическим машинам. Рекомендуется 1 физическая машина - 1 master нода.	Каждый час - полный snapshot etcd. (Документация https://etcd.io/docs/v3.3.12/op-guide/recovery/)

Worker	• kubelet	• 4 CPU cores • 16GB RAM	1. Не менее двух	Распределить по физическим машинам. Рекомендуется 1 физическая машина - 2 worker-ноды.	Не требуется

4.1.2. Общая схема

Перед Kubernetes кластером необходимо разместить балансировщики на базе haproxy для балансировки как внутреннего трафика kubernetes (worker<->master), так и внешнего (ingress). Балансировщики должны иметь общий Virtual IP адрес, который настраивается через keepalived. Этот адрес должен использоваться как адрес мастера при создании кластера. Таким образом, в случае отказа мастера, трафик с воркеров будет успешно достигать одного из мастеров. Схематически процесс изображен ниже:



4.1.3. Балансировка

Конфигурация haproxy (на тестовом кластере, реальные параметры подбираются исходя из доступных вычислительных мощностей):

- 3 Master узла
- 6 Worker узлов
- 3 балансера

4.1.3.1. Входящий трафик

Во внешнюю сеть должны быть открыты только эти порты:

- 80;
- 443;

В тестовых средах дополнительно может быть открыт порт 6443 для работы с kubernetes кластером.

4.2. Развёртывание внутри Kubernetes

Ссылка на проект для развертывания Shield в kubernetes
предоставляется заказчику по запросу.

4.2.1. Состав репозитория

- helmfile.yaml - конфиг helmfile
- values - папка с values для каждого чарта; они являются шаблонными и забирают значения из values окружения (файла default.yaml)
- default.yaml - значения по умолчанию
- versions.gen.yaml - файл, содержащий последние стабильные версии сервисов
- docs - документация по каждому Helm Chart, который устанавливается в систему
- documentation - документация до проекту (руководство по установке, руководство администратора, метрики)
- dashboards - дашборды, json-файлы.

4.2.2. Описание helmfile

Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов Shield. Сами компоненты поставляются в виде образов (images), из которых развертываются Docker-контейнеры.

В helmfile.yaml содержится следующая информация:

```
repositories:
  - name: < >
    url: <url ,    chart'>

releases:
  - name: < chart'>
    namespace: < namespace'>
    chart: <  chart'  >
    version: < chart'>
    condition: <  chart'>
    labels:  #     chart'a
      app: < chart'>
      project: shield-< chart'>
      product: shield
      service: < chart'>
      stage: <,     chart'a>
    values: #
      - "./values/< chart'>/<___>.gen.yaml.gotmpl"
      - "./values/< chart'>/<___>.yaml.gotmpl"
```

4.2.3. Описание values

В директории values хранятся values для каждого чарта, они являются шаблонными и забирают значения из values окружения (файла default.yaml)

Фактические значения для шаблонов хранятся в файле default.yaml, что позволяет адаптировать развертывание под конкретное окружение, используя одинаковые щаблоны.

Если значение отсутствует в default.yaml, то оно может быть задано в качестве дефолтного внутри шаблона.

4.2.4. Описание default

В файле default хранятся значения, которые необходимы для подстановки в конфигурационные шаблоны.

Ниже описана структура default файла:

```
---
namespace: < namespace'a  >

serviceMonitor:
  enabled: false
  internal: 10s

image: #   image
  pullPolicy: Always
  imagePullPolicy: Always

tracer: #  Jaeger
  enabled: 0
  agent_host: ""
  sampler_host: ""

global: #
  log:
    level: info #
  image: #   image
    pullPolicy: Always
  replicas: 1
  pg: #    postgres
    host: "<   postgres>"
    port: "<    >"
  redis: #    redis
    address: "<   redis>"
    password: "<    redis>"
  nats: #    nats
    address: "< nats>"
  etcd: #    etcd
    endpoints: "< etcd>"

#        :
< >:
  enabled: <   >
  pg:
    user: "< >"
    password: "< >"
    database: "< >"

< >Ingress:
  enabled: < ingress'  >
  hosts:
    - < >.< namespace'a  >.<>
```

4.3. Настройка CD для продукта, опубликованного в Releases

Ссылка на документ с описанием настроек предоставляется по запросу заказчика.

4.4. Действия после развёртывание Shield внутри Kubernetes

4.4.1. Создание сущности Realm по умолчанию

ВНИМАНИЕ! Процедура выполняется после успешного развертывания shield в kubernetes.

При развёртывании shield в самом сервисе Shield не будет ни одной сущности Realm, даже по умолчанию.

В связи с этим нужно указать Realm по умолчанию с помощью команды:

```
kubectl -n shield_space_name exec shield-__ -- ./shield id set default id-provider:
8080
kubectl -n shield_space_name exec shield-__ -- ./shield id set admin admin-accounts:
8080
```

4.4.2. Настройка конфигурации

Информация о конфигурации сервисов приведена в Руководстве администратора Shield (входит в комплект поставки).

Дополнительные требования:

После развертывания сервиса id_provider в переменной окружения ID_PROVIDER_NOTIFY_ADDRESS (idProvider.notify.address) необходимо задать валидный адрес notify-server в формате "notify-server-server-svc".

5. Настройка взаимодействия с внешними системами

5.1. DRE Config Manager

ВНИМАНИЕ! Для коректной работы продукта Shield 2.0 необходимо использовать продукт DRE Config Manager версии не ниже 1.4.0

1. В DRE Config Manager для типа приложения aptp_smh и кода оператора tricolor должны быть созданы параметры:
    a. drm_auth_url, drm_monetization
    b. okydrom_url, notify_time, notify_time, notify_repeat_interval, notify_types_list, create_domain_notify_body, add_apptype_notify_body.
2. В DRE Config Manager для типа приложения aptp_smh и кода необходимого оператора должны быть импортированы тексты пользовательских ошибок.