

Интерфейс системы развертывания клиентов

Определения

Среда клиента - совокупность docker-образа, сервиса k8s, записи обратного прокси сервера и двух баз clickhouse и postgres, необходимых для функционирования клиента в сервисе.

k8s - kubernetes - оркестратор развертывания сред клиентов

ingress - [reverse-proxy](#) сервер управляющий маршрутизацией запросов между экземплярами клиентских сред

Требования

- Система разрабатывается на базе платформы 1C
- Система должна хранить список клиентов:
 - Идентификатор клиента формируется по шаблону C<Номер клиента>, например C42.
 - Наименование организации клиента
 - Контактные данные клиента:
 - Контактные лица
 - Номера телефонов
 - Почтовые ящики
- Система взаимодействует с кластером k8s по rest-api(методы и шаблоны описаны ниже)
- Система должна хранить набор predetermined шаблонов запросов, для взаимодействия с кластером k8s
- Шаблон запроса - текст в формате JSON с макросами, которые заменяются при исполнении запроса значениями определяющими среду клиента, например: идентификатор клиента, секретные ключи, образы docker и т.д. Все поля общие для каждого клиента указываются в шаблоне "как есть" в виде текста.
- Система должна хранить варианты настроек(справочники):
 - Путь к образу сервиса
 - Идентификатор пространства имен, в котором развертывается среда: [lms-dev,

- lms-prod]
- Корневой URL api кластера k8s
- Авторизационный токен для доступа к api кластера k8s
- Публичный адрес сервиса
- Секретный rsa-ключ среды клиента
- Система должна хранить набор переменных окружения для создания/обновления? среды клиента
- Система должна хранить системные параметры доступа для postgres и clickhouse
- Система должна предоставлять возможность создания нового клиента, в процессе которого выполняет запросы:
 - Создание задания кластеру k8s на баз clickhouse и postgres
 - Загрузка переменных окружения среды клиента в кластер k8s
 - Загрузка секретов в кластер k8s
 - Создание сервиса k8s и записи ingress
 - Проверка статуса развертывания среды и вывод информации об успешности или не успешности развертывания

Доступ

- Доступ к api кластера k8s выполняется через rest-протокол. Корневой url: <https://rancher.rarus.cloud/k8s/clusters/c-m4hlh/>
- Сейчас доступна одно пространство имен lms-dev, но необходимо предусмотреть появление других в будущем
- Доступ снаружи возможен только при условии добавления IP адреса с владельцами кластера. Соответственно, перед началом тестирования запросов нужно передать эту информацию ответственному со стороны ДОС.

Шаблоны запросов

Общие параметры

api-root-url	Корневой URL api кластера k8s	https://rancher.rarus.cloud/k8s/clusters/c-m4hlh/
namespace-id	Идентификатор пространства имен кластера	[lms-dev, lms-prod]
client-id	Идентификатор среды клиента	Формируется системой по шаблону C<Номер среды>

Ниже все значения параметров указаны для пространства имен lms-dev

Создание базы clickhouse

Запрос : POST {{api-root-url}}/apis/batch/v1/namespaces/{{namespace-id}}/jobs

```
{
  "apiVersion": "batch/v1",
  "kind": "Job",
  "metadata": {
    "name": "clickhouse-setup-{{client-id}}",
    "namespace": "{{namespace-id}}"
  },
  "spec": {
    "template": {
      "metadata": {
        "labels": {
          "job-name": "clickhouse-setup-{{client-id}}"
        }
      },
      "spec": {
        "containers": [
          {
            "args": [
              "clickhouse-client --host $CLICKHOUSE_HOST --user $CLICKHOUSE_USER --password $CLICKHOUSE_PASSWORD --query \"CREATE DATABASE IF NOT EXISTS $USER_NAME;\"\\n",
              "clickhouse-client --host $CLICKHOUSE_HOST --user $CLICKHOUSE_USER --password $CLICKHOUSE_PASSWORD --query \"CREATE USER IF NOT EXISTS $USER_NAME IDENTIFIED BY '$CH_PASSWORD';\"\\n",
              "clickhouse-client --host $CLICKHOUSE_HOST --user $CLICKHOUSE_USER --password $CLICKHOUSE_PASSWORD --query \"GRANT ALL ON $USER_NAME.* TO $USER_NAME;\"\\n",
              "clickhouse-client --host $CLICKHOUSE_HOST --user $CLICKHOUSE_USER --password $CLICKHOUSE_PASSWORD --query \"GRANT POSTGRES ON *.* TO $USER_NAME WITH GRANT OPTION;\"\\n",
              "clickhouse-client --host $CLICKHOUSE_HOST --user $CLICKHOUSE_USER --password $CLICKHOUSE_PASSWORD --query \"GRANT CREATE TEMPORARY TABLE ON *.* TO $USER_NAME WITH GRANT OPTION;\"\\n"
            ],
            "command": [
              "/bin/bash",
              "-c"
            ],
            "env": [
```

```
{
  "name": "USER_NAME",
  "value": "{{client-id}}"
},
{
  "name": "CLICKHOUSE_PASSWORD",
  "value": "{{clickhouse-admin-password}}"
},
{
  "name": "CLICKHOUSE_USER",
  "value": "{{clickhouse-admin-user}}"
},
{
  "name": "CLICKHOUSE_HOST",
  "value": "{{clickhouse-host}}"
},
{
  "name": "CH_PASSWORD",
  "value": "{{clickhouse-user-password}}"
}
],
"image": "yandex/clickhouse-client",
"imagePullPolicy": "Always",
"name": "clickhouse-setup"
}
],
"restartPolicy": "Never"
}
}
}
```

Параметры:

clickhouse-admin-password	Пароль администратора clickhouse	ABWsBh+Q	
clickhouse-user-password	Пароль пользователя базы клиента	Генерируется системой	

Создание базы postgres

Запрос : **POST {{api-root-url}}/apis/batch/v1/namespaces/{{namespace-id}}/jobs**

```
{
  "apiVersion": "batch/v1",
  "kind": "Job",
  "metadata": {
    "name": "postgres-setup-{{client-id}}",
    "namespace": "{{namespace-id}}"
  },
  "spec": {
    "template": {
      "metadata": {
        "labels": {
          "job-name": "postgres-setup-{{client-id}}"
        }
      },
      "spec": {
        "containers": [
          {
            "args": [
              "export PGPASSWORD=$PGPASSWORDADMIN\npsql -h $POSTGRES_HOST -U $POSTGRES_USER\n-c \"CREATE USER $USER_NAME WITH PASSWORD ' \\ '$PG_PASSWORD' \\';\" || true\npsql -h\n$POSTGRES_HOST -U $POSTGRES_USER -c \"CREATE DATABASE $USER_NAME WITH OWNER $USER_NAME;\" ||\ntrue\npsql -h $POSTGRES_HOST -U $POSTGRES_USER -c \"GRANT ALL PRIVILEGES ON DATABASE\n$USER_NAME TO $USER_NAME;\" || true\npsql -h $POSTGRES_HOST -U $POSTGRES_USER -c \"ALTER ROLE\n$USER_NAME SUPERUSER;\" || true\nexport PGPASSWORD=$PG_PASSWORD\npsql -h $POSTGRES_HOST -U\n$USER_NAME -c \"CREATE EXTENSION IF NOT EXISTS pgcrypto;\"\npsql -h $POSTGRES_HOST -U\n$USER_NAME -c \"CREATE EXTENSION IF NOT EXISTS \\\"\"uuid-osspl\\\";\"\nexport\nPGPASSWORD=$PGPASSWORDADMIN\npsql -h $POSTGRES_HOST -U $POSTGRES_USER -c \"ALTER ROLE\n$USER_NAME WITH NOSUPERUSER;\" \n\n"
            ],
            "command": [
              "/bin/bash",
```

```

        "-c"
    ],
    "env": [
        {
            "name": "PGPASSWORDADMIN",
            "value": "{{postgres-admin-password}}"
        },
        {
            "name": "USER_NAME",
            "value": "{{client-id}}"
        },
        {
            "name": "POSTGRES_USER",
            "value": "{{postgres-admin-user}}"
        },
        {
            "name": "POSTGRES_HOST",
            "value": "{{postgres-host}}"
        },
        {
            "name": "PG_PASSWORD",
            "value": "{{postgres-user-password}}"
        }
    ],
    "image": "postgres:latest",
    "imagePullPolicy": "Always",
    "name": "postgres-client"
}
],
"restartPolicy": "Never"
}
}
}
}

```

Параметры:

postgres-admin-user	Администратор postgres	postgres	
postgres-admin-password	Пароль администратора postgres	8HgbfhiNFRstbC0NMtd	
postgres-user-password	Пароль пользователя базы клиента	Генерируется системой	

Создание/изменение? конфигурации

Запрос : POST {{api-root-url}}/api/v1/namespaces/{{namespace-id}}/configmaps

```
{
  "apiVersion": "v1",
  "data": {
    "LMS_CLIENT_ID": "{{client-id}}",

    "LMS_SERVER_DEBUG": "true",
    "LMS_SERVER_PORT": ": 8080",
    "LMS_SERVER_STATIC": "/mnt/image/static",

    "LMS_KEYS_SECRET_SEED": "{{keys-secret-seed}}",
    "LMS_KEYS_TOKEN_TTL_HOURS": "{{keys-token-ttl-hours}}",
    "LMS_KEYS_PRIVATE_KEY_PATH": "{{keys-rsa-path}}"

    "LMS_LOGS_DIRECTORY": "logs",
    "LMS_LOGS_FILENAME": "log.log",
    "LMS_LOGS_LOG_TO_FILE": "false",
    "LMS_LOGS_MAX_AGE_DAY": "1",
    "LMS_LOGS_MAX_BACKUPS": "3",
    "LMS_LOGS_MAX_SIZE_MB": "1",

    "LMS_NOTIFICATIONS_SMS4B_PASSCODE_LIFETIME": "{{sms4b-passcode-lifetime-seconds}}",
    "LMS_NOTIFICATIONS_SMS4B_SENDER": "{{sms4b-sender}}",
    "LMS_NOTIFICATIONS_SMS4B_TOKEN": "{{sms4b-token}}",

    "LMS_STORAGE_CLICKHOUSE_DEMO": "internal/server/demodata/clickhouse",
    "LMS_STORAGE_CLICKHOUSE_DSN": "clickhouse://{{client-id}}:{{clickhouse-user-password}}@{{clickhouse-host}}:9000/{{client-id}}",
    "LMS_STORAGE_CLICKHOUSE_INTERVAL_MS": "100",
```

```

"LMS_STORAGE_CLICKHOUSE_MIGRATIONS": "file: //internal/server/migrations/clickhouse",
  "LMS_STORAGE_CLICKHOUSE_UPDATE_ATTEMPTS": "3",

  "LMS_STORAGE_POSTGRES_DEMO": "internal/server/demodata/postgres",
  "LMS_STORAGE_POSTGRES_DSN": "postgres: //{{client-id}}: {{postgres-user-
password}}@{{postgres-host}}: 5432/{{client-id}}?sslmode=disable",
  "LMS_STORAGE_POSTGRES_INTERVAL_MS": "100",
  "LMS_STORAGE_POSTGRES_MIGRATIONS": "file: //internal/server/migrations/postgres",
  "LMS_STORAGE_POSTGRES_UPDATE_ATTEMPTS": "3"
},
"kind": "ConfigMap",
"metadata": {
  "name": "lms-config-{{client-id}}",
  "namespace": "{{namespace-id}}"
}
}

```

Параметры:

Имя параметра	Описание	lms-dev	lms-prod
keys-secret-seed	Ключ для формирования подписи токенов авторизации. Генерируется системой как 64-байтная последовательность записанная в шестнадцатиричной системе	Генерируется для каждого клиента	Генерируется для каждого клиента
keys-token-ttl-hours	Время жизни токена доступа пользователя	8 часов	8 часов
sms4b-passcode-lifetime-seconds	Время жизни временного кода подтверждения телефона клиента	120	Зависит от пожеланий клиента
sms4b-sender	Отправитель сервисных смс-сообщений	1C-RARUS	?
sms4b-token	Токен доступа к api sms4b	В личном кабинете sms4b	В личном кабинете sms4b
keys-rsa-path	Путь к файлу закрытого ключа среды клиента	/etc/lms/rsa_key	/etc/lms/rsa_key

Создание закрытого ключа среды клиента

Запрос : **POST** `{{api-root-url}}/api/v1/namespaces/{{namespace-id}}/secrets`

```
{
  "apiVersion": "v1",
  "data": {
    "rsa_key": "{{rsa-content}}"
  },
  "kind": "Secret",
  "metadata": {
    "name": "rsa-secret-{{client-id}}",
    "namespace": "{{namespace-id}}"
  },
  "type": "Opaque"
}
```

Параметры:

Имя параметра	Описание	lms-dev	lms-prod
rsa-content	2048-байтный rsa-ключ в кодировке base64	Генерируется для каждого клиента	Генерируется для каждого клиента

Создание сервиса

Запрос : **POST** `{{api-root-url}}/api/v1/namespaces/{{namespace-id}}/services`

```
{
  "apiVersion": "v1",
  "kind": "Service",
  "metadata": {
    "name": "lms-service-{{client-id}}",
    "namespace": "{{namespace-id}}"
  },
  "spec": {
    "ports": [
      {
        "port": 8080,
        "protocol": "TCP",

```

```

        "targetPort": 8080
    }
],
"selector": {
    "app": "{{client-id}}-lms"
},
"type": "ClusterIP"
}
}

```

Создание записи ingress

Запрос : POST {{api-root-url}}/apis/networking.k8s.io/v1/namespaces/{{namespace-id}}/ingresses

```

{
  "apiVersion": "networking.k8s.io/v1",
  "kind": "Ingress",
  "metadata": {
    "annotations": {
      "nginx.ingress.kubernetes.io/configuration-snippet": "add_header ' ' Access-Control-Allow-Methods' ' ' ' GET, PUT, POST, OPTIONS, DELETE' ' '; \nadd_header ' ' Access-Control-Allow-Headers' ' ' ' DNT, X-CustomHeader, Keep-Alive, User-Agent, X-Requested-With, If-Modified-Since, Cache-Control, Content-Type, Authorization' ' '; \nadd_header ' ' Access-Control-Expose-Headers' ' ' ' Content-Length, Content-Range' ' ' ;",
      "nginx.ingress.kubernetes.io/enable-cors": "true",
      "nginx.ingress.kubernetes.io/rewrite-target": "/$1"
    },
    "name": "client-lms-ingress-{{client-id}}",
    "namespace": "{{namespace-id}}"
  },
  "spec": {
    "rules": [
      {
        "host": "{{public-dns}}",
        "http": {
          "paths": [
            {
              "backend": {
                "service": {

```

```

        "name": "lms-service-{{client-id}}",
        "port": {
            "number": 8080
        }
    },
    "path": "/{{client-id}}/(.*)",
    "pathType": "Prefix"
}
]
}
},
],
"tls": [
    {
        "hosts": [
            "{{public-dns}}"
        ],
        "secretName": "tls"
    }
]
}
}

```

Параметры:

Имя параметра	Описание	lms-dev	lms-prod
public-dns	Публичное имя сервиса	backend.lms-dev.rarus-cloud.ru	?

Создание развертывания

Запрос : POST {{api-root-url}}/apis/apps/v1/namespaces/{{namespace-id}}/deployments

```

{
    "apiVersion": "apps/v1",
    "kind": "Deployment",
    "metadata": {
        "labels": {

```

```

        "app": "{{client-id}}-lms"
    },
    "name": "lms-deployment-{{client-id}}",
    "namespace": "{{namespace-id}}"
},
"spec": {
    "replicas": 1,
    "selector": {
        "matchLabels": {
            "app": "{{client-id}}-lms"
        }
    },
    "strategy": {
        "rollingUpdate": {
            "maxSurge": 1,
            "maxUnavailable": 1
        },
        "type": "RollingUpdate"
    },
    "template": {
        "metadata": {
            "labels": {
                "app": "{{client-id}}-lms"
            }
        },
        "spec": {
            "affinity": {
                "nodeAffinity": {
                    "requiredDuringSchedulingIgnoredDuringExecution":
{
                    "nodeSelectorTerms": [
                        {
                            "matchExpressions": [
                                {
                                    "key": "rarus",
                                    "operator": "In",
                                    "values": [
                                        "lms"
                                    ]
                                }
                            ],
                            "containers": [

```

```

{
  "envFrom": [
    {
      "configMapRef": {
        "name": "lms-config-{{client-
id}}"

      }
    }
  ],
  "image": "{{docker-image}}",
  "imagePullPolicy": "Always",
  "name": "{{client-id}}",
  "ports": [
    {
      "containerPort": 8080,
      "name": "8080tcp",
      "protocol": "TCP"
    }
  ],
  "volumeMounts": [
    {
      "mountPath": "/etc/lms",
      "name": "key"
    },
    {
      "mountPath":
"/mnt/image/static",

      "name": "data",
      "subPath": "{{client-id}}"
    }
  ]
},
"imagePullSecrets": [
  {
    "name": "repository-token"
  }
],
"restartPolicy": "Always",
"volumes": [

```

```
        {
            "name": "data",
            "nfs": {
                "path": "/mnt/nfs",
                "server": "{{nfs-server}}"
            }
        },
        {
            "name": "key",
            "secret": {
                "defaultMode": 511,
                "secretName": "rsa-secret-{{client-id}}"
            }
        }
    ]
}
}
```

Параметры:

Имя параметра	Описание	lms-dev	lms-prod
docker-image	docker-образ, из которого будут разворачиваться новые среды клиентов	registry.gitlab.corp.rarus.cloud/rarus-lms/backend:dev	?
nfs-server	Сервер сетевой файловой системы	10.66.139.73	?

Мониторинг pod-a

Запрос : GET {{api-root-url}}/api/v1/namespaces/{{namespace-id}}/pods?labelSelector=app={{client-id}}-lms

В зависимости от статуса в ответе будет структура **state**, по которой нужно анализировать статус pod-a

Revision #15

Created 2 April 2024 08:12:25 by Морозов Сергей

Updated 15 January 2025 08:29:58 by Морозов Сергей