

Сведения об установке экземпляра программы

«Система динамического управления диапазонами SIM-карт Forward DSA»

Правообладатель:
ООО «Форвард-Телеком»

Forward DSA (Dynamic SIM Allocation) – является информационной системой динамического управления диапазонами SIM-карт. DSA дает возможность производить и распространять один общий тип SIM-карт для любого типа мобильных услуг: предоплаченные, постоплатные, SIM-карты для передачи данных. Это позволяет операторам распространять SIM-карты по разным каналам с использованием одного общего типа SIM-карты и предоставлять их в сети только в случае необходимости отказавшись от предварительной подготовки SIM-карт.

С помощью решения DSA операторы мобильной связи могут оптимизировать процесс производства и распространения SIM-карт за счет внедрения эффективной логистики, значительной экономии средств и устранения зависимости от срока действия SIM-карт. Система поддерживает кластерное масштабирование и может использоваться в распределенных сетях крупных федеральных и региональных операторов связи.

Контактная информация: support@fw-t.ru

1 Дистрибутив

1.1 Распространение дистрибутива

Дистрибутив комплекта программ распространяется на физических носителях (USB-flash накопителе) или по ссылке на облачное хранилище.

Дистрибутив формируется для каждого заказчика с учетом технологической экосистемы и необходимых модулей расширения функционала.

Подробная документация поставляется после адаптации системы в процессе внедрения с учетом особенностей кастомизации.

1.2 Содержание дистрибутива

Перечень устанавливаемых компонентов осуществляется в соответствии со спецификацией поставки.

Дистрибутивы поставляются как docker-образ и обеспечивают в дальнейшем корректную установку DSA.

При необходимости, отдельно передаются дополнительные файлы (.yml, .env).

2 Порядок установки программного обеспечения

Модуль DSA устанавливается из докер-образа.

Специалистами Форвард-Телеком проводится групповое обучение специалистов Заказчика по разворачиванию и настройкам системы в целях обеспечения её дальнейшей корректной эксплуатации.

2.1 Подготовка к установке

Перед установкой DSA необходимо создать Postgres схему.

Дополнительно, для корректной работы перед запуском DSA в биллинговой системе:

- Должна быть заведена группа идентификаторов для временных MSISDN: «Временные номера»
- Должен быть сформирован в системе пул виртуальных IMSI.
- Должны быть сгенерированы наборы доступных региональных номеров MSISDN.
- Должны быть сгенерированы в достаточном количестве региональные IMSI.
- Загружены актуальные данные в справочник Регионов.
- Добавлены вендоры SIM-карт профиля в KeyStore и их связь с регионами в специальном интерфейсе (#Flexy660092).

2.2 Установка

Порядок инсталляции DSA представляет собой последовательное выполнение действий по:

1. Получению Docker-образа DSA
2. Конфигурированию Docker compose DSA
3. Настройке DSA
4. Запуску DSA

2.2.1 Получение Docker-образа

При наличии сетевого доступа к GitLab (Container Registry) получить docker-образ и выполнить команду вида:

```
docker pull registry.gitlab.ru/group/project/release:latest
```

2.2.2 Конфигурирование Docker compose

Использовать отдельно передаваемый docker-compose.yml или создать его, содержащего данные вида:

```

version: 'X'
services:
dsa:
image: registry.gitlab.com/group/project/release:latest
container_name: dsa-module
environment:
- DB_SCHEMA=dsa_schema
- KEYSTORAGE_AUTO_GENERATE=true
volumes:
- ./config:/app/config
ports:
- "XXXX:XXXX"

```

2.2.3 Настройка DSA

Конфигурация генерируется автоматически в директории вида /dsa/config/properties

При необходимости, производится настройка следующих системных параметров:

Наименование	Комментарий
server.port	Порт приложения
spring.application.name	Название приложения
spring.datasource.hikari.connection-timeout	Таймаут подключения к БД
spring.datasource.hikari.maximum-pool-size	Размер пула подключений
app.default.lockAtMostFor	Максимальное время, на которое устанавливается распределённая блокировка shedlock
app.shedlock.lockAtMostFor	
app.scheduler.initialDelay	
app.scheduler.events.exportFixedRate	Периодичность отправки событий о загрузке нового файла
app.scheduler.events.lockAtLeastFor	Минимальное время блокировки шедулера - экспорт событий в MQ
app.async.threads	
app.private.key	Приватный RSA ключ для шифрования транспортных ключей профилей

Наименование	Комментарий
app.store.profile.extIdent	Название профиля, в котором лежит наш актуальный транспортный ключ
app.raw.types.allowed	Comma-separated список пар "Id сущности во внешней систем" и "Тип ключа", разделенных двоеточием (например ICCID:KIC1) определяющих сущности, для которых можно возвращать значения в открытом виде в API

Для кастомизации возможно использовать отдельно передаваемый или создать .env файл с переменными окружения, вида:

- DATASOURCE_URL
- DATASOURCE_USERNAME
- DATASOURCE_PASSWORD

Примечание: при первом запуске будет автоматически создана public/private пара RSA-ключей. Публичный передаётся Клиенту, с помощью него он будет шифровать транспортные ключи партнеров – для добавления в систему.

2.2.4 Запуск DSA

Для запуска системы необходимо выполнить команду вида:

```
docker compose up -d
```

После запуска рекомендуется проверить работоспособность системы. Для этого необходимо:

- проверить логи контейнера (вида `docker logs dsa-module --tail 100`)
- проверить health-check (вида `curl http://localhost:8080/health`)

Примечание: для остановки системы необходимо выполнить команду вида `docker compose stop`

2.3 Дополнительные сведения

Liquibase-миграции выполняются автоматически при запуске контейнера.

Для принудительного выполнения миграций выполнить команду вида `docker compose run dsa liquibase update`.

Операции с базой данных выполняются администратором БД.

Управление Docker-контейнерами осуществляется DevOps-специалистами Форвард.

2.4 Тестовый сценарий (пример)

2.4.1 Создание профиля (вендор)

```
curl --request POST \  
  --url http://localhost:8120/encryption/profile \  
  --header 'Content-Type: application/json' \  
  --header 'User-Agent: insomnia/10.2.0' \  
  --data '{  
    "name": "vendor1",  
    "description": "vendor1",  
    "extIdent": "vendor1"  
  }'
```

2.4.2 Создание ключа шифрования (вендор)

```
curl --request POST \  
  --url http://localhost:8120/encryption/key \  
  --header 'Content-Type: application/json' \  
  --header 'User-Agent: insomnia/10.2.0' \  
  --data '{  
    "profileId": "16d75751-47ac-4bf5-9ebe-5bf338580741",  
    "startDateTime": "2030-01-13T13:53:15+03:00",  
    "endDateTime": "2031-01-13T13:53:15+03:00",  
    "keyValue":  
    "5275cafac0b3427df39b53277b63c6a1a10d08b857b9dca264c6c5480cb426a01b53928f9a1  
2c7685fccd1692ed2d11fb0207bb1b4eba2829dd993eb4197b51ee0d841bf7b7f0ee3bb41c13  
ae75032e02a961f6be3d0844e1b90b8cb37aad1c89d84c26d2de224d6fe676a9b706c3f98f60  
327e58b7626373590b20547e03459"  
  }'
```

2.4.3 Создание профиля INTERNAL с ключом для перешифровки данных

```
curl --request POST \  
  --url http://localhost:8120/encryption/profile \  
  --header 'Content-Type: application/json' \  
  --header 'User-Agent: insomnia/10.2.0' \  
  --data '{  
    "name": "INTERNAL",  
    "description": "INTERNAL",  
    "extIdent": "INTERNAL"  
  }'  
  
curl --request POST \  
  --url http://localhost:8120/encryption/key \  
  --header 'Content-Type: application/json' \  
  --data '{  
    "profileId": "16d75751-47ac-4bf5-9ebe-5bf338580741",  
    "startDateTime": "2030-01-13T13:53:15+03:00",  
    "endDateTime": "2031-01-13T13:53:15+03:00",  
    "keyValue":  
    "5275cafac0b3427df39b53277b63c6a1a10d08b857b9dca264c6c5480cb426a01b53928f9a1  
2c7685fccd1692ed2d11fb0207bb1b4eba2829dd993eb4197b51ee0d841bf7b7f0ee3bb41c13  
ae75032e02a961f6be3d0844e1b90b8cb37aad1c89d84c26d2de224d6fe676a9b706c3f98f60  
327e58b7626373590b20547e03459"  
  }'
```

```
--header 'User-Agent: insomnia/10.2.0' \
--data '{
"profileId": "d19cbfe1-9e1d-4581-8998-e1e640ce6a02",
"startDateTime": "2030-01-13T13:53:15+03:00",
"endDateTime": "2035-01-13T13:53:15+03:00",
"keyValue":
"5275cafac0b3427df39b53277b63c6a1a10d08b857b9dca264c6c5480cb426a01b53928f9a1
2c7685fccd1692ed2d11fb0207bb1b4eba2829dd993eb4197b51ee0d841bf7b7f0ee3bb41c13
ae75032e02a961f6be3d0844e1b90b8cb37aad1c89d84c26d2de224d6fe676a9b706c3f98f60
327e58b7626373590b20547e03459"
}'
```

2.4.4 Добавление в хранилище

```
curl --request POST \
--url http://localhost:8120/encryption/store \
--header 'Content-Type: application/json' \
--header 'User-Agent: insomnia/10.2.0' \
--data '[
{
  "masterType": "ICCID",
  "masterInst": "89701190000000000806",
  "valueType": "KIK1",
  "encryptedValue": "BFF1FB08AE75E867C6745A613BE15F35",
  "profileId": "16d75751-47ac-4bf5-9ebe-5bf338580741",
  "dateTime": "2030-01-13T16:53:15+03:00"
}
]'
```