Продукт Platform V Print (DCG)

Компонент Сервис генерации документов (DCGN)

Руководство по установке

Содержание

Руководство по установке компонента DCGN (DCGN)

В руководстве приведены инструкции по установке компонента DCGN продукта Platform V Print (DCG).

Термины и определения

Список терминов и определений приведен в одноименном разделе документа «Описание функциональных характеристик».

Системные требования

Сервисы компонента DCGN представляют собой облачное решение, поставляемое для установки в контейнеризованную среду (Kubernetes или OpenShift (опционально)).

Системное программное обеспечение

Ниже представлено описание категорий системного программного обеспечения, используемых для установки и настройки DCGN. В каждой категории перечислены все поддерживаемые продукты сторонних правообладателей. Отдельно обозначены варианты, которые рекомендует АО «СберТех» (маркировка «Рекомендовано» в столбце «Продукт, функциональная совместимость с которым подтверждена»). Для установки DCGN необходимо выбрать один из продуктов в каждой категории.

Категория ПО	**Обязательность установки (да/нет)***	**Наименование ПО**	**Версия**	**Продукт, функциональная совместимость с которым подтверждена****	**Описание**
Операционная система	Да	Альт 8 СП	altlinux-sp8-5.4.113-release	Рекомендовано	ОС контейнеров для запуска модулей компонента
		Red Hat Enterprise Linux	3.10 и выше	Опционально	ОС контейнеров для запуска модулей компонента
Среда контейнеризации	Да	Kubernetes	1.21.0	Рекомендовано	Платформа контейнеризации для запуска компонентов сервиса
		Red Hat OpenShift	1.20.0	Опциоально	Платформа контейнеризации для запуска компонентов сервиса
Инструмент сборки, тестирования, развертывани	Да	Platform V DevOps Tools (CDJE)	D-01.038.862-443	Рекомендовано	Компонент для развертывания Платформы и бизнес приложений на тестовые и

Категория ПО	Обязательность установки (да/нет)*	Наименование ПО	Версия	Продукт, функциональная совместимость с которым подтверждена**	Описание
я контейеризированных приложений					промышленные стенды
Java-машина	Да	OpenJDK	11	Рекомендовано	Окружение для работы модулей компонента
		OracleJDK	1.8 и выше	Опционально	Окружение для работы модулей компонента
Система управления базами данных (СУБД)	Да	PostgreSQL	11 и выше	Рекомендовано. Правообладателем АО «СберТех» рекомендована СУБД, основанная на PostgreSQL – Platform V Pangolin SE, см. раздел «Платформенные зависимости»	ПО, взаимодействующее с конечными пользователями, приложениями и базой данных для сбора и анализа данных
Сервер приложений	Нет	Apache Tomcat	9.0.45	Рекомендовано	СПО для тестирования, отладки и исполнения веб-приложений на основе Java. Не требуется отдельное развертывание инструмента, Сервер приложений поставляется в составе компонента DCGN
Браузер	Нет	Яндекс.Браузер для ОС Windows	не ниже 21.9.0	Рекомендовано	Браузер для входа в UI
	Нет	Яндекс.Браузер для MAC OS	не ниже 21.2.0	Рекомендовано	Браузер для входа в UI
	Нет	SberBrowser	не ниже 2.1.0	Опционально	Браузер для входа в UI
	Нет	Mozilla Firefox	не ниже 98.0.0	Опционально	Браузер для входа в UI

Категория ПО	Обязательность установки (да/нет)*	Наименование ПО	Версия	Продукт, функциональная совместимость с которым подтверждена**	Описание
	Нет	Safari	не ниже 15.2.0	Опционально	Браузер для входа в UI
	Нет	Google Chrome	не ниже 98.0.4758	Опционально	Браузер для входа в UI
Сервис централизованного хранения репозиториев артефактов (хранилище артефактов)	Да	Nexus-Public	2.5.1 и выше	Рекомендовано	Хранение конфигураций при автоматизированной установке
Сервис интеграции и оркестрации микросервисов в облаке	Да	Istio	2.0.6-2	Рекомендовано. Правообладателем АО «СберТех» такжерекомендован сервис интеграции и оркестрации микросервисов в облаке, снованный на Istio – Platform V Synapse Service Mesh (SSM), см. раздел «Платформенные зависимости»	Сервис интеграции микросервисов в облаке

Примечание:

*

- **Да** — категория ПО обязательна для функционирования сервиса (это означает, что сервис не может выполнять свои основные функции без установки данной категории ПО).
- **Нет** — категория ПО необязательна для функционирования сервиса (это означает, что сервис может выполнять свои основные функции без установки данной категории ПО).

**

- **Рекомендовано** — рекомендованный правообладателем АО «СберТех» продукт.
- **Опционально** — альтернативный по отношению к рекомендованному правообладателем АО «СберТех» продукт.

Здесь и далее поддерживаемой системой приложений-контейнеров является Kubernetes (использование OpenShift – опционально), в инструкциях по настройке, в

именах переменных и параметрах системы могут встречаться названия систем контейнеризации, которые одинаковы и применимы для обоих сред контейнеризации.

Платформенные зависимости

Для настройки, контроля и функционирования компонента реализована интеграция с программными продуктами, правообладателем которых является АО «СберТех»:

Наименование продукта	**Код продукта**	**Версия**	**Код и наименование компонента**	**Обязательность установки**	**Описание**
Platform V Audit SE	AUD	2.1	AUDT Аудит	опционально	Компонент для аудирования событий
Platform V Monitor	OPM	4.1	LOGA Журналирование	опционально	Компонент для хранения лог-файлов
Platform V Monitor	OPM	4.1	MONA Объединенный мониторинг Unimon	опционально	Компонент для сбора прикладных и инфраструктурных метрик и отправки их в целевую систему хранения
Platform V Pangolin SE	PSQ	5.1.0	-	опционально	Система управления базами данных, основанная на PostgreSQL
Platform V DevOps Tools	DOT	1.2	CIJE Build Tools	опционально	Компонент для автоматизации сборки дистрибутивов сервисов Платформы, бизнес-приложений и bundle Solution/MonoSolution
Platform V DevOps Tools	DOT	1.2	CDJE Deploy Tools	опционально	Компонент для развертывания Платформы и бизнес приложений на тестовые и промышленные стенды
Platform V Backend	#BD	4.3	OTTS One-Time Password (OTP) / OTT	опционально	Компонент для аутентификации и авторизации межсервисных взаимодействий
Platform V IAM SE	IAM	1.3	AUTZ Объединенный	обязательно	Компонент для управления

Наименование продукта	Код продукта	Версия	Код и наименование компонента	Обязательность установки	Описание
			сервис авторизации (OCA)		доступом к информационным ресурсам, необходим для авторизации пользователей
Platform V IAM SE	IAM	1.3	AUTH IAM Proxy	обязательно	Компонент для управления доступом к информационным ресурсам, необходим для аутентификации пользователей
Platform V Synapse Service Mesh	SSM	2.10	IGEG Граничный прокси	обязательно	Компонент для управления запросами (трафик), приходящих и исходящих из проекта одной системы

Примечание:

- **Обязательно** — компонент или продукт необходим для функционирования компонента DCGN.
- **Опционально** — необязательный для функционирования компонент или продукт, рекомендуется его установка, но допускается использование аналога других производителей.

Тип зависимости определяется:

- One-Time Password (OTP) / OTT (OTTS) — можно отключить на уровне конфигурационых файлов ingress/egress. Управляется поставкой разных дистрибутивов конфигураций. Критично только для инсталляций с обязательным требованием использования One-Time Password (OTP) / OTT.
- Deploy tools (CDJE) — рекомендуется установка дистрибутива с помощью Deploy Tools, при желании можно использовать другой инструмент развертывания.
- Объединенный мониторинг Unimon (MONA) — устанавливаются при необходимости. Поставляется отдельным дистрибутивом MONA, установка в namespace выполняется через Deploy Tools.
- Журналирование (LOGA) — допустимо сконфигурировать отправку логов в другую систему журналирования.
- Аудит (AUDT) — DCGN может взаимодействовать с платформенным компонентом Аудит.
- IAM proxy (AUTH), Объединенный сервис авторизации (OCA) (AUTZ) — обязательны для работы через пользовательский интерфейс.

- Platform V Synapse Service Mesh (SSM) — обязательна к установке указанная версия и сборка, если инсталляция с One-Time Password (OTP) / OTT и для некоммунальной инсталляции.

Аппаратные требования

Список минимальных требований к ресурсам КТС приведен ниже. Допустимы два способа развертывания DCGN - в коммунальную и автономную (изолированную) инсталляции.

Коммунальная инсталляция предполагает установку в один namespase продукта (DCGN) совместно с продуктами Платформы.

Автономная инсталляция предполагает установку в один namespase одного продукта (DCGN).

Конфигурация БД для тестовых стендов

№	ПО	CPU Тест	MEM, GB Тест	HDD, GB Тест
1	PostgreSQL Master	4	16	300 (/pgdata 230 GB /pgarclogs 50 GB)

Конфигурация БД для НТ, ПРОМ стендов

Требуется наличие кластера Patroni (PostgreSQL + pgbouncer + Patroni).

№	ПО	CPU НТ/Пром значение	MEM, GB НТ/Пром значение	HDD, GB НТ/Пром значение
1	PostgreSQL Master	16	128	400 (/pgdata 300 GB /pgarclogs 50 GB)
2	PostgreSQL Slave	16	128	400 (/pgdata 300 GB /pgarclogs 50 GB)
3	Patroni (арбитр)	4	16	200 (/pgdata 150 GB /pgarclogs 30 GB)

Требование квот кластера (коммунальная инсталляция) для НТ, ПРОМ стендов

№	ПО	CPU Минимальное значение (тестовый стенд)	CPU Пром значение (НТ, ПРОМ стенды)	MEM, GB Минимальное значение (тестовый стенд)	MEM, GB Пром значение (НТ, ПРОМ стенды)
1	K8s или OSE (опционально)	16	24	32	38

Требование квот кластера (автономная инсталляция)

№	ПО	CPU Минимальное значение	CPU Пром значение	MEM, GB Минимальное значение	MEM, GB Пром значение
1	K8s или OSE (опционально)	16	43	32	57

Ресурсы, необходимые для сервисов компонента DCGN

Автономная инсталляция:

Имя Pod	**Имя сервиса**	**Кол-во реплик**	**CPU(lim)**	**CPU(req)**	**Mem(lim)**	**Mem(req)**	**Итог CPU на Pod**	**Итог Mem на Pod**
Docgen-service	docgen-service	2	3500m	2500m	6144Mi	3072Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Docgen-service	fluent-bit	2	200m	200m	512Mi	512Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Docgen-service	istio	2	300m	200m	256Mi	256Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Egress	istio	2	1100m	200m	1024Mi	800Mi	1650m С учетом кол. подов = 3200m	2048 Mi С учетом кол. подов = 4096 Mi
Egress	OTTS	2	550m	300m	1024Mi	512Gi	1650m С учетом кол. подов = 3200m	2048 Mi С учетом кол. подов = 4096 Mi
Ingress	istio	3	2800m	1000m	1024Mi	800Mi	4000m С	3072 Mi

Имя Pod	Имя сервиса	Кол-во реплик	CPU(lim)	CPU(req)	Mem(lim)	Mem(req)	Итог CPU на Pod	Итог Mem на Pod
							учетом кол. подов = 12000m	С учетом кол. подов = 9216 Mi
Ingress	OTTS	3	1200m	700m	2048Mi	1024Mi	4000m С учетом кол. подов = 12000m	3072 Mi С учетом кол. подов = 9216 Mi
Template-provider	template-provider	2	2700m	1200m	2500Mi	1500Mi	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7084 Mi
Template-provider	fluent-bit	2	200m	200m	512Mi	512Mi	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7048 Mi
Template-provider	istio	2	1100m	600m	512Mi	512MI	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7048 Mi
Template-registry	template-registry	2	2500m	1250m	4096Mi	2048Mi	3000m С учетом кол. подов = 6000m	4864 Mi С учетом кол. подов = 9728 Mi
Template-registry	fluent-bit	2	200m	200m	512Mi	512Mi	3000m С учетом	4864 Mi С

Имя Pod	Имя сервиса	Кол-во реплик	CPU(lim)	CPU(req)	Mem(lim)	Mem(req)	Итог CPU на Pod	Итог Mem на Pod
							кол. подов = 6000m	учетом кол. подов = 9728 Mi
Template-registry	istio	2	300m	200m	256Mi	256Mi	3000m С учетом кол. подов = 6000m	4864 Mi С учетом кол. подов = 9728 Mi
Unimon-agent	unimon-agent	1	500m	500m	4096Mi	2048Mi	1000m С учетом кол. подов = 1000m	4808 Mi С учетом кол. подов = 4808 Mi
Unimon-agent	logger-forward-sidecar	1	200m	100m	200Mi	100Mi	1000m С учетом кол. подов = 1000m	4808 Mi С учетом кол. подов = 4808 Mi
Unimon-agent	istio	1	300	200	512Mi	256Mi	1000m С учетом кол. подов = 1000m	4808 Mi С учетом кол. подов = 4808 Mi
Unimon-sender	unimon-sender	1	500m	200m	1500Mi	1500Mi	700m С учетом кол. подов = 700m	1700Mi С учетом кол. подов = 1700Mi
Unimon-sender	logger-forward-sidecar	1	200m	100m	200Mi	100Mi	700m С учетом кол.	1700Mi С учетом кол. подов

Имя Pod	Имя сервиса	Кол-во реплик	CPU(lim)	CPU(req)	Mem(lim)	Mem(req)	Итог CPU на Pod	Итог Mem на Pod
							подов = 700m	= 1700Mi

Коммунальная инсталляция:

Имя Pod	Имя сервиса	Кол-во реплик	CPU(lim)	CPU(req)	Mem(lim)	Mem(req)	Итог CPU на Pod	Итог Mem на Pod
Docgen-service	docgen-service	2	3500m	2500m	6144Mi	3072Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Docgen-service	fluent-bit	2	200m	200m	512Mi	512Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Docgen-service	istio	2	300m	200m	256Mi	256Mi	4000m С учетом кол. подов = 8000m	6912 Mi С учетом кол. подов = 13824 Mi
Template-provider	template-provider	2	2700m	1200m	2500Mi	1500Mi	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7084 Mi
Template-provider	fluent-bit	2	200m	200m	512Mi	512Mi	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7048 Mi

Имя Pod	Имя сервиса	Кол-во реплик	CPU(lim)	CPU(req)	Mem(lim)	Mem(req)	Итог CPU на Pod	Итог Mem на Pod
Template-provider	istio	2	1100m	600m	512Mi	512MI	4000m С учетом кол. подов = 8000m	3524 Mi С учетом кол. подов = 7048 Mi
Template-registry	template-registry	2	2500m	1250m	4096Mi	2048Mi	3000m С учетом кол. подов = 6000m	4864 Mi С учетом кол. подов = 9728 Mi
Template-registry	fluent-bit	2	200m	200m	512Mi	512Mi	3000m С учетом кол. подов = 6000m	4864 Mi С учетом кол. подов = 9728 Mi
Template-registry	istio	2	300m	200m	256Mi	256Mi	3000m С учетом кол. подов = 6000m	4864 Mi С учетом кол. подов = 9728 Mi

Состав дистрибутива

Дистрибутив компонента DCGN представляет собой ZIP-архив (как для коммунальной, так и для автономной инсталляций):

№	Директория	Описание
1	./package/bh	Содержит бинарные файлы сервисов (JAR-артефакты)
2	./package/conf	Содержит набор конфигураций, необходимых для работы сервиса (yml-конфигурации, необходимые для развертывания сервисов в Kubernetes или OpenShift (опционально))
3	./package/db	Содержит Liquibase-скрипты миграции БД
4	./package/data/security	Содержит данные для импорта в Объединенный сервис авторизации (ОСА)

Настройки DCGN, в том числе рекомендуемые настройки безопасности окружения, приведены в настоящем документе, а также в документе «Руководство по безопасности».

Подготовка окружения

Для установки и работы DCGN необходимо провести подготовительные мероприятия, а именно, подготовить КТС. Подготовка КТС подразумевает получение кластера БД и namespase K8s или OSE (опционально). Размер квот соответствует указанному в разделе «Аппаратные требования» настоящего документа.

До развертывания DCGN необходимо выполнить развертывание сервисов и инфраструктурных компонентов Platform V. Наличие платформенных зависимостей приведено выше в разделе «Платформенные зависимости» настоящего документа. Развертывание сервисов и инфраструктурных компонентов Platform V производится в соответствии с документацией на установку соответствующего сервиса или инфраструктурного компонента.

Установка

Рекомендации по наполнению продукта зависимостями при сборке с помощью Build Tools (Solution Merger Job)

Образы контейнеров не входят в состав дистрибутива компонента DCGN. Базовые образы и образы sidecar необходимо указывать при сборке продукта используя Solution Merger Job. Перед запуском инструмента дополнения продукта зависимостями в конфигурации необходимо указать, какие базовые образы следует использовать вместо образов, указанных в Dockerfile по умолчанию.

> **Важно.** Образы определяются по правилам инсталляции, в которую будет установлен собранный продукт.

Базовый образ

Рекомендуемый стек:

1. Альт 8 СП
2. OpenJDK версии 11:7.6.
3. Пакет для управления шрифтами «fontconfig».

Альтернативный (опциональный) стек:

1. RHEL7.
2. OpenJDK версии 11:7.6.
3. Пакет для управления шрифтами «fontconfig».

Пример из merger.yml

```
base_image_mapping:         # маппинг базовых docker образов
    - from: .*openjdk11.*
      to: <укажите базовый образ необходимый для инсталляции> # пример nexus_
host/repository/base/rhel7openjdk11:7.6-252.1561619826-86
```

Образы sidecar контейнеров

В конфигурации развертывания DCGN присутствуют ссылки на образы sidecar контейнеров, Dockerfile которых отсутствует в дистрибутиве продукта (образы технологических сервисов, поставляемых в других платформенных продуктах), поэтому в конфигурации Solution Merger Job следует явно указать какие хеши образов использовать для таких sidecar контейнеров. Необходимо указывать образы sidecar контейнеров соответствующие версиям, указанным в данном руководстве в разделе «Платформенные зависимости».

Пример из merger.yml

```
image_link_mapping:         # маппинг ссылок на образы (применяется, чтобы ука
зать актуальные ссылки на образы sidecar контейнеров)
  # IGEG (Istio SE)
  ":.+proxyv2@sha256:[0-9a-f]{64}": ": <укажите образ необходимый для инсталл
яции>" # пример nexus_host/repository/synapse_security/istio/proxyv2@sha256:c
dd42336b164955a0550a8ce338b577177dcdd975799af3d0d8a7352f2d20fb9
  # OTTS 4.1.8, если необходима интеграция с One-Time Password (OTP) / OTT
  ":.+ott-client-api@sha256:[0-9a-f]{64}": ": <укажите образ необходимый для
инсталляции>" # пример nexus_host/repository/ott-client-api@sha256:7640852159
41ecb7e90429326113fe63b78b024e8595ac3249c5d88d128574ac
  # LOGA fluent-bit
  ":.+fluent-bit@sha256:[0-9a-f]{64}": ": <укажите образ необходимый для инст
алляции>" # пример nexus_host/repository/ulogger/fluent-bit@sha256:04da83ed2a
f92600f7e0b4055d707c038c6e549ed14dba6372b0a2a50ddae32d
```

Автоматизированная установка DCGN с использованием Deploy Tools

Установка DCGN включает в себя следующие этапы:

1. Создание пользователя и схемы БД.
2. Создание namespace в среде контейнеризации.
3. Создание и настройка сертификатов.
4. Настройка параметров.
5. Установка DCGN в контейнеризованную среду (Kubernetes или OpenShift (опционально)).

В разделах ниже приведено более подробное описание по каждому этапу установки.

Создание пользователя и схемы БД

Для создания пользователя и схемы БД необходимо выполнить скрипт согласно указанному примеру:

Пример создания пользователя и схемы

```
create user dcgn_<BLOCK_ID> with encrypted password '<пароль>';
create schema dcgn_<BLOCK_ID>;
grant connect on database postgres to dcgn_<BLOCK_ID>;
grant all on schema dcgn_<BLOCK_ID> to dcgn_<BLOCK_ID>;
alter user dcgn_<BLOCK_ID> VALID UNTIL 'INFINITY';
grant usage on schema dcgn_<BLOCK_ID> to dcgn_<BLOCK_ID>;

create tablespace dcgn_ts_data owner dcgn location '/pgdata/ts/dcgn_ts_data';
create tablespace dcgn_ts_idx owner dcgn location  '/pgdata/ts/dcgn_ts_idx';
create tablespace dcgn_ts_lob owner dcgn location '/pgdata/ts/dcgn_ts_lob';
```

В указанном примере параметр «BLOCK_ID» — постфикс блока/контура, в который устанавливается DCGN.

Если установка производится инструментами **Deploy Tools**, то значение параметра можно посмотреть в файле **ansible/common.conf.yml** common-репозитория блока/контура.

При установке компонента DCGN инструментами **Deploy Tools**, создание пользователя и схемы БД можно произвести в полуавтоматическом режиме, смотрите раздел «Инициализация БД».

> **Примечание.**
>
> **Рекомендуемое значение параметров**
>
> для БД - **max_connection=200**
>
> для pgBouncer значение **max_client_conn=2000 (max_connection*10)**

Создание namespace в среде контейнеризации

На этом шаге нужно создать namespase в кластере K8s или OSE (опционально), в котором будут создаваться объекты, необходимые для работы DCGN.

Созданный namespase необходимо подключить к Istio Service mesh или OpenShift Service mesh (опционально).

Создание и настройка сертификатов

Сертификаты mTLS

В поставляемой конфигурации DCGN настроен mTLS. Поэтому необходимо выпустить egress и ingress сертификаты для блока/контура, где производится

установка DCGN. Сертификаты необходимо поместить в защищенное хранилище и его параметры передать в соответствующие конфигурации DCGN.

Параметр	**Описание**
dcgn.ssl.ose.istio.keyStore.egress.keyStoreFromFile	Путь до хранилища сертификатов egress
dcgn.ssl.ose.istio.keyStore.egress.password	Пароль от хранилища
dcgn.ssl.ose.istio.keyStore.egress.certAlias	Alias сертификата для egress
dcgn.ssl.ose.istio.keyStore.egress.rootCertAlias	Alias для сертификата корневого УЦ
dcgn.ssl.ose.istio.keyStore.ingress.keyStoreFromFile	Путь до хранилища сертификатов ingress
dcgn.ssl.ose.istio.keyStore.ingress.password	Пароль от хранилища
dcgn.ssl.ose.istio.keyStore.ingress.certAlias	Alias сертификата для ingress
dcgn.ssl.ose.istio.keyStore.ingress.rootCertAlias	Alias для сертификата корневого УЦ

Авторизация внешних/внутренних вызовов

В поставляемой конфигурации DCGN настроен механизм аутентификации и авторизации (входящие/исходящих вызовы). Поэтому необходимо добавить параметр, который проверяет сертификат клиента на соответствие регулярному выражению.

Параметр	**Описание**
dcgn.ose.istio.ingress.envoy_filter.regex_match.client.cn	Регулярное выражение для проверки CN (common name) входящих SSL-сертификатов

Сертификаты для One-Time Password (OTP) / OTT

В поставляемой конфигурации DCGN подключен компонент Платформы One-Time Password (OTP) / OTT, опционально. Для взаимодействия с компонентом One-Time Password (OTP) / OTT необходимо выпустить сертификаты для блока/контура, где производится установка DCGN. Сертификаты необходимо поместить в защищенное хранилище и его параметры передать в соответствующие конфигурации DCGN.

Параметр	**Описание**
ott-sidecar.ssl.ose.keyStore.ott.truststore.keyStoreFromFile	Путь до хранилища с сертификатом OTTS Сервиса
ott-sidecar.ssl.ose.keyStore.ott.keyStoreFromFile	Путь до хранилища с сертификатом OTTS DCGN
ott-sidecar.ssl.ose.keyStore.ott.certAlias	Alias сертификата OTTS DCGN
ott-sidecar.ssl.ose.keyStore.ott.truststore.certAlias	Alias сертификата OTTS Сервиса
ott-sidecar.ssl.ose.keyStore.ott.password	Пароль для сертификата OTTS DCGN
ott-sidecar.ssl.ose.keyStore.ott.truststore.password	Пароль для сертификата OTTS Сервиса
ott-sidecar.ssl.ose.keyStore.ott.truststore.rootAliases	Aliases для сертификата корневого УЦ

Более подробное описание конфигурации приводится в документации Platform V Backend на компонент One-Time-Token в «Руководстве по установке».

Сертификаты для SSL-взаимодействия с Kafka для Журналирования (LOGA)

В поставляемой конфигурации DCGN используется SSL-взаимодействие с Kafka для Журналирования (LOGA). Поэтому необходимо выпустить сертификаты Kafka для блока/контура, где производится установка DCGN.

Параметр	**Описание**
fluent-bit-sidecar.ssl.ose.keyStore.keyStoreFromFile	Путь до хранилища сертификатов kafka компонента Журналирования (LOGA)
fluent-bit-sidecar.ssl.ose.keyStore.password	Пароль от хранилища
fluent-bit-sidecar.ssl.ose.keyStore.rootCertAlias	Alias для сертификата корневого УЦ
fluent-bit-sidecar.ssl.ose.keyStore.certAlias	Alias для сертификата

Сертификаты для SSL-взаимодействия с БД

В поставляемой конфигурации DCGN используется SSL-взаимодействие с БД. В директории /home/postgres/ssl проверить наличие сгенерированных серверных сертификатов, при их отсутствии сгенерировать серверные сертификаты. Сгенерировать клиентский сертификат (используя корневые сертификаты) и при необходимости осуществить подписание, как серверных сертификатов, так и клиентских сертификатов в УЦ. Корневой и клиентский сертификаты необходимо поместить в защищенное хранилище сертификатов и передать его параметры в соответствующие конфигурации DCGN.

Параметр	**Описание**
dcgn.registry-db.ssl.keyStore.keyStoreFromFile	Путь до хранилища сертификатов
dcgn.registry-db.ssl.keyStore.passwordVariableName	Имя переменной с паролем от хранилища
dcgn.registry-db.ssl.keyStore.rootCertAliases	Alias для сертификата корневого УЦ
dcgn.registry-db.ssl.keyStore.certAlias	Alias для клиентского сертификата

Настройка параметров

Компонент IAM Proxy

Для отображения ссылки на сервис template-registry необходимо добавить junction-конфигурацию в сервис RDS:

```
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/applyJctRequestFilter=
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/junctionPoint=/template-registry
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/junctionName=DCGN. Template Registry
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/indexUrl=/template-registry/ui/
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/authorizeByRoleTemplate=
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/sslCommonN
```

```
ame=*
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/transparen
t=true
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/serverAddr
esses={nginx_ui_host:port};
rds-server@ZoneConfig[Core]/JunctionConfig[DCGN_templaterregistry]/https=true
```

Компонент Объединенный сервис авторизации (OCA)

Для создания в Объединенном сервисе авторизации (OCA) привилегий, наборов привилегий (ролей) и групп пользователей DCGN нужно импортировать ролевую модель DCGN в Объединенный сервис авторизации (OCA) (AUTZ). Импорт производится путем REST-запроса на импорт ролевой модели в Объединенный сервис авторизации (OCA) (AUTZ). Импорт может быть выполнен как на шаге установки дистрибутива (смотрите раздел «Выполнение установки»), так и после установки DCGN. При установке с помощью компонента Deploy Tools импорт происходит при выборе плейбука `IMPORT_SECURITY_PARAMS`.

Установка DCGN в контейнеризованную среду (Kubernetes или OpenShift (опционально))

Установка производится путем создания объектов в контейнеризированной среде на основе поставляемых в дистрибутиве yaml-конфигураций и установкой для них параметров, характерных для блока/контура.

Далее приведен пример установки с помощью компонента Deploy Tools.

Установка с помощью компонента Deploy Tools

Перед началом установки убедитесь, что инструменты Deploy Tools сконфигурированы должным образом. Для более тонкой настройки смотрите инструкции компонента Deploy Tools.

Создание репозитория конфигураций

Необходимо создать репозиторий для миграции и хранения конфигураций компонента DCGN. Репозиторий создается в той же проектной области, где находится common-репозиторий Deploy Tools.

Common-репозиторий — git-репозиторий глобальных настроек (переменных), которые используются всеми компонентами Platform V в рамках блока/контура.

Имя репозитория формируется по маске `\<PREFIX\>_\<CHANNEL\>_\<FPI_NAME\>_\<ENVIR\>\[_\<REPO_BLOCK_ID\>\]`, где:

- `PREFIX` — в большинстве случаев константа *ci00380023_efs*;
- `CHANNEL` — сегмент или инсталляция (например: ukofl, emp, corp, fis и т. п.);
- `FPI_NAME` — dcgn;
- `ENVIR` — окружение, тип стенда (например: dev, st, ift, nt, psi и т. п.);

- REPO_BLOCK_ID — блок/контур (например: b1, sb_sk и т. п.), в случае блочной структуры хранения конфигурации.

Значение параметров определяются в соответствии с конфигурацией Deploy Tools.

Далее в созданном репозитории нужно создать ветку \<BRANCH_NAME\> , где «BRANCH_NAME» — наименование ветки репозитория.

Конфигурация common-репозитория

Для установки компонента DCGN необходимо внести информацию в common-репозиторий стенда. Для этого нужно выполнить следующие действия:

1. В файл **subsystems.json** добавить блок настроек для возможности установки DCGN инструментами Deploy Tools согласно примеру ниже:

   Пример минимальной конфигурации для указанной версии компонента Deploy Tools

   Если версия отличается, смотрите инструкции компонента Deploy Tools.

```
 "DOCGEN": {                                    // наименование компонента Platform V
    "nexus_host": "<заполнить>",                // host нексус-репозитория
    "nexusPathToArtifact": "<заполнить>",       // путь до артефакта
    "nexus_repo": "<заполнить>",                // наименование нексус-репозитория
    "groupId": "<заполнить>",
    "artifactId": "<заполнить>",                // идентификатор maven артефакта
    "fpType": "bts",                            // тип компонента Platform V
    "fpi_name": "dcgn",                         // идентификатор компонента Platform V
    "serviceName": "dcgn",
    "strict": "true",
    "versionFilter": "D-01",                    // регулярное выражения для фильтрации версий артефактов
    "repoFullName": "<заполнить>",              // репозиторий конфигурации
    "fpi_name_ose": "<заполнить>",              // наименование созданного namespace
    "registryPath": "<заполнить>",             // базовый путь до каталога с образами DCGN
  "agents": {                                  // перечисление подключаемых агентов
    "UFS_MONITORING_CLIENT": {                  // наименование агента компонента Объединенный мониторинг Unimon
          "groupId": "<заполнить>",
          "artifactId": "<заполнить>", // идентификатор maven артефакта
          "version": "<заполнить>",             // версия клиента компонента O
```

```
бъединенный мониторинг Unimon
         "fpi_name": "ufs-monitoring",   // идентификатор компонента Pl
atform V
         "exclude": [                    // исключения, необходимо указ
ать пути ingress/egress конфигураций согласно расположению в дистрибути
ве агента компонента Объединенный мониторинг Unimon
             "package/conf/openshift/istio/config/ingress/*",
             "package/conf/k8s/base/istio/config/egress/egressgateway-m
onitoring",
             "package/conf/k8s/base/istio/config/egress/egressgateway-m
onitoring-client",
             "package/conf/k8s/base/istio/config/egress/egressgateway-k
afka-monitoring",
             "package/conf/openshift/istio/deployments/ingress/*",
             "package/conf/openshift/istio/deployments/egress/*
          ]
          }
      }
 }
```

В данном руководстве рассматривается вариант установки агента Объединенный мониторинг Unimon, сконфигурированного как подключаемый агент к основной ФП, это описывается в блоке `UFS_MONITORING_CLIENT`. Потребитель оставляет за собой право выбрать любой способ, предложенный компонентом Deploy tools. Так как используется pull-модель, дистрибутив DCGN не содержит в себе средства тиражирования метрик, а дистрибутив Объединенный мониторинг Unimon поставляется отдельно. В параметре `version` необходимо указать версию Объединенный мониторинг Unimon. Протестированные версии и как их сконфигурировать описано в разделе «Изменение конфигурации компонента Объединенный мониторинг Unimon».

2. В common-репозиторий необходимо добавить конфигурации подключения к БД DCGN:

   – В файл **ansible/common.conf.yml** для выполнения liquibase-скриптов:

     ```
     DCGN_POSTGRES_DB_URL: <URL подключения к БД Postgres>
     ```

   – В файл **ansible/_passwords.conf**:

     ```
     jdbc.DOCGEN.user=<Имя созданной учетной записи БД>
     jdbc.DOCGEN.password=<Пароль созданного учетной записи БД>
     ```

   – Для необходимости задания разных учетных записей для DCGN и для выполнения скриптов мигарации БД, в целях разделения полномочий на DDL и DML операции, необходимо добавить в файл **ansible/_passwords.conf**:

     ```
     liquibase.jdbc.DOCGEN.user=<Имя учетной записи БД для выполнения
     сприптов liquibase>
     ```

```
liquibase.jdbc.DOCGEN.password=<Пароль учетной записи БД для выпо
лнения сприптов liquibase>
```

– В файл **installer/system/efs/config/parameters/_global.jdbc.conf** для подключения к БД из приложений:

```
jdbc.dcgn_postgres.ssl.mode=<Режим работы SSL>
jdbc.dcgn_postgres.url=<URL подключения к БД Postgres>
```

Если не задан DCGN_POSTGRES_DB_URL, то параметр `jdbc.dcgn_postgres.url` используется и для выполнения liquibase-скриптов.

Подробнее о возможных режимах работы SSL описано в документе «Руководство по системному администрированию», в разделе «Настройка и конфигурирование DCGN». Пример формирования URL для подключения к БД с использованием параметров SSL:

```
jdbc:postgresql://<db-host>:<db-port>/<db-name>?sslmode=${jdbc.dc
gn_postgres.ssl.mode}&sslfactory=org.postgresql.ssl.DefaultJavaSS
LFactory
```

Параметр `sslfactory=org.postgresql.ssl.DefaultJavaSSLFactory` является обязательным для испольования защищенного хранилища сертификатов Java-приложения.

3. В common-репозиторий в файл **multiclusters.json** добавить путь до реестра образов контейнеров:

```
"registry": "<URL registry, где хранятся образы контейнеров>",
"registry_path": "${fpConfig.registryPath}"
```

Используемые глобальные параметры внешних продуктов

Для интеграции со внешними и инфраструктурными компонентами в файлах конфигурации DCGN используются ссылки на глобальные параметры из common-репозитория. В глобальных параметрах содержится информация об адресах подключения к БД, базовые URL для http(-s) взаимодействий с компонентами, адреса подключения к серверам kafka и др.

Внимание!

Отсутствие глобальных параметров без изменения поставляемой конфигурации DCGN может привести к ошибкам во время установки.

№	Параметр	Описание
-	**Настройки, связанные с K8s или OSE (опционально)**	
1	global.multiClusters.openshiftNewRoute	Используется для формирование route
2	global.ose.platform.egress.http.port	Порт для egress gateway, через который будут происходить

№	Параметр	Описание
		обращения сервисов компонента Platform V во внешние компоненты
3	global.revisionHistoryLimit	Максимальное количество ревизий deployment для хранения
4	global.ufs.strategy.maxSurge	Параметр стратегии RollingUpdate. Максимальное количество подов, которые могут быть запланированы выше исходного количества
5	global.ufs.strategy.maxUnavailable	Параметр стратегии RollingUpdate. Максимальное количество подов, которые могут быть недоступны в процессе обновления
6	global.multiClusters.openshiftControlPlaneProject	Наименование namespace с развернутым control plane
7	global.multiClusters.openshiftControlPlaneIstiodService	Адрес сервиса istiod Control Plane
8	global.multiClusters.openshiftControlPlaneIstiodPort	Порт сервиса istiod Control Plane
9	global.multiClusters.openshiftControlPlaneJaegerService	Адрес сервиса jaeger Control Plane
10	global.multiClusters.openshiftControlPlaneJaegerPort	Порт сервиса jaeger Control Plane
-	**Настройки подключения к БД**	
11	DB_SCHEMA_SUFFIX	Суффикс БД, используется при миграции скриптов liquibase
12	jdbc.dcgn_postgres.ssl.mode	Режим работы SSL-подключения к БД
13	jdbc.dcgn_postgres.url	URL подключения к БД
14	global.jdbc.spring.datasource.hikari.maximum-pool-size	Максимальное количество соединений в pool
15	global.jdbc.spring.datasource.hikari.minimum-idle	Минимальное количество соединений в pool
16	global.jdbc.spring.datasource.hikari.connection-timeout	Время, в течение которого пользователь должен получить соединение, при достижении тайм-аута выпадает SQLException
17	global.jdbc.spring.datasource.hikari.validation-timeout	Время, в течение которого должен выполниться запрос на валидацию соединения
18	global.jdbc.spring.datasource.hikari.max-lifetime	Время, по истечении которого соединение будет удалено из-

№	Параметр	Описание
		за старости, измеряется в миллисекундах
19	global.jdbc.spring.datasource.hikari.idle-timeout	Время, в течение которого соединение может не обслуживать запросы прежде, чем может быть удален по idle timeout, измеряется в миллисекундах
20	global.jdbc.spring.datasource.hikari.connection-test-query	Параметр отвечает за то, каким SQL-запросом будет выполнена проверка. Если значение не задано, то проверка будет осуществлена с помощью драйвера
21	global.jdbc.spring.datasource.hikari.leak-detection-threshold	Определяет время, в течение которого соединение может быть вне pool, прежде, чем в лог будут отправляться сообщения о возможной утечке соединений
22	global.jdbc.spring.datasource.hikari.initializationFailTimeout	Определяет поведение хикари при первом взятии соединения. Положительное значение — количество миллисекунд, в течение которых должно быть осуществлено взятие первого соединения
23	global.jdbc.postgres.spring.datasource.hikari.data-source-properties.socketTimeout	Время, в течение которого SQL-запрос пользователя должен выполниться, измеряется в секундах
24	global.jdbc.postgresql.spring.datasource.hikari.connection-init-sql	SQL-запрос, который будет выполнен при создании соединения с БД для проверки работоспособности соединения
25	global.jdbc.postgresql.spring.datasource.driver-class-name	Имя класса с драйвером
26	global.jdbc.hikari.housekeeping.periodMs	Частота, с которой хикари ищет idle-соединения в pool, измеряется в миллисекундах
27	global.jdbc.hikari.aliveBypassWindowMs	Перед передачей соединения пользователю хикари проверяет его работоспособность, если после проверки соединение вернется в pool, то в течение этого времени соединение будет считаться проверенным и не будет проверено перед

№	Параметр	Описание
		выдачей пользователю, измеряется в миллисекундах
-	**Настройки для интеграции с компонентом Аудит (AUDT)**	
28	global.platform.audit2.protocol	Протокол для подключения к Аудиту
29	global.platform.audit2.host	Хост прокси приложения Аудит
30	global.platform.audit2.port	Порт для подключения к Аудиту
-	**Настройки для интеграции с компонентом Журналирование (LOGA)**	
31	global.platform.logger.kafka.bootstrap.servers	Адреса хостов Kafka
32	global.platform.logger.kafka.security.protocol	Используемый протокол PLAINTEXT/SSL
33	global.platform.logger.kafka.topic	Наименование топика Kafka
34	global.platform.ose.kafka.ports	Список портов Kafka
-	**Настройки SSL для Журналирования (LOGA)** В настройку пароль от хранилища передается как наименование параметра в *«_passwords.conf»*.	
35	ssl.ose.keyStore.mq.keyStoreFromFile	Путь до хранилища сертификатов Kafka компонента Журналирования (LOGA)
36	ssl.ose.keyStore.mq.password	Пароль от хранилища
37	ssl.ose.keyStore.mq.RootAlias| ssl.ose.istio.keyStore.RootCertAlias	Alias для сертификата корневого УЦ
38	ssl.ose.keyStore.mq.CertAlias	Alias для сертификата
-	**Настройки SSL для istio egress/ingress** В настройку пароль от хранилища передается как наименование параметра в *«_passwords.conf»*.	
39	ssl.ose.istio.keyStore.egress.KeyStoreFromFile	Путь до хранилища сертификатов egress
40	ssl.ose.istio.keyStore.egress.password	Пароль от хранилища
41	ssl.ose.istio.keyStore.egress.CertAlias	Alias сертификата для egress
42	ssl.ose.istio.keyStore.RootCertAlias	Alias для сертификата корневого УЦ
43	ssl.ose.istio.keyStore.ingress.KeyStoreFromFile	Путь до хранилища сертификатов ingress
44	ssl.ose.istio.keyStore.ingress.password	Пароль от хранилища
45	ssl.ose.istio.keyStore.ingress.CertAlias	Alias сертификата для ingress
-	**Авторизация внешних/внутренних вызовов для istio egress/ingress**	
46	global.ufs.istio.ingress.envoy_filter.regex_match.client.cn	Регулярное выражение для проверки CN (common name) входящих SSL-сертификатов

№	Параметр	Описание
-	**Настройки SSL для One-Time Password (OTP) / OTT —** опционально. В настройку пароль от хранилища передается как наименование параметра в *«_passwords.conf»*.	
47	ssl.ose.keyStore.truststore.keyStoreFromFile	Путь до хранилища с сертификатом OTTS Сервиса
48	ssl.ose.keyStore.truststore.password	Пароль для сертификата OTTS Сервиса
49	ssl.ose.keyStore.keystore.keyStoreFromFile	Путь до хранилища с сертификатом OTTS DCGN
50	ssl.ose.keyStore.password	Пароль для сертификата OTTS DCGN
-	**Настройки для интеграции с компонентом One-Time Password (OTP) / OTT**	
51	global.ott.ose_deploy	Признак, указывающий на необходимость установки One-Time Password (OTP) / OTT
52	global.ott.service.hosts	Адреса хостов сервера One-Time Password (OTP) / OTT
53	global.ott.service.url	URL сервера One-Time Password (OTP) / OTT
-	**Настройки для интеграции с компонентом Объединенный сервис авторизации (AUTZ)**	
54	global.platform.ufs-security.host	Хост подключения к Объединенному сервису авторизации (AUTZ)
55	global.platform.ufs-security.port	Порт подключения к Объединенному сервису авторизации (AUTZ)
56	global.platform.ufs-security.protocol	Протокол подключения к Объединенному сервису авторизации (AUTZ)
57	global.platform.ufs-security.url	URL подключения к Объединенному сервису авторизации (AUTZ)
-	**Настройки для интеграции с компонентом IAM proxy (AUTH)**	
58	global.platform.iam.jwks.core.protocol	Протокол подключения к IAM proxy
59	global.platform.iam.jwks.core.host	Хост подключения к IAM proxy
60	global.platform.iam.jwks.core.port	Порт подключения к IAM proxy
61	global.platform.iam.auth.publickey.locations	Шаблоны для сопоставления token issuer с public key location

Конфигурация DCGN

Дистрибутив поставляется с рекомендуемыми значениями параметров конфигурации. Если какая-либо интеграция не используется, то соответствующие параметры могут отсутствовать. Набор поставляемых конфигураций может отличаться в зависимости от поставки.

Общий список конфигураций:

- **dcgn.all.conf** — общие настройки сервисов DCGN. В основном содержит ссылки на глобальные параметры.
- **dcgn.docgen-service.conf** — настройки для Docgen service.
- **dcgn.template-provider.conf** — настройки для Template provider.
- **dcgn.template-registry.conf** — настройки для Template registry.
- **dcgn.fluent-bit-sidecar.all.conf** — настройки, которые используются для интеграции с сервисом Журналирования (LOGA).
- **dcgn.istio.all.conf** — общие настройки для istio. Включают в себя настройки для интеграции с сервисом Аудит (AUDT).
- **dcgn.ott-sidecar.all.conf** — настройки, которые используются для интеграции с сервисом OTTS.

Описание всех настроек из файлов конфигураций приведено в документе «Руководство по системному администрированию», в разделе «Настройка и конфигурирование DCGN».

При установке агента мониторинга добавляются файлы конфигурации компонента Объединенный мониторинг Unimon:

- *-monitoring.all.conf — общие настройки сервиса UNIMON.
- *-monitoring.unimon-agent.conf — настройки unimon-agent
- *-monitoring.unimon-sender.conf — настройки unimon-sender.

Как сконфигурировать мониторинг, смотрите в документе «Руководство по установке» клиентской части Объединенный мониторинг Unimon.

Миграция конфигурации из дистрибутива в репозиторий

Для миграции конфигурации из дистрибутива в репозиторий нужно выполнить следующие действия:

1. Перейти к job Jenkins, предназначенной для развертывания (установки) приложений.

2. В меню слева нажать на опцию «Build with parameters».

3. Установить параметры сборки:

   – **SUBSYSTEM**: *DOCGEN;*

- **COMPONENTS**: *Основная ФП DOCGEN;* - так же выбрать *UFS_MONITORING_CLIENT* если установка происходит совместно с агентом Объединенный мониторинг Unimon
- **DISTRIB_VERSION**: *<выбрать версию дистрибутива>;*
- **OSE_CLUSTERS**: *<выбрать кластер K8s/OpenShift (опционально)>;*
- **Репозиторий/ветка с настройками ФП**: *<основная ветка конфигурации в соответствии с настройками Deploy Tools>;*
- **PARAMS (набор playbook)**: *<выбрать MIGRATION_FP_CONF>.*

4. Запустить сборку, нажав кнопку «Build».

В результате миграции в репозиторий конфигурации в выбранную ветку попадет часть конфигурационных файлов дистрибутива, которые можно отредактировать.

Ниже приведен пример структуры репозитория после миграции:

```
├── config
│   └── parameters
│       ├── dcgn.all.conf
│       ├── dcgn.docgen-service.conf
│       ├── dcgn.istio.all.conf
│       ├── dcgn.template-provider.conf
│       └── dcgn.template-registry.conf
├── inventory
│   ├── inventory
│   └── nginx_upstreams_inventory
└── version.conf
```

Инициализация БД

Конфигурация Deploy Tools

Для запуска скриптов инициализации БД необходимо настроить инструменты **Deploy Tools**.

1. Настроить правила распаковки родительского дистрибутива и его зависимостей для профиля **db-init**, файл **environment.json**.

```
{
    ...
    "downloadProfiles": {
      ...
      "db-init": {
        "parentUnpackIncludes": [
          "/package/conf/**"
        ],
        "parentUnpackExcludes": [
          "/package/conf/distrib.yml"
        ],
        "dependenciesUnpackIncludes": [
          "/package/**"
```

```
        ],
        "dependenciesUnpackExcludes": []
      }
      ...
    }
    ...
}
```

2. Добавить плейбук **DB_INIT** в список возможных сценариев развертывания, файл **environment.json**.

```
    ...
    "playbooks_fpi": {
      "DB_INIT": {
        "id": 1 // идентификатор плейбука, порядковый номер относительно друг
их плейбуков в репозитории
      },
      ...
    }
    ...
```

3. Добавить параметры в файл **ansible/common.conf.yml**, значения указаны в качестве примера.

```
DB_SCHEMA_SUFFIX: "_DEV" # суффикс БД - уникальный идентификатор
...
DCGN_POSTGRES_DB_HOST: 127.0.0.1 # IP-адрес БД
DCGN_POSTGRES_DB_PORT: 5432      # порт
DCGN_POSTGRES_DB_NAME: postgres  # имя БД
DCGN_POSTGRES_DB_URL: jdbc:postgresql://{{ DCGN_POSTGRES_DB_HOST }}:{{ DCGN_P
OSTGRES_DB_PORT }}/{{ DCGN_POSTGRES_DB_NAME }}
# базовая директория для хранения табличных пространств (опционально)
DCGN_POSTGRES_DB_TS_LOCATION: /pgdata/ts
# табличное пространство для основных данных (наименование и расположение на
файловой системе)
DCGN_POSTGRES_DB_TS_DATA: dcgn_ts_data
DCGN_POSTGRES_DB_TS_DATA_LOCATION: "{{ DCGN_POSTGRES_DB_TS_LOCATION }}/{{ DCG
N_POSTGRES_DB_TS_DATA }}" # /pgdata/ts/dcgn_ts_data
# табличное пространство для индексов (наименование и расположение на файлово
й системе)
DCGN_POSTGRES_DB_TS_IDX: dcgn_ts_idx
DCGN_POSTGRES_DB_TS_IDX_LOCATION: "{{ DCGN_POSTGRES_DB_TS_LOCATION }}/{{ DCGN
_POSTGRES_DB_TS_IDX }}"   # /pgdata/ts/dcgn_ts_idx
# табличное пространство для больших объектов (наименование и расположение на
файловой системе)
DCGN_POSTGRES_DB_TS_LOB: dcgn_ts_lob
DCGN_POSTGRES_DB_TS_LOB_LOCATION: "{{ DCGN_POSTGRES_DB_TS_LOCATION }}/{{ DCGN
_POSTGRES_DB_TS_LOB }}"   # /pgdata/ts/dcgn_ts_lob
```

4. Добавить учетную запись БД в **_passwords.conf** под которой будут выполняться скрипты инициализации БД.

```
# глобальная учетная запись, для инициализации пользователей и схем БД компон
ентов платформы
POSTGRES_DB_INIT_USERNAME=<имя пользователя>
POSTGRES_DB_INIT_PASSWORD=<пароль>
# или специфичная для компонента DCGN учетная запись
DCGN_POSTGRES_DB_INIT_USERNAME=<имя пользователя>
DCGN_POSTGRES_DB_INIT_PASSWORD=<пароль>

# Пользователь БД будет создан с этими данными
jdbc.DOCGEN.user=<имя пользователя>
jdbc.DOCGEN.password=<пароль>
```

5. Создать директории табличных пространств на файловой системе БД.

```shell script # директория, указанная в параметре DCGN_POSTGRES_DB_TS_DATA_LOC
ATION mkdir -p /pgdata/data/ts/dcgn_ts_data chown postgres:postgres /pgdata/data/ts/
dcgn_ts_data

директория, указанная в параметре DCGN_POSTGRES_DB_TS_IDX_LOCATION

mkdir -p /pgdata/data/ts/dcgn_ts_idx chown postgres:postgres /pgdata/data/ts/dcgn_ts_i
dx

 # директория, указанная в параметре DCGN_POSTGRES_DB_TS_LOB_LOCATION

mkdir -p /pgdata/data/ts/dcgn_ts_lob chown postgres:postgres /pgdata/data/ts/dcgn_ts_l
ob
```

```
###### Запуск скриптов инициализации БД

1. Перейти к job Jenkins, предназначенной для развертывания (установки) прило
жений.

2. В меню слева нажать на опцию «Build with parameters».

3. Установить параметры сборки:

   * **SUBSYSTEM**: *DOCGEN;*
   * **COMPONENTS**: *Основная ФП DOCGEN;*
   * **DISTRIB_VERSION**: *\<выбрать версию дистрибутива\>;*
   * **OSE_CLUSTERS**: *\<выбрать кластер K8s/OpenShift (опционально)\>;*
   * **Репозиторий/ветка с настройками ФП**: *\<основная ветка конфигурации в
соответствии с настройками Deploy Tools\>;*
   * **PARAMS (набор playbook)**: \<*выбрать DB_INIT\>.*

4. Запустить сборку, нажав на кнопку «Build».

В результате выполнения будут созданы следующие объекты БД:

* пользователь (с именем - _jdbc.DOCGEN.user_ и паролем - _jdbc.DOCGEN.passwo
```

```
rd_) и одноименная схема БД;
* табличные пространства, указанные в параметрах: _DCGN_POSTGRES_DB_TS_DATA,
DCGN_POSTGRES_DB_TS_IDX, DCGN_POSTGRES_DB_TS_LOB_.

##### Настройка мультитенантности

###### Коммунальная инсталляция

Для поддержки мультитенантности в соответствующих конфигурациях (смотрите раз
дел «Конфигурация DCGN») требуется задать следующие параметры:

```properties
dcgn.template-provider.unimonId
dcgn.template-registry.unimonId
dcgn.docgen-service.unimonId
dcgn.template-provider.rn
dcgn.template-registry.rn
dcgn.docgen-service.rn
```

Platform V Backend

Для поддержки мультитенантности в Platform V Backend требуется выполнить следующие действия:

1. Для обратной совместимости с бэкенд в конфигурации мониторинга нужно включить автономный режим (см. в документе «Руководство по установке» клиентской части Объединенный мониторинг Unimon).

2. В соответствующих конфигурациях (смотрите раздел «Конфигурация DCGN») не задавать параметры:

```
dcgn.template-provider.unimonId
dcgn.template-registry.unimonId
dcgn.docgen-service.unimonId
```

Примечание.

Для инсталляции с настроенным взаимодействием через Topic (Kafka) и без unimon server параметр «unimonId» задается произвольным значением (смотрите раздел «Конфигурация DCGN» ufs-monitoring.unimon-sender.conf — настройки unimon-sender).

Изменение конфигурации компонента Объединенный мониторинг Unimon

Необходимо сконфигурировать клиента мониторинга согласно инструкции компонента Объединенный мониторинг Unimon с учетом требований к сервису DCGN, выставить в файлах конфигурации клиента мониторинга значения настроек:

- unimon-sender.sidecar.istio.rewriteAppHTTPProbers=true
```

- unimon.server.enable=false, если на инсталяции не предполагается взаимодейтсиве с сервером Unimon.
- metric.label.rn - очистить значение, так как информация о RN передается на уровне метрик сервиса;

Проверка конфигурации

Для проверки конфигурации нужно выполнить следующие действия:

1. Перейти к job Jenkins, предназначенной для развертывания (установки) приложений.
2. В меню слева нажать на опцию «Build with parameters».
3. Установить параметры сборки:
   - **SUBSYSTEM**: *DOCGEN;*
   - **COMPONENTS**: *Основная ФП DOCGEN;* - так же выбрать *UFS_MONITORING_CLIENT* если установка происходит совместно с агентом Объединенный мониторинг Unimon
   - **DISTRIB_VERSION**: *<выбрать версию дистрибутива>;*
   - **OSE_CLUSTERS**: *<выбрать кластер K8s/OpenShift (опционально)>;*
   - **Репозиторий/ветка с настройками ФП**: *<основная ветка конфигурации в соответствии с настройками Deploy Tools>;*
   - **PARAMS (набор playbook)**: <*выбрать FP_CONF_CHECK>.*
4. Запустить сборку, нажав кнопку «Build».

После выполнения сборки проанализировать лог на предмет наличия некорректной конфигурации.

Ниже приведен пример отсутствия глобальных переменных:

```
-------------------------------------------------------------------------------
-------------
Проверка соответствия параметров в конфигурационных файлах <fp_name>.conf и _
global.conf
-------------------------------------------------------------------------------
-------------
Не найдено значений для global.platform.pprb.baseurl.audit в файле dcgn.templ
ate-registry.conf
Не найдено значений для global.platform.ingress.route.http в файле dcgn.istio
.all.conf
Не найдено значений для global.platform.ingress.route.https в файле dcgn.isti
o.all.conf
```

Выполнение установки

Для установки ПО необходимо выполнить следующие действия при запуске Job-развертывания:

1. Установить параметры сборки:
    - **SUBSYSTEM**: *DOCGEN;*
    - **COMPONENTS**: *Основная ФП DOCGEN;* - так же выбрать *UFS_MONITORING_CLIENT* если установка происходит совместно с агентом Объединенный мониторинг Unimon
    - **DISTRIB_VERSION**: *<выбрать версию дистрибутива>;*
    - **OSE_CLUSTERS**: *<выбрать кластер K8s/OpenShift (опционально)>;*
    - **Репозиторий/ветка с настройками ФП**: *<основная ветка конфигурации в соответствии с настройками Deploy Tools>;*
    - **PARAMS (набор playbook)**: *<выбрать SHIFT_OK>.*
2. Запустить сборку, нажав кнопку «Build».

После завершения выполнения сборки необходимо проверить, что лог не содержит ошибок.

В случае если плейбук `SHIFT-OK` отсутствует, то нужно выбрать плейбуки, приведенные в таблице ниже.

Название	**Описание**
DB_UPDATE	Запуска liquibase-скриптов инициализации и миграции БД
IMPORT_SECURITY_PARAMS	Импорт ролевой модели в Объединенный сервис авторизации (AUTZ)
OPENSHIFT_INGRESS_EGRESS_DEPLOY	Установка компонентов istio ingress/egress
OPENSHIFT_DEPLOY	Установка в K8s или OSE (опционально)

Набор возможных плейбуков зависит от конфигурации инструментов Deploy Tools.

Рекомендации после установки компонентом Deploy Tools

Для облегчения интеграции потребителей с DCGN, которые используют Deploy Tools, рекомендуется:

- добавить глобальные параметры для подключения к сервисам DCGN,
- добавить плейбук импорта архива шаблона в DCGN.

Глобальные параметры для подключения к сервисам

В файл **installer/system/efs/config/parameters/_global.resources.conf** добавить глобальные параметры для подключения к сервисам:

- сервис генерации документов (docgen-service),
- провайдер шаблонов (template-provider).

№	**Параметр**	**Описание**	**Значение**
1	global.platform.baseurl.docgen	Базовый URL для подключения сервису генерации документов	protocol://host:port/

№	Параметр	Описание	Значение
2	global.platform.baseurl.template-provider	Базовый URL для подключения провайдеру шаблонов	protocol://host:port/
3	resourceName	Переменная для задания resourceName в global.platform.url.template-registry. Задано как условие	{% if fpConfig.resourceName is defined %}{{fpConfig.resourceName}}{% else %}*ZAPOLNI_RUKAMI_PARAMETER_resourceName*{% endif %}
4	global.platform.url.template-registry	URL API для импорта в Реестр шаблонов	protocol://host:port/template-registry/api/v1/rn/${resourceName}/templates

Примечание. Наименования параметров носит рекомендательный характер.

Плейбук импорта шаблонов

Для возможности импортирования архивов шаблонов рекомендуется добавить плейбук импорта шаблонов — `IMPORT_DCGN_PARAMS`. Deploy Tools поставляется с common-конфигурациями, в которых по умолчанию настроен плейбук.

Если необходимо настроить плейбук вручную (например, если в common-репозитории переопределен блок playbooks_import для среды):

- В common-репозитории в *environment.json* добавить:

```
"IMPORT_DCGN_PARAMS": {
   "id": 14, // идентификатор плейбука, порядковый номер относительно других плейбуков в репозитории
   "description": "Импорт архивов шаблонов DCGN", // Необязательный параметр для отображения при выборе плейбука и в отчете в результатах
   "dataDir": "dcgn_templates" // Обязательный параметр, указывающий на директорию в conf/data/ в дистрибутиве с архивами шаблонов для сервиса
   "authType": "ott" // тип авторизации, для OTTS необходимо указать "ott", по-умолчанию - "basic" (Basic Authorization)
 },
```

- Занести в _global.resources.conf адрес нового загрузчика в формате:

global.import.service.dcgn.url=${global.platform.url.template-registry}

В pipeline для плейбука должны быть настроенны OTT-авторизация и SSL, подробнее по настройке в документации компонента Deploy Tools.

Обновление

Обновление на версию DCGN 1.2.0 происходит по следующим шагам:

1. При сборке с помощью Build Tools (Solution Merger Job) наполнить продукт зависимостями согласно рекомендациям в пункте «Рекомендации по

наполнению продукта зависимостями при сборке с помощью Build Tools (Solution Merger Job)».

2. Сконфигурировать параметры, которые были добавлены в версии 1.2.0. Новые параметры описаны в «Руководстве по системному администрированию» в разделе «Настройка и конфигурирование DCGN», для новых параметров в столбце «Версия ПО» указывается версия 1.2.0.

3. Сконфигурировать SSL для БД согласно разделу «Сертификаты для SSL-взаимодействия с БД».

4. В файл **subsystems.json** добавить базовый путь до каталога с образами DCGN согласно примеру в разделе «Конфигурация common-репозитория» в пункте 1.

5. Выполнить настройки, которые приведены в разделе «Конфигурация common-репозитория» в пункте 3.

6. Установить версию 1.2.0 (установка через Deploy Tools описана в разделе «Выполнение установки»).

Удаление

Для удаления DCGN необходимо выполнить следующие действия:

1. Удалить созданный namespace в кластере K8s или OSE (опционально);

2. Если установка производилась Deploy Tools, удалить:

- репозиторий конфигураций;
- созданные глобальные переменные;
- созданный плейбук импорта.

3. Удалить пользователя и схему БД.

Проверка работоспособности

Проверка корректности работы DCGN включает проверку по следующим позициям:

1. В контейнеризованной среде (Kubernetes или OpenShift (опционально)) созданы объекты, соответствующие конфигурации в дистрибутиве.

2. Поды сервисов запущены (находятся в статутсе `Running`) и контейнеры работают без ошибок.

3. На вызов Endpoint с healthcheck приходит ответ: `{"success":true,"body":"ON"}`. Формирование ссылки описано в документе «Руководство по системному администрированию», раздел «Правила формирования ссылки до healthcheck сервисов». Пример curl запроса, с целевым взаимодействием с сервисом через mTLS и передачей в запросе клиентского и корневого сертификатов:

```
curl --cert tls.crt --key tls.key --cacert root.crt протокол://хост:порт/docg
en-service/healthcheck
```

4. Метрики состояния сервисов возвращают значение доступности сервиса (метрики указаны в документе «Руководство по системному администрированию», в разделе «События мониторинга»).

5. Пользователю UI DCGN после успешной аутентификации в IAM Proxy или СУДИР и авторизации открывается приложение «Реестр шаблонов документов» на вкладке меню «Шаблоны документов».

6. Доступ к функциональности UI Реестр шаблонов для пользователя ограничен в соответствии с ролевой моделью (согласно назначенным им ролям. Подробнее про разграничение ролей описано в документе «Руководство по системному администрированию», в разделе «Принципы разграничения доступа к функциям UI»).

Откат

В общем случае для отката к начальным настройкам или любой из предыдущих версий необходимо выполнить установку требуемой версии согласно поставляемой инструкции по установке (документ «Руководство по установке»). Для отката к предыдущей версии обязательного удаления текущей не требуется.

Часто встречающиеся проблемы и пути их устранения

№	Ошибка	Способ устранения
1	При ошибке выполнения liquibase скриптов	Посмотреть лог сборки в Jenkins на предмет ошибки; Проверить подключение к БД; Проверить привилегии пользователя; Произвести соответствующие логам правки в common-репозитории Deploy Tools
2	При ошибке импорта в сервисы	Проверить корректность настройки URL для импорта в common-репозитории Deploy Tools. Проверить работоспособность компонента Platform V, при импорте в который произошла ошибка
3	Ошибка установки на Nginx	Посмотреть лог сборки в Jenkins; Произвести соответствующие правки в common-репозитории Deploy Tools или в репозитории с конфигурацией DCGN
4	Ошибки параметризации конфигурации	Посмотреть лог сборки в Jenkins; Произвести соответствующие правки в common-репозитории Deploy Tools или в репозитории с конфигурацией DCGN
5	Ошибка установки в K8s или OSE (опционально)	Посмотреть лог сборки в Jenkins на предмет ошибки; Произвести соответствующие правки в common репозитории Deploy Tools или репозитории с конфигурацией DCGN

Чек-лист валидации установки

В целях проверки корректности установки необходимо пройти по всем пунктам чек-листа.

№	Выполненные действия	Признак обязательности	Примечание
1	Создана схема и пользователь БД	Да	Смотрите раздел «Создание пользователя и схемы БД»
2	Создан репозиторий для хранения конфигурации	Нет	Для установки компонентом Deploy Tools смотрите раздел «Установка с помощью компонента Deploy Tools»
3	Создан namespace в кластере K8s или OSE (опционально)	Да	Смотрите раздел «Создание namespace в K8s или OSE (опционально)»
4	Произведена настройка common репозитория	Нет	Для установки компонентом Deploy Tools смотрите раздел «Установка с помощью компонента Deploy Tools»
5	Проведена миграция конфигурации	Нет	Для установки компонентом Deploy Tools смотрите раздел «Миграция конфигурации из дистрибутива в репозитори»
6	Внесены изменения в стандартную конфигурацию	Нет	Для установки компонентом Deploy Tools смотрите раздел «Установка с помощью компонента Deploy Tools»
7	Выполнена миграция liquibase-скриптов	Да	Для установки компонентом Deploy Tools - при установке был выбран плейбук `DB_UPDATE` смотрите раздел «Установка с помощью компонента Deploy Tools». В БД созданы таблицы в соответствии со скриптами в дистрибутиве
8	Выполнена установка компонентов istio (ingress gateway / egress gateway) в кластер K8s или OSE (опционально)	Да	Для установки компонентом Deploy Tools - при установке был выбран сплейбук `OPENSHIFT_INGRESS_EGRESS_DEPLOY`, смотрите раздел «Установка с помощью компонента Deploy Tools». В K8s или OSE (опционально) созданы соответствующие объекты
9	Выполнена установка сервисов DCGN: docgen-service, template-provider, template-registry	Да	Для установки компонентом Deploy Tools - при установке был выбран плейбук `OPENSHIFT_DEPLOY` смотрите раздел «Установка с помощью компонента Deploy Tools». В K8s или OSE (опционально) созданы соответствующие объекты
10	Выполнена проверка работоспособности	Да	Смотрите раздел «Проверка работоспособности»
11	Добавлены глобальные параметры для подключения к сервисам DCGN	Нет	Для установки компонентом Deploy Tools смотрите раздел «Установка с помощью компонента Deploy Tools»
12	Добавлен плейбук импорта в DCGN	Нет	Для установки компонентом Deploy Tools смотрите раздел «Установка с помощью компонента Deploy Tools»
13	Успешно выполнена работа плейбука для создания в компоненте авторизации привилегий, наборов	Да	Для установки компонентом Deploy Tools — при установке был выбран плейбук, смотрите раздел «Установка с помощью компонента Deploy Tools»

№	Выполненные действия	Признак обязательности	Примечание
	привилегий (ролей) и групп пользователей DCGN — IMPORT_SECURITY_PARAMS		