

**Инструкция по установке Arenadata Integration Module (AIM) и
развертыванию компонентов интеграции продуктов Arenadata с
технологическими сервисами платформы ГосТех**

Содержание:

1	Общая информация.....	4
1.1	Общие требования для использования	4
1.1.1	Требования к хосту запуска оркестратора и на хостах установки компонентов интеграции	5
1.1.2	Требования в отношении продуктов Arenadata.....	5
1.1.3	Требования к технологическим сервиса ЖАМ.....	6
1.2	Выполнение установки.....	6
1.2.1	Установка дашборда Журналирования для ADQM	7
2	Оркестратор.....	8
2.1	Описание реализации и запуск Оркестратора.....	8
2.2	До первого запуска - настройка	9
2.2.1	Реквизиты доступа к ADCM API	9
2.2.2	Реквизиты подключения к Журналированию	10
2.2.3	Реквизиты подключения к Аудиту	11
2.2.4	Реквизиты подключения к Мониторингу.....	12
2.2.5	Реквизиты доступа к кластерам-источникам событий аудита.....	12
2.2.6	Реквизиты подключения к серверу установки компонентов интеграции	13
2.2.7	Информация о ПАК.....	14
2.3	После первого запуска	14
2.3.1	При развертывании ADQM	15
2.3.2	При развертывании ADH или ADS	15
2.3.3	Если в компонентах отсутствуют ожидаемые продукты	16
2.4	После второго запуска	16
2.5	Третий и последующие запуски.....	17
2.6	Описание параметров запуска оркестратора.....	19
2.6.1	Общие.....	19
2.6.2	ADCM реквизиты подключения и настройки	20
2.6.3	Журналирование - реквизиты приемника и настройки	22
2.6.4	Аудит: реквизиты приемника и настройки	24
2.6.5	Аудит: настройки для ADB	25
2.6.6	Аудит: настройки для ADQM.....	26
2.6.7	Аудит: настройки для ADH и ADS (ADPS)	27
2.6.8	Мониторинг: реквизиты приемника и общие параметры	28
2.6.9	Мониторинг: метрики heartbeat	30
2.6.10	Мониторинг: ADB метрики кол-во запросов.....	30
2.6.11	Системы-источники	30
2.6.12	Ansible параметры для генерации и выполнения.....	32
3	Ansible плейбуки установки компонентов интеграции	35
3.1	Плейбук Журналирование.....	35
3.1.1	Цель	35
3.1.2	Дистрибутив.....	35
3.1.3	Порядок развертывания.....	35
3.2	Плейбук Аудит для ADB	36
3.2.1	Цель	36
3.2.2	Дистрибутив.....	36
3.2.3	Порядок развертывания.....	37
3.2.4	Публикация метамодели Аудита для ADB	37

3.3	Плейбук Аудит для ADQM	38
3.3.1	Цель	38
3.3.2	Дистрибутив.....	38
3.3.3	Порядок развертывания.....	38
3.3.4	Публикация метамодели Аудита ADQM.....	39
3.4	Плейбук настройки для Аудита ADQM	39
3.4.1	Цель	39
3.4.2	Дистрибутив.....	39
3.4.3	Порядок развертывания.....	40
3.5	Плейбук Аудит для ADH/ADS (ADPS).....	40
3.5.1	Цель	40
3.5.2	Дистрибутив.....	40
3.5.3	Порядок развертывания.....	41
3.5.4	Публикация метамодели Аудита ADPS.....	41
3.6	Плейбук Мониторинг - метрики Heart beat	42
3.6.1	Цель	42
3.6.2	Дистрибутив.....	42
3.6.3	Порядок развертывания.....	42
3.7	Плейбук Мониторинг - ADB метрики количества запросов	43
3.7.1	Цель	43
3.7.2	Дистрибутив.....	43
3.7.3	Порядок развертывания.....	43
3.8	Плейбук Мониторинг - пересылка метрик	44
3.8.1	Цель	44
3.8.2	Дистрибутив.....	44
3.8.3	Порядок развертывания.....	44

1 Общая информация

Развертывание компонентов интеграции продуктов Arenadata с Журналированием, Мониторингом, Аудитом реализована в виде централизованного Оркестратора, который на основе метаданных ADCM выполняет подготовку целевой конфигурации компонентов интеграции, генерирует инвентори-файлы, и выполняет ansible плейбуки для непосредственного выполнения установки.

1.1 Общие требования для использования

Запуск конфигурирования и установки штатно выполняется с использованием Оркестратора, доступен запуск непосредственно на хосте, либо в рамках докер-контейнера (который включает все используемые зависимости).

Сетевой доступ:

- от управляющей машины Ansible ко всем серверам развертывания (с продуктами AD и интеграционными сервисами) по ssh
- от всех серверов развертывания к технологическим сервисам Гостех ЖАМ (подразумеваются технологические сервисы Журналирование (Ж), Аудит (А), Мониторинга(М)), керберос (используемый для работы ADH/ADS)
- VPN доступ ко всем серверам развертывания, управляющей машине Ansible и вспомогательным сервисам UI.

Технологические учетные записи:

- учетные записи с правами доступа для всех приемников ЖАМ
- набор сертификатов TLS/SSL для подключения к приемникам ЖАМ
- учетные записи с правами работы в WebUI интерфейсах ЖАМ, включая право публикации дашборда в Grafana Журналирования (для ADQM)
- пользователь Kerberos для доступа к Solr серверам ADPS - включая keytab файлы принцепала этого пользователя к конкретным серверам с Solr ADPS (для ADH, ADS)
- учетная запись ADQM с правами подключения к СУБД (Clickhouse) и чтения таблицы system_log - для выгрузки событий аудита (для ADQM)

Паспорт среды, включая:

- список серверов установки продуктов Arenadata (включая ip, полные dns-имена, SSH порт)
- учетные записи (логин и пароль) с sudo правами к этим серверам
- реквизиты доступа к технологическим сервисам ЖАМ.

Одинаковая временная зона на всех серверах развертывания продуктов в рамках одного инстанса ADCM.

1.1.1 Требования к хосту запуска оркестратора и на хостах установки компонентов интеграции

- ОС ALT p9 8.4.1-alt0.p9.1
- подключены репозитории Alt Linux
- ansible (2.9, 2.10)
 - community.docker - коллекции community.docker для ansible (https://docs.ansible.com/ansible/latest/collections/community/docker/docker_image_module.html)
- python (3.7.4-3.7.17)
- python пакеты (доступны через pip3)
 - Docker API >= 1.25,
 - python3-jmespath
 - python3-requests
- docker version 20.10.11, build dea9396
- docker-compose version 1.27.4.

Работоспособность компонентов интеграции и их конфигурирования / установки протестирована с указанными версиями ПО, работоспособность с другими версиями может потребовать дополнительных проверок и тестирования.

При выполнении установки из докер-контейнера Оркестратора многие требуемые зависимости уже установлены в докер-имидже контейнера.

1.1.2 Требования в отношении продуктов Arenadata

Версии продуктов Arenadata:

- ADCM:
 - докер-имидж hub.arenadata.io/adcm/adcm:2023.04.21.12
 - adcm_host_ssh_v2.10-1_community
- ADET: adcm_cluster_et_v2023020200-1_community

- ADB:
 - adcm_cluster_adb_v6.23.3_arenadata44_b1-1_enterprise
 - установка утилиты Ncat 7.80 (<https://nmap.org/ncat>) на сервере ADCC
 - установка утилиты psql 12.9 на сервере ADCC
- ADQM: adcm_cluster_adqm_v22.8.12.45_arenadata1_b1-1_enterprise
- ADH: adcm_cluster_hadoop_v2.1.8_b3-1_enterprise
- ADS: adcm_cluster_ads_v1.8.1_b1-1_enterprise
- ADPS: adcm_cluster_platform_security_v1.0.4_b4-1_enterprise

Работоспособность компонентов интеграции и их конфигурирования / установки протестирована с указанными версиями ПО, работоспособность с другими версиями может потребовать дополнительных проверок и тестирования.

Настройки продуктов Arenadata согласно инструкциям Arenadata для развертывания на Платформе ГосТех.

1.1.3 Требования к технологическим сервисам ЖАМ

Предоставление интерфейсов интеграции согласно версии API на 27 октября 2023 (см. <https://platform-docs.v-serv.ru/online-documentation/home/maintain-tools/iam/> разделы 1.14 - 1.16 на указанную дату).

1.2 Выполнение установки

Выполнение установки компонентов интеграции включает:

- подготовку необходимого для развертывания и выполнение установки через запуск Оркестратора (см. в "Оркестратор")
- публикацию дашборда ADQM в WebUI Журналирование (Grafana)
- проверка корректной работы компонентов интеграции - наличие данных от развернутых продуктов Arenadata в WebUI Журналирования, Аудита и Мониторинга (для ADQM с использованием опубликованного дашборда)

Дистрибутив поставляется в виде TAR архива, перед использованием его необходимо распаковать:

```
tar -xvzf distr-1.0.1-20.tar.gz
```

Развернутый архив включает каталоги:

- ansible - каталог с файлами плейбуков ansible для установки компонентов интеграции
- orchestrator - каталог с файлами оркестратора
- images - каталог с файлами докер-имиджей
- docker-compose.yml - докер-файл для запуска оркестратора в контейнере
- start-docker.sh - файл старта оркестратора в рамках контейнера с необходимыми зависимостями
- VERSION - файл с версиями докер-имиджей компонентов интеграции

1.2.1 Установка дашборда Журналирования для ADQM

Установка требует для пользователя права администратора Grafana на момент выполнения установки.

Файл дашборда размещается в дистрибутиве плейбука "Мониторинг - пересылка метрик" `./ansible/deploy-mon-adcm-resender/files/grafana/AD-dashboard-prometheus-20231023.json`.

Для установки в WebUI Журналирования (Grafana) необходимо вызвать импорт дашборда из файла, и выбрать указанный файл для загрузки.

При корректной установке и штатной работе компонент интеграции при установке ADQM спустя некоторую задержку по времени должны начать поступать метрики мониторинга, в том числе отображаемые на импортированном дашборде.

2 Оркестратор

2.1 Описание реализации и запуск Оркестратора

Оркестратор реализован на Python3, с использованием дополнительных файлов и Python 3 библиотек, и предназначен для запуска всей процедуры установки компонентов интеграции.

Последовательно выполняет шаги:

- Выгрузка метаданных из ADCM - по хостам и компонентам интеграции (с сохранением в файл)
- Подготовка метаданных для конфигурирования и установки компонентов.
- Генерация файлов - включая инвентори-файлы для плейбуков
- Последовательный запуск выполнения плейбуков

Параметры Оркестратора вынесены в скрипт `./Python/setup_adcm_integration_env.py`, запускаемый скрипт `./Python/setup_adcm_integration.py`, запуск установки выполняется из командной строки (терминала):

```
python3 ./python/setup_adcm_integration.py
```

Для запуска Оркестратора в контейнере необходимо:

- все используемые дополнительные файлы разместить в каталоге `./files`, и в конфиге - файле оркестратора использовать относительные пути к ним `./files/...`, это позволит использовать один и тот же конфиг-файл оркестратора для запуска и в контейнере и на хосте
- стартовать докер-контейнер скриптом, войти в контейнер - и выполнить ту же команду что на хосте.

```
sudo start-docker.sh  
sudo docker exec -it orchestrator bash  
python3 ./python/setup_adcm_integration.py
```

Если запуск контейнера Оркестратора выполняется повторно - необходимо предварительно выполнить остановку контейнера (может потребоваться при обновлении дистрибутива компонент интеграции), либо войти в существующий - если сам оркестратор и его контейнера не менялись.

```
sudo docker stop orchestrator  
sudo docker rm orchestrator
```


При запуске Оркестратора выполняется одна полная итерация, которая выполняет все шаги.

2.2 До первого запуска - настройка

Необходимо актуализировать информацию в конфиг-файле Оркестратора:

- реквизиты подключения к ADCM (включая секрет vault)
- реквизиты подключения к Журналированию
- реквизиты подключения к Мониторингу
- реквизиты подключения к Аудиту
- выставить режим чтения метаданных из ADCM (REFRESH_ADCM_META_FLG = True), но без выполнения установки (ANSIBLE_EXEC_FLG = False).

Перед первичным запуском установки оркестратора администратору требуется выложить доступные оркестратору файлы с актуальными TLS сертификатами доступа к централизованным системам Журналирования и Аудита, указать следующие параметры среды.

2.2.1 Реквизиты доступа к ADCM API

В параметрах ADCM_* содержится информация по подключению оркестратора к ADCM API, к конфигурации продуктов из которого выполняется развертывание компонентов интеграции.

```
ADCM_LOGIN      = 'admin'  
ADCM_PASSWORD   = 'admin'  
ADCM_HOST       = 'adcm-arenadata.dvpaio.novalocal'  
ADCM_PORT       = 8000  
ADCM_TIMEOUT    = 2000  
ADCM_API_PATH_ROOT = '/api/v1'  
ADCM_STATUS_PATH_ROOT = '/status/api/v1'
```

В случае необходимости трассировки какие запросы к ADCM отправляются и какие ответы возвращаются - есть возможность включить записи в файлы всех выполняемых к ADCM API запросов и ответов. Для этого используются параметры ADCM_REPLIES_SAVE_FLG (True - выполнять сбор), и ADCM_REPLIES_DIR (каталог куда записывать файлы с ответами ADCM API).

```
ADCM_REPLIES_SAVE_FLG = False  
ADCM_REPLIES_DIR      = './tmp/adcm_replies'
```

2.2.1.1 Пароль ADCM vault

Для выполнения запуска ansible плейбуков с использованием зашифрованных значений используется шифрование пароля. Значения паролей из ADCM возвращаются в зашифрованном виде. Для возможности использования этих паролей при выполнении установки необходимо указать оркестратору файл в котором хранится пароль vault ADCM.

Пароль размещается как правило в файле `/opt/adcm/var/secrets.json` на хосте запуска ADCM контейнера, значение из его атрибута "password" необходимо скопировать как текст в файл, указываемый в параметре Оркестратора `CONST_VAULT_PASSWORD_FILE` (по умолчанию `./files/env/vault_password.txt`) без каких-либо дополнительных кавычек, см. пример.

```
CONST_VAULT_PASSWORD_FILE = './files/env/vault_password.txt'
```

2.2.2 Реквизиты подключения к Журналированию

В параметрах `LOG_TARGET_*` содержится информация по подключению к Журналированию для передачи содержимого логов - реквизиты приемника и файлы TLS сертификата. В случае если аутентификация делается только с использованием CA сертификата, параметры `key` и `cert` файлом необходимо указать как пустые строки. Если приемник без TLS аутентификации и доступен по HTTP - параметры с файлами TLS можно не заполнять (указать как пустые строки):

```
LOG_TARGET_URL = "https://logstash-client-02-arenadata.dvpaio.novalocal:5045"  
LOG_TARGET_TLS_HOSTPATH_CA_CERT = "./files/env/certs/mtls-root-cert.pem"  
LOG_TARGET_TLS_HOSTPATH_USER_CERT = "./files/env/certs/ad-client-cert.pem"  
LOG_TARGET_TLS_HOSTPATH_USER_KEY = "./files/env/certs/tls.key"
```

При передаче логов происходит приведения времени из файла к Epoch формату. В отдельных форматах лог-файлов временная зона отсутствует, и применяется приведение согласно заданной временной зоне. По умолчанию ожидается что все сервера работают во временной зоне МСК ('Europe/Moscow') и все файлы журналов на них ведутся согласно этой временной зоне. При необходимости это значение может быть изменено в параметре `LOG_LOGSTASH_TIMEZONE` (см. варианты значений названий временной зоны на странице https://en.wikipedia.org/wiki/List_of_tz_database_time_zones)

```
LOG_LOGSTASH_TIMEZONE = 'Europe/Moscow'
```

В случае если необходимо изменить временную зону для отдельных компонентов, параметр LOG_LOGSTASH_TIMEZONE может быть задан для конкретного компонента в параметре CUSTOM_PARAM_BY_KEY по ключу компонента. Ключ формируется согласно метаданных компонента в ADCM API, включает название продукта, название сервиса и название компонента, перечисленные через двоеточие. См. пример настройки для Web сервера Aiflow и для сервера Kafka Manager:

```
CUSTOM_PARAM_BY_KEY = {  
  # Для Airflow временная зона UTC  
  'adh:airflow:webserver': {  
    "LOG_LOGSTASH_TIMEZONE": 'Etc/UTC'  
  },  
  # Для Kafka-broker временная зона UTC  
  'ads:kafka_manager:server': {  
    "LOG_LOGSTASH_TIMEZONE": 'Etc/UTC'  
  }  
}
```

2.2.3 Реквизиты подключения к Аудиту

В параметрах AUDIT_TARGET_* содержится информация по подключению к Аудиту для передачи событий Аудита и регистрации метамодели:

```
AUDIT_TARGET_HOST = 'fis-mtls.gt-sol-dev-coreplatform-01.apps.k8s.dvp.tech'  
AUDIT_TARGET_PORT = '443'  
AUDIT_TARGET_EVENT_PATH = '/audit/proxy/v2/rn/inv-pms-arenadat/event'  
AUDIT_TARGET_METAMODEL_PATH = '/audit/proxy/v2/rn/inv-pms-arenadat/metamodel'  
AUDIT_TARGET_TLS_HOSTPATH_CA_CERT = './files/certs/mtls-root-cert.pem'  
AUDIT_TARGET_TLS_HOSTPATH_USER_CERT = './files/certs/ad-client-cert.pem'  
AUDIT_TARGET_TLS_HOSTPATH_USER_KEY = './files/certs/ad-client-key.pem'  
AUDIT_TARGET_TLS_TOP8_HOSTPATH_USER_KEY = './files/certs/tls-top8.key'
```

Сертификат для Аудита ADB необходим в формате для использования в Logstash в формате TOP8, он может быть получен из исходного ключа в формате KEY используя утилиту openssl:

```
openssl pkcs8 -topk8 -nocrypt -in tls.key -out tls-top8.key
```

По необходимости можно изменить параметры интеграции с Аудит развертывании конкретных продуктов - ADB, ADQM, ADH и/или ADS (интегрируются через ADPS). Они содержатся в параметрах AUDIT_ADB_*, AUDIT_ADQM_* и AUDIT_ADPS_* соответственно.

2.2.4 Реквизиты подключения к Мониторингу

В параметрах `MONITORING_TARGET_*` содержится информация по подключению к Мониторингу для передачи метрик мониторинга. Аутентификация при этом не производится.

Пересылка метрик может быть включена на приемник в формате Prometheus Push Gateway, либо в формате Graphite, либо одновременно в оба.

Режим подключения (`prometheus`, `graphite`, `both`) указываются в параметре `MONITORING_TARGET_MODE`.

Параметры приемника Graphite (этим приемником может выступать штатный Graphite Arenadata, запущенный с приемником метрик на альтернативном порту):

```
MONITORING_TARGET_GRAPHITE_HOST = "adcm-arenadata.dvpaio.novalocal"  
MONITORING_TARGET_GRAPHITE_PORT = "8015"
```

В случае если пересылка выполняется на штатный Graphite Arenadata - целесообразно отключить выполнения остановки Graphite при развертывании "Мониторинг - пересылка метрик", используя выставление параметра `MONITORING_AD_GRAPHITE_OFF` в `False` (по умолчанию он имеет значение `True`, так как этот компонент устанавливается с запуском работы на том же порту).

Параметры приемника Prometheus:

```
MONITORING_TARGET_PROMETHEUS_HOST = "vmagent-arenadata.dvpaio.novalocal"  
MONITORING_TARGET_PROMETHEUS_PORT = "8080"
```

2.2.5 Реквизиты доступа к кластерам-источникам событий аудита

Для получения данных событий аудита из источников необходимо для конкретных кластеров ADQM и ADH/ADS (ADPS) подготовить учетные данные для аутентификации при доступе к этим системам. Сопутствующие файлы как правила размещаются в каталоге `./files/env/`.

Эти реквизиты должны быть указаны в параметрах оркестратора после первого запуска с учетом дополнительной информации (см. "После первого запуска").

Параметры для аудита ADQM включают логин и пароль технологического пользователя, имеющего доступ к таблице `system.session_log` СУБД Clickhouse ADQM:

- `username` - логин тех. пользователя для подключения к СУБД Clickhouse (ADQM)

- password - пароль тех. пользователя для подключения к СУБД Clickhouse (ADQM).

Параметры для аудита ADH и ADS (подключение к Solr ADPS) включают параметры подключения принципалом kerberos технологического пользователя, к которому подключен Solr ADPS:

- username - логин тех. пользователя Solr
- realm - значение REALM kerberos, к которому подключен Solr (как правило полный домен в верхнем регистре)
- user_keytab_hostfile - полный путь к keytab файлу с принципалом тех. пользователя Solr для доступа к хосту Solr
- kdc_host - полное название хоста с KDC kerberos к которому подключен Solr.

2.2.6 Реквизиты подключения к серверу установки компонентов интеграции

Для выполнения установки отдельных компонентов интеграции задается хост для установки в параметрах DEPLOYMENT_*. По умолчанию в качестве этого сервера можно использовать сервер ADCM. См. пример:

```
DEPLOYMENT_HOST = "adcm-arenadata.dvpaio.novalocal"  
DEPLOYMENT_PORT = "9022"  
DEPLOYMENT_USER = "gt_prom"  
DEPLOYMENT_PASSWORD = "pw123"  
DEPLOYMENT_SSH_ARGS = "-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null"
```

На этом хосте разворачиваются следующие типы компонентов интеграции:

- Мониторинг - метрики heart beat
- Аудит ADQM
- Аудит ADPS

Для обеспечения безопасности пароль доступа к серверу развертывания компонентов (DEPLOYMENT_PASSWORD) можно не указывать в явном виде, если он совпадает с одним из серверов развертывания: пароль можно будет взять из файла метаданных в зашифрованном виде. При выполнении развертывания на этот сервер к паролю будет применяться расшифровка тем же секретом, что и для хостов описанных в ADCM (из файла согласно параметру CONST_VAULT_PASSWORD_FILE).

2.2.7 Информация о ПАК

При установке продуктов Arenadata на ПАК есть возможность указывать ID ПАК при передачи информации в тех.сервисы - для этого необходимо в JSON файле (задается параметром PAC_DICT_FILE) указать соотнесение к ID ПАК хостов. При этом рекомендуется указать не только полное доменное имя хоста, но IP адрес в виде строкового значения.

Хост и IP адрес при этом могут соответствовать только одному ПАК. Контроль отнесения доменного имени и соответствующего ему IP адреса к одному и тому же ПАК на стороне Оркестратора не выполняется, справочник используется "как есть".

В случае если файл не указан, или не содержит привязки хоста к ID ПАК, атрибуты заполняются пустой строкой.

См. пример содержимого файла для двух ПАК:

```
{
  "pac-id-1": [
    "host-name-1.domain",
    "host-name-2.domain",
    "172.21.22.23",
    "172.21.22.24"
  ],
  "pac-id-2": [
    "host-name-3.domain",
    "172.21.23.20"
  ]
}
```

2.3 После первого запуска

После первого запуска оркестратора в консоль выводится информация о ходе выполнения всех шагов. Выгруженные из ADCM метаданные и производные метаданные для развертывания сохраняются в файлах (названия файлов задаются параметрами ADCM_META_FILEPATH, OUT_INT_META_FILEPATH, LOG_OUT_META_FILEPATH).

В случае если каких-то параметров недостаточно - в консоль будут выведены соответствующие сообщения с указанием предупреждений и ошибок.

При последующих запусках Оркестратора (при ожидании что метаданные в ADCM не изменялись) актуализацию метаданных из ADCM можно временно отключить (REFRESH_ADCM_META_FLG = False).

2.3.1 При развертывании ADQM

Для ADQM будет выведено сообщение о том, что для компонента с указанным идентификатором не найдены метаданные, с перечислением каждого сервера СУБД в качестве компонента. См. ниже пример для кластера 14 на сервере adqm-arenadata.dvpaio.novalocal:

```
ADQM meta: cluster [3]
WARNING prepareADQMMeta (5): access meta does not found in SOURCE_ACCESS_PARAMS for
adqm.3:adqm-arenadata.dvpaio.novalocal, pass
```

В этом случае необходимо указать в параметрах для этого компонента логин и пароль (с указанным в сообщении идентификатором). Для кластера достаточно указать его для первого из перечисляемых компонентов. Пример значения параметра в файле параметров Оркестратора:

```
SOURCE_ACCESS_PARAMS = {
...
  "adqm.3:adqm-arenadata.dvpaio.novalocal": {
    "username": 'adqm_user_login',
    "password": 'adqm_user_password'
  },
...
}
```

2.3.2 При развертывании ADH или ADS

При развертывании ADH и/или ADS будет выведено сообщение о том, что для компонента с указанным идентификатором не найдены метаданные. В качестве компонента указывается хост Solr сервера ADPS, указываемого в метаданных ADCM для ADH или ADS кластера. См. ниже пример для кластера 14 на сервере Solr ADPS adps-arenadata.dvpaio.novalocal:

```
ADPS meta: cluster [14]
ADPS solr server: adps:solr:server on adps-arenadata.dvpaio.novalocal
(!) WARNING prepareADPSMeta (5): access meta does not found in SOURCE_ACCESS_PARAMS for
adps.14:adps-arenadata.dvpaio.novalocal, pass
```

В этом случае необходимо указать в параметрах для этого компонента необходимое для подключения к Solr ADPS, работающем с аутентификацией kerberos (с указанным в сообщении идентификатором):

- username - логин пользователя
- realm - REALM Kerberos для сервера Solr

- `user_keytab_hostfile` - путь к `keytab` файлу, доступного Оркестратору для чтения (как правило размещается в оркестраторе в каталоге `./files/env/{имя хоста}`)
- `kdc_host` - хост KDC Kerberos, с которым интегрирован Solr

Пример значения параметра в файле параметров Оркестратора:

```
SOURCE_ACCESS_PARAMS = {  
...  
  "adps.14:adps-arenadata.dvpaio.novalocal": {  
    "username": 'solr',  
    "realm": 'EXPERIMENTAL.NOVALOCAL',  
    "user_keytab_hostfile": './files/env/adps-arenadata.dvpaio.novalocal/solr.service.keytab',  
    "kdc_host": "pkles-gt0007068.experimental.novalocal"  
  }  
...  
}
```

2.3.3 Если в компонентах отсутствуют ожидаемые продукты

В случае если перечень компонентов для развертывания не показал ожидаемые - например при установке ADQM не отобразились компоненты для установки ADQM - необходимо проверить корректность справочника соответствия названий бандлов ADCM идентификаторам продуктов Arenadata. Для проверки названий бандлов - использовать WebUI интерфейс ADCM.

Справочник задается параметром Оркестратора `CONST_BUNDLE_PRODUCTS`, значения справочника по умолчанию:

```
{  
  "ADB": "adb",  
  "ADQM": "adqm",  
  "hadoop": "adh",  
  "ADS": "ads",  
  "Arenadata Enterprise Tools": "adet",  
  "platform_security": "adps"  
}
```

2.4 После второго запуска

В случае если все параметры Оркестратора были корректно заданы, не должно быть выведено сообщение об ошибках или предупреждений о том, что компонент не найден.

Для выполнения развертывания компонент интеграции ее необходимо активировать (`ANSIBLE_EXEC_FLG = True`).

2.5 Третий и последующие запуски

Третий и последующие запуски при включенном режиме установки производит непосредственно установку компонентов интеграции.

Для последовательной установки компонентом есть возможность комментировать в конфиг-файле Оркестратора типы плейбуков, выполняющие установку. В этом случае при наличии метаданных о выполнении установки будет выводиться предупреждение что соответствующего типа плейбук не найден. После снятия комментария с параметров и последующем запуске оркестратора будет выполнено выполнение всех установок с использованием этого типа плейбука.

Ниже пример значения параметра для выполнения установки компонентов с Журналированием, и пропуска установки Мониторинга (пересылка метрик):

```
ANSIBLE_META = {
# 'monitoring_heartbeats': {
#   'dir': '../ansible/deploy-mon-adcm-hball',
#   'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}'
# },
'log_all': {
  'dir': '../ansible/deploy-log-all',
  'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml -v --vault-password-file {{ vault_password_file }}'
},
}
```

Типы плейбуков и их соответствие компонентам интеграции приведены ниже.

Тип плейбука	Компонент
audit_adb_events	Аудит для ADB (включая метамодель)
audit_adps	Аудит для ADH и ADS (включая метамодель)
audit_adqm	Аудит для ADQM (включая метамодель)
audit_adqm_settings	Аудит для ADQM - настройка СУБД для сбора событий аудита
log_all	Журналирование
monitoring_adb_counts	Мониторинг - ADB метрики кол-ва запросов
monitoring_resend	Мониторинг - пересылка метрик
monitoring_heartbeats	Мониторинг - метрики heart beat

В текущей версии оркестратора поддерживается автоматизированное развертывание компонентов интеграции при условии описания в ADCM хостов с доступом к серверам по логину и паролю.

После выполнения установки требуется разово выполнить регистрацию модели метаданных - один раз для каждого продукта. Повторное выполнение регистрации требуется только в случае изменения метамодели - одновременно с переустановкой соответствующих компонентов интеграции.

В Оркестраторе предусмотрена автоматизированная публикация метамодели при каждом выполнении развертывания компонентов интеграции с Аудитом, для этого необходимо значения соответствующих параметров выставить в True:

- AUDIT_ADB_METAMODEL_PUBLISH_FLG - для ADB
- AUDIT_ADQM_METAMODEL_PUBLISH_FLG - для ADQM
- AUDIT_ADPS_METAMODEL_PUBLISH_FLG - для ADH и ADS (интеграция с ADPS).

2.6 Описание параметров запуска оркестратора

Все параметры Оркестратора вынесены в файл `./python/setup_adcm_integration_env.py`. Прочие параметры, не зависящие напрямую от метаданных ADCM и работы Оркестратора размещаются в компонентах плейбуков (в том числе в шаблонах inventory файлов).

2.6.1 Общие

Название параметра	Описание	Пример значения
CONST_TRACE_LEVEL	Уровень трассировки, общий для оркестратора. 0: информация (CONST_TRACE_LEVEL_INFO), 1: +ошибки (CONST_TRACE_LEVEL_ERROR), 2: +трассировка (CONST_TRACE_LEVEL_TRACE)	CONST_TRACE_LEVEL_ERROR
ANSIBLE_EXEC_FLG	Флаг выполнения плейбуков - в случае False выполняется только подготовка с генерацией всех файлов на основе метаданным, и вывод сообщений в терминал о том, какие плейбуки могли быть выполнены	False
REFRESH_ADCM_META_FLG	Флаг - выполнять чтение / обновление файла метаданных от ADCM	True
COMPONENTS_VERSIONS_PATH	Файл с версиями докер-имиджа. Если файл не указан, или в нем не указана версия для конкретного имиджа - используется значение latest	'../VERSION'
DEPLOYMENT_HOST	Хост для установки - с входом по SSH	adcm-arenadata.dvpaio.novalocal

Название параметра	Описание	Пример значения
DEPLOYMENT_PORT	Порт для входа по SSH	9022
DEPLOYMENT_USER	логин (пользователя с правом входа по SSH и SUDO доступом на сервере)	gt_prom
DEPLOYMENT_PASSWORD	Пароль для установки, в открытом виде или в зашифрованном формате с паролем ansible vault	\$ANSIBLE_VAULT;1.1;AES256\n356539653736343631313762303265303632343538653133613761613939313334373232366364623866623862613365653762383865623430363433336330320a383034326636396638643330323861383632303734623761383162646637623165393566383837333036636439373466656533663731323264383166613632650a3061646362666162386337613236323461623432653066623630333830343337
DEPLOYMENT_SSH_ARGS	Параметры для работы с SSH	-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null
PAC_DICT_FILE	Файл справочника отнесения ПАК ID к хостам, файла в формате JSON	"./files/env/pac_ids.json"

2.6.2 ADCM реквизиты подключения и настройки

Название параметра	Описание	Пример значения
ADCM_LOGIN	ADCM API: логин пользователя	admin'
ADCM_PASSWORD	ADCM API: пароль пользователя	admin'
ADCM_HOST	ADCM API: хост	adcm-arenadata.dvpaio.novalocal'
ADCM_PORT	ADCM API: порт	8000

Название параметра	Описание	Пример значения
ADCM_TIMEOUT	ADCM API: таймаут на HTTP запрос к API (в мс)	2000
ADCM_API_PATH_ROOT	ADCM API: путь в HTTP запросе к общим данным	/api/v1'
ADCM_STATUS_PATH_ROOT	ADCM API: путь в HTTP запросе к статусу компонента	/status/api/v1'
ADCM_REPLIES_SAVE_FLG	Флаг для сохранения ответов от всех запросов к ADCM	False
ADCM_REPLIES_DIR	Каталог для сохранения ответов ADCM (для тестирования / трассировки), каждый ответ отдельным файлом	./tmp/adcm_replies'
ADCM_META_FILEPATH	Файл с метаданными из ADCM	'./tmp/adcm_meta.json'
CONST_VAULT_PASSWORD_FILE	Файл с паролем к ansible vault ADCM. Пароль размещается как правило в файле /opt/adcm/var/secrets.json на хосте запуска ADCM контейнера, необходимо значение из атрибута "password"	'./files/vault_password.txt'
CONST_BUNDLE_PRODUCTS	Справочник соответствия названия бандла ID продукта в JSON формате. Требуется актуализации в случае изменения названия бандлов ADCM.	{ "ADB": "adb", "ADQM": "adqm", "hadoop": "adh", "ADS": "ads", "Arenadata Enterprise Tools": "adet", "platform_security": "adps" }
OUT_INT_META_FILEPATH	Файл - с итоговыми метаданными для установки компонент	./tmp/out_int_meta.json'

Название параметра	Описание	Пример значения
	интеграции (на основе метаданных ADCM и параметров оркестратора)	

2.6.3 Журналирование - реквизиты приемника и настройки

Название параметра	Описание	Пример значения
LOG_TARGET_URL	Полный URL метода с хостом и портом	"https://logstash-client-02-arenadata.dvpaio.novalocal:5045"
LOG_TARGET_TLS_DOCKER_PATH	Каталог с сертификатами - внутри контейнера, часть до /config - должна соответствовать logstash_home из inventory для log_all	"/usr/share/logstash/config/certs"
LOG_TARGET_TLS_HOSTPATH_CA_CERT	Полный файл с корневым сертификатом - на хосте плейбука, в формате pem	"./files/certs/mtls-root-cert.pem"
LOG_TARGET_TLS_HOSTPATH_USER_CERT	Полный файл с сертификатом пользователя - на хосте плейбука, в формате pem	"./files/certs/ad-client-cert.pem"
LOG_TARGET_TLS_HOSTPATH_USER_KEY	Полный файл с ключем пользователя - на хосте плейбука, в формате rsa	"./files/certs/tls.key"
LOG_DICT_LOGS_FILEPATH	Файл со справочником по логам потоков - в формате JSON массива	./files/logs/log_dict.json'
LOG_OUT_META_FILEPATH	Исходящий файл - с итоговым справочником, тиражированные конфигурации ADCM на потоки логов, с заменой всех параметров (каталоги путей, кол-во сегментов ADB)	'./tmp/log_out_config_meta.json'
LOG_OUT_DIR_HOST_COMPOSE	Каталог Исходящих файлов с каталогами в docker-	'./tmp/log_host_compose_dirs'

Название параметра	Описание	Пример значения
_DIRS	compose файл для каждого хоста	
LOG_OUT_INVENTORY_FILENAME	Исходящий файл с полным названием - inventory	'./tmp/inventory/inventory_log_all.yml'
LOG_INVENTORY_TEMPLATE	Входящий шаблон inventory файла	'./files/inventory_templates/log_all.yml'
LOG_IN_DIR_CONFIG_TEMPLATES	Корневой каталог для входящих шаблонов файлов-конфигов - берем из файлов плейбука	'./ansible/deploy-log-all-main/templates'
LOG_OUT_DIR_CONFIGS	Корневой каталог для исходящих файлов-конфигов - внутри будет выполняться создание каталогов для продуктов и в них файлы	'./tmp/log_configs'
LOG_LOGSTASH_TIMEZONE	Значение таймзоны на серверах - для атрибута TZ_values в конфигах Logstash Журналирования, используемого для приведения времени лог-файлов к Epoch формату. Возможные варианты - см. https://en.wikipedia.org/wiki/List_of_tz_database_time_zones	'Europe/Moscow'
LOG_PXF_CONF_VALUE	Значение переменной PXF_CONF для путей логов PXF ADB	'/var/lib/pxf'
LOG_ADH_AIRFLOW_LOG_FOLDER_DEFAULT	Размещение каталога для логов Airflow ADH	'/data/airflow/home/logs'
CUSTOM_PARAM_BY_KEY	Параметры компонентов, переопределяемых относительно значений общих параметров. Ключ формируется согласно метаданным компонента в ADCM API, включает название продукта, название	{ # Для Airflow временная зона UTC 'adh:airflow:webserver': {

Название параметра	Описание	Пример значения
	<p>сервиса и название компонента, перечисленные через двоеточие.</p> <p>Доступно определение параметра LOG_LOGSTASH_TIMEZONE - временной зоны, используемой для приведения времени событий из лог-файла к формату Epoch.</p>	<pre>"LOG_LOGSTASH_TIMEZONE": 'Etc/UTC' }, # Для Kafka-broker временная зона UTC 'ads:kafka_manager:server': { "LOG_LOGSTASH_TIMEZONE": 'Etc/UTC' } }</pre>

2.6.4 Аудит: реквизиты приемника и настройки

Название параметра	Описание	Пример значения
AUDIT_TARGET_HOST	Аудит приемник: хост	fis-mtls.gt-sol-dev-coreplatform-01.apps.k8s.dvp.tech'
AUDIT_TARGET_PORT	Аудит приемник: порт	443'
AUDIT_TARGET_EVENT_PATH	Аудит приемник: путь HTTP запроса для публикации событий	/audit/proxy/v2/rn/inv-pms-arenadat/event'
AUDIT_TARGET_METAMODEL_PATH	Аудит приемник: путь HTTP запроса для публикации метамодели	/audit/proxy/v2/rn/inv-pms-arenadat/metamodel'
AUDIT_TARGET_TLS_HOSTPATH_CA_CERT	Аудит приемник - TLS сертификаты доступа: путь к файлу с сертификатом CA в формате pem - на хосте	./files/certs/mtls-root-cert.pem'
AUDIT_TARGET_TLS_HOSTPATH_USER_CERT	Аудит приемник - TLS сертификаты доступа: путь к файлу с сертификатом пользователя в формате	./files/certs/ad-client-cert.pem'

Название параметра	Описание	Пример значения
	pem - на хосте	
AUDIT_TARGET_TLS_HOSTPATH_US ER_KEY	Аудит приемник - TLS сертификаты доступа: путь к файлу с ключем пользователя в формате pem - на хосте	./files/certs/ad-client-key.pem'
AUDIT_TARGET_TLS_TOP8_HOSTPA TH_USER_KEY	"Аудит приемник - TLS сертификаты доступа: путь к файлу с ключем пользователя в формате top8 - на хосте Формирование: openssl pkcs8 -topk8 -nocrypt -in tls.key -out tls-top8.key	'./files/certs/tls-top8.key'

2.6.5 Аудит: настройки для ADB

Название параметра	Описание	Пример значения
AUDIT_ADB_METAMODEL_MODULE	Аудит названия метамодели для ADB	"adb-audit"
AUDIT_ADB_METAMODEL_VERSION	Аудит версия метамодели для ADB	"10"
AUDIT_ADB_LOGFILEMASK	Лог-файл с событиями аудита ADB, ADB_MASTER_DATADIR - параметр из метаданных ADB clusters -> {ID} -> adbMeta -> master_datadir	{{ADB_MASTER_DATADIR}}/master/gpseg-1/pg_log/gpdb-*.csv'
AUDIT_ADB_EVENT_INVENTORY_TE MPLATE	Файл шаблона inventory файла событий Аудит ADB	./files/inventory_templates/audit_adb_events.yml'
AUDIT_ADB_EVENT_INVENTORY_O UT_FILE	Файл генерируемого inventory файла событий Аудит ADB	./tmp/inventory/inventory_audit_adb_events_{{cluster_id}}_{{host}}.yml'

Название параметра	Описание	Пример значения
AUDIT_ADB_METAMODEL_INVENTORY_TEMPLATE	Файл шаблона inventory файла метамодели Аудит ADB	./files/inventory_templates/audit_adb_metamodel.yml'
AUDIT_ADB_METAMODEL_INVENTORY_OUT_FILE	Файл генерируемого inventory файла метамодели Аудит ADB	./tmp/inventory/inventory_audit_adb_metamodel_{{cluster_id}}_{{host}}.yml'
AUDIT_ADB_TRACE_LEVEL	Уровень трассировки для Аудита ADB. 0: информация (CONST_TRACE_LEVEL_INFO), 1: +ошибки (CONST_TRACE_LEVEL_ERROR), 2: +трассировка (CONST_TRACE_LEVEL_TRACE)	CONST_TRACE_LEVEL_ERROR
AUDIT_ADB_APPLY_SETTINGS	Флаг для автоматизированного выполнения настроек ADB для вывод событий аудита в лог на стороне СУБД, с вызовом рестарта СУБД ADB при установке компонента интеграции	False
AUDIT_ADB_METAMODEL_PUBLICATION_FLG	Флаг автоматической публикации актуальной метамодели Аудита ADB при установке компонента интеграции	True

2.6.6 Аудит: настройки для ADQM

Название параметра	Описание	Пример значения
AUDIT_ADQM_METAMODEL_MODULE	Аудит названия метамодели для ADQM	"adqm-audit"

Название параметра	Описание	Пример значения
AUDIT_ADQM_METAMODEL_VERSION	Аудит версия метамодели для ADQM	"10"
AUDIT_ADQM_INVENTORY_TEMPLATE	Файл шаблона inventory файла Аудит ADQM	./files/inventory_templates/audit_adqm.yml'
AUDIT_ADQM_INVENTORY_OUT_FILE	Файл генерируемого inventory файла Аудит ADQM	./tmp/inventory/inventory_audit_adqm_{{cluster_id}}_{{host}}.yml'
AUDIT_ADQM_ITERATION_EVENTS_LIMIT	Лимит кол-ва на вычитывание событий Аудита ADQM (строк из Clickhouse)	"1000"
AUDIT_ADQM_TRACE_LEVEL	Уровень трассировки для Аудита ADQM. 0: информация (CONST_TRACE_LEVEL_INFO), 1: +ошибки (CONST_TRACE_LEVEL_ERROR), 2: +трассировка (CONST_TRACE_LEVEL_TRACE)	CONST_TRACE_LEVEL_ERROR
AUDIT_ADQM_APPLY_SETTINGS	Флаг для автоматизированного выполнения настроек ADQM для сбора событий аудита на стороне СУБД, с вызовом рестарта СУБД ADQM при установке компонента интеграции	False
AUDIT_ADQM_METAMODEL_PUBLICATION_FLG	Флаг автоматической публикации актуальной метамодели Аудита ADQM при установке компонента интеграции	True

2.6.7 Аудит: настройки для ADH и ADS (ADPS)

Название параметра	Описание	Пример значения
AUDIT_ADPS_METAMODEL_MODULE	Аудит названия метамодели для ADPS	"adps-audit"

Название параметра	Описание	Пример значения
AUDIT_ADPS_METAMODEL_VERSION	Аудит версия метамодели для ADPS	"10"
AUDIT_ADPS_INVENTORY_TEMPLATE	Файл шаблона inventory файла Аудит ADPS	'./files/inventory_templates/audit_adps.yml'
AUDIT_ADPS_INVENTORY_OUT_FILE	Файл генерируемого inventory файла Аудит ADPS	'./tmp/inventory/inventory_audit_adps_{{cluster_id}}_{{host}}.yml'
AUDIT_ADPS_ITERATION_EVENTS_LIMIT	Лимит кол-ва на вычитывание событий Аудита ADPS (строк из Solr ADPS)	"1000"
AUDIT_ADPS_TRACE_LEVEL	Уровень трассировки для Аудита ADPS. 0: информация (CONST_TRACE_LEVEL_INFO), 1: +ошибки (CONST_TRACE_LEVEL_ERROR), 2: +трассировка (CONST_TRACE_LEVEL_TRACE)	CONST_TRACE_LEVEL_ERROR
AUDIT_ADPS_EVENTS_DICT_HOSTFILE	Файл справочника отслеживаемых событий Аудита из ADPS - в формате JSON	'./files/audit-adps/audit_adps_dict.json'
AUDIT_ADPS_METAMODEL_PUBLICATION_FLG	Флаг автоматической публикации актуальной метамодели Аудита ADH/ADS (ADPS) при установке компонента интеграции	True

2.6.8 Мониторинг: реквизиты приемника и общие параметры

Название параметра	Описание	Пример значения
MONITORING_TARGET_GRAPHITE_HOST	Хост приемника метрик в формате Graphite	"adcm-arenadata.dvpaio.novalocal"
MONITORING_TARGET_GRAPHITE_PORT	Порт приемника метрик в формате Graphite	"8015"

Название параметра	Описание	Пример значения
MONITORING_TARGET_PROMETHEUS_HOST	Хост приемника метрик в формате Prometheus	"vmagent-arenadata.dvpaio.novalocal"
MONITORING_TARGET_PROMETHEUS_PORT	Порт приемника метрик в формате Prometheus	"8080"
MONITORING_TARGET_MODE	Режим работы компонента по целевым приемникам: <ul style="list-style-type: none"> • graphite - только в приемник Graphite • prometheus - только в приемник Prometheus • both - в оба приемника одновременно 	"prometheus"
MONITORING_AD_GRAPHITE_OFF	Флаг отключения при установке сервиса мониторинга Graphite на хосте развертывания	False
MONITORING_TARGET_TIMEOUT	Таймаут на HTTP запрос подключения к приемнику централизованного мониторинга (в мс)	2000
MONITORING_INTERNAL_TIMEOUT	Таймаут на HTTP запрос подключения к приемнику внутреннего мониторинга инстанса ADCM (в мс)	2000
MONITORING_RESENDER_INVENTORY_TEMPLATE	Файл шаблона inventory файла - пересылка метрик внутреннего мониторинга	'./files/inventory_templates/monitoring_resender.yml'
MONITORING_RESENDER_INVENTORY_OUTPUT_FILE	Файл генерируемого inventory файла - пересылка метрик внутреннего мониторинга	'./tmp/inventory/inventory_monitoring_resender.yml'

2.6.9 Мониторинг: метрики heartbeat

Название параметра	Описание	Пример значения
MONITORING_HEARTBEATS_INVENTORY_TEMPLATE	Файл шаблона inventory файла - снятия и отправки метрик Мониторинга на основе Heart beats ADCM	'./files/inventory_templates/monitoring_heartbeats.yml'
MONITORING_HEARTBEATS_INVENTORY_OUT_FILE	Файл генерируемого inventory файла - снятия и отправки метрик Мониторинга на основе Heart beats ADCM	'./tmp/inventory/inventory_monitoring_heartbeats.yml'
MONITORING_HEARTBEATS_TRACE_LEVEL	Уровень трассировки для снятия и отправки метрик Мониторинга на основе Heart beats ADCM. 0: информация (CONST_TRACE_LEVEL_INFO), 1: +ошибки (CONST_TRACE_LEVEL_ERROR), 2: +трассировка (CONST_TRACE_LEVEL_TRACE)	CONST_TRACE_LEVEL_ERROR

2.6.10 Мониторинг: ADB метрики кол-во запросов

Название параметра	Описание	Пример значения
MONITORING_ADB_COUNTS_INVENTORY_TEMPLATE	Файл шаблона inventory файла - снятия и отправки метрик Мониторинга для ADB (кол-ва по запросам)	'./files/inventory_templates/monitoring_adb_counts.yml'
MONITORING_ADB_COUNTS_INVENTORY_OUT_FILE	Файл генерируемого inventory файла - снятия и отправки метрик Мониторинга для ADB (кол-ва по запросам)	'./tmp/inventory/inventory_monitoring_adb_counts_{{host}}.yml'

2.6.11 Системы-источники

Название параметра	Описание	Пример значения
SOURCE_ACCESS_PARAMS	Параметры доступа к системам-источникам - логины / пароли / кей-таб файлы и принципалы.	{ "adqm.3:adqm-arenadata.dvpaio.novalocal": { "username": 'username',

Название параметра	Описание	Пример значения
	<p>Формат названия ключа для конкретного кластера (пример - adqm.3:adqm-arenadata.dvpaio.novalocal): {тип продукта AD}.{ID кластера согласно ADCM}:{Имя основного хоста кластера согласно ADCM}</p> <p>Параметры для аудита ADQM:</p> <ul style="list-style-type: none"> ▪ username - логин тех.пользователя для подключения к СУБД Clickhouse (ADQM) ▪ password - пароль тех.пользователя для подключения к СУБД Clickhouse (ADQM) <p>Параметры для аудита ADH и ADS (подключение к Solr ADPS):</p> <ul style="list-style-type: none"> ▪ username - логин тех.пользователя Solr ▪ realm - значение REALM kerberos, к которому подключен Solr (как правило полный домен в верхнем регистре) ▪ user_keytab_hostfile - полный путь к keytab файлу с принципалом тех.пользователя Solr для доступа к хосту Solr ▪ kdc_host - полное название хоста с KDC kerberos к которому подключен Solr 	<pre> "password": 'pw123' }, "adps.12:adps-arenadata.dvpaio.novalocal": { "username": 'user2', "realm": 'DVPAIO.NOVOLOCAL', "user_keytab_hostfile": './files/env/adps-arenadata.dvpaio.novalocal/user2.keytab', "kdc_host": "kdc.dvpaio.novalocal" } </pre>

2.6.12 Ansible параметры для генерации и выполнения

Название параметра	Описание	Пример значения
ANSIBLE_PYTHON_INTERPRETER	Путь к исполняемому файлу Python 3	"/usr/bin/python3"
ANSIBLE_META	<p>Параметры вызова выполнения плейбуков в зависимости от его типа - директория и маска команды.</p> <p>В случае необходимости исключения из развертывания конкретного типа компонент интеграции - соответствующие этому типу ветви JSON должны быть закомментированы</p>	<pre>{ 'audit_adb_events': { 'dir': '../ansible/deploy-audit-adb', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml -v --vault-password-file {{ vault_password_file }}' }, 'audit_adqm': { 'dir': '../ansible/deploy-ts-audit-adqm', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}' }, 'audit_adps': { 'dir': '../ansible/deploy-typescript-audit-solr-kdc', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}' }, 'monitoring_resend': { 'dir': '../ansible/deploy-mon-adcm-resender', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}' }</pre>

Название параметра	Описание	Пример значения
		<pre> vault_password_file }}" }, 'monitoring_adb_counts': { 'dir': '../ansible/deploy-mon-adb-queries', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}" }, 'monitoring_heartbeats': { 'dir': '../ansible/deploy-mon-adcm-hball', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml --vault-password-file {{ vault_password_file }}" }, 'log_all': { 'dir': '../ansible/deploy-log-all', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml -v --vault-password-file {{ vault_password_file }}" }, 'audit_adqm_settings': { 'dir': '../ansible/deploy-audit-adqm-config', 'cmd': 'ansible-playbook -i {{ inventory_file }} main.yml -v --vault-password-file {{ vault_password_file }}" }, }, </pre>

Название параметра	Описание	Пример значения
		}
ANSIBLE_OUT_LOGS_CATALOG	Каталог для вывода лог-файлов с журналом выполнения плейбука. Название файла генерируется с включением типа плейбука и меткой времени начала выполнения плейбука.	'./tmp/logs'

3 Ansible плейбуки установки компонентов интеграции

3.1 Плейбук Журналирование

3.1.1 Цель

Устанавливает на все хосты компоненты интеграции с Журналированием, которые читают и отправляют содержимое файлов логов в централизованный приемник.

При установке:

- использует сгенерированные конфиг-файлы Logstash и файлы с перечнем каталогов, подключаемых к контейнеру.
- установку выполняет на каждый сервер, для которого сформирован хотя бы один конфиг-файл Logstash (есть хотя бы один поток мониторинга)
- выкладывает на каждый сервер конфиг-файлы Logstash, готовит файл `pipelines.yml` для Logstash с перечнем параллельно работающих пайпланов чтения логов
- выкладывает на каждый сервер докер-имидж Logstash, публикует его и запускает с использованием сгенерированного `docker-compose` файла, при генерации использует файл с перечнем подключаемых каталогов
- при включенной настройке - выполняет конфигурирование хостов с компонентами технологий Spark и Livy для сбора логов в целевом формате.

3.1.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `./ansible/deploy-log-all`.

Название докер-имиджа: `logstash-oss-with-opensearch-output-plugin`

3.1.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить `inventory`-файлы
- подготовить конфиг-файлы Logstash
- подготовить файлы с каталогами для каждого хоста
- выполнить запуск плейбука для выполнения установки на всех хостах.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На каждом из хостов установки, с которого должны публиковаться лога-файлы, работает докер-контейнер на основе имиджа logstash-oss-with-opensearch-output-plugin
- В логе контейнера отсутствуют ошибки
- Содержимое лог-файлов с задержкой (штатная задержка - до 10 секунд) отображаются в WebUI тех.сервиса Журналирование

В случае необходимости выполнить развертывание отдельного продукта Arenadata, необходимо выполнить запуск плейбука с дополнительным указанием продукта (допустимые значения - adb, ads, adh, adqm):

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем} -l {adb или ads или adh или adqm}
```

3.2 Плейбук Аудит для ADB

3.2.1 Цель

Устанавливает на хост серверов "мастер" и "стендбай" ADB компонент интеграции с Мониторингом, который на основе лог-файла фиксирует и отправляет события аудита в тех.сервис Аудит.

При установке:

- генерирует и выкладывает конфиг-файл Logstash для чтение лог-файла ADB и отправки метрик
- установку выполняет на каждый хост "мастер" и "стендбай" ADB
- выкладывает на сервер докер-имидж Logstash, публикует его и запускает с использованием сгенерированного docker-compose файла
- при выставленной настройке - выполняет включение настроек ADB для сбора событий аудита в лог и выполняет рестарт СУБД ADB.

3.2.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `./ansible/deploy-audit-ADB`.

Название докер-имиджа: `logstash-oss-with-opensearch-output-plugin`

3.2.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- выполнить запуск плейбука для выполнения установки.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На хосте установки работает докер-контейнер на основе имиджа logstash-oss-with-opensearch-output-plugin
- В логе контейнера отсутствуют ошибки
- После попытки подключения к ADB с не существующим пользователем, с задержкой (штатная задержка по умолчанию - до 10 секунд) отображается событие о неуспешной попытке подключения в WebUI тех.сервиса Аудит.

3.2.4 Публикация метамодели Аудита для ADB

Для публикации метамодели Аудита ADB в ручном режиме необходимо выполнить вызов скрипта в докер-файле (в примере название контейнера приведено с фиксированным ID сервиса и компонента, реальное название контейнера необходимо выяснить из инвентори-файла плейбука, или из перечня выполняемых контейнеров с префиксом "adb-audit_", или по названию каталога):

```
# войти в контейнер
docker exec -it adb-audit_4_507 bash

# выполнить скрипт публикации метамодели, выставив привилегию на запуск скрипта
chmod 777 /usr/share/logstash/metamodel/metamodel-audit-adb.sh
/usr/share/logstash/metamodel/metamodel-audit-adb.sh
```

Ожидаемый ответ успешного выполнения в консоли аналогичен следующему (название и версия согласно заданным в инвентори-файле):

```
Metamodel adb-audit version 12
STATUS_CODE: 201
{"id":"23cd6fc3953e1d96df15fd0cc37622bf"}
```

3.3 Плейбук Аудит для ADQM

3.3.1 Цель

Устанавливает на DEPLOYMENT хост компоненты интеграции с Аудитом для ADQM, которые читают и отправляют события из СУБД ADQM в централизованный приемник Аудита. Публикует метамодель для событий Аудита ADQM.

При установке:

- установку выполняет на один заданный сервер - для одного конкретного инстанса компонента интеграции (для одного кластера ADQM)
- выкладывает конфиг-файл для компонента интеграции (файл .env.js)
- выкладывает на сервер докер-имидж с компонентом интеграции, публикует его и запускает с использованием сгенерированного docker-compose файла
- публикует метамодель (при выставленном параметре автоустановки метамодели Аудита ADQM).

3.3.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `./ansible/deploy-ts-audit-adqm`.

Название докер-имиджа: `audit_adqm`

3.3.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- подготовить файлы с сертификатом TLS для Аудита
- выполнить запуск плейбука для выполнения установки на всех хостах.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На хосте установки DEPLOYMENT работает докер-контейнер на основе имиджа `audit_adqm`
- В логе контейнера отсутствуют ошибки

- В логе компонента отсутствуют ошибки (файл на хосте в каталоге установки, который включает ID сервисам и ID компонента, например /opt/adqmdb_3_146/logs/ cron_adqmdb_3_146.log)
- После попытки подключения с ADQM с не существующим пользователем, с задержкой (штатная задержка - до 90 секунд) отображается событие о неуспешной попытке подключения в WebUI тех.сервиса Аудит.

3.3.4 Публикация метамодели Аудита ADQM

Для публикации метамодели Аудита ADQM в ручном режиме необходимо выполнить вызов скрипта в докер-файле:

```
# войти в контейнер
docker exec -it audit_adqm sh
# сменить рабочую директорию
cd /usr/src/app/build
# запустить скрипт метамодели
node metamodel-app.js
```

Ожидаемый ответ успешного выполнения в консоли аналогичен следующему:

```
{"id":"71146465b7b1f82d11e80b5ab798f7fc"}
```

3.4 Плейбук настройки для Аудита ADQM

3.4.1 Цель

Проверяет и устанавливает на хостах СУБД ADQM режим работы со сбором событий Аудита.

При выполнении:

- проверяет наличие на заявленных хостах наличие конфг-файла с ожидаемой настройкой. Настройка может быть выполнена в нескольких разных файлах - но проверяется только файл, который выкладывается автоматизированно данным плейбуком
- по необходимости выкладывает конфиг-файл на сервер кластера ADQM
- если был выложен конфиг-файл хотя бы на один сервер кластера и выставлен флаг применения настроек - выполняет рестарт кластера.

3.4.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог "./ansible/deploy-audit-adqm-config".

3.4.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги конфигурируются автоматизированного Оркестратором, но выполнение не производится - вместо этого выдается сообщение о ручном выполнении плейбука):

- подготовить inventory-файл
- выполнить запуск плейбука для выполнения установки на всех хостах.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На сервере ADQM в таблице system_log фиксируются все события подключения к СУБД и отключения от него.

3.5 Плейбук Аудит для ADH/ADS (ADPS)

3.5.1 Цель

Устанавливает на DEPLOYMENT хост компоненты интеграции с Аудитом для ADH и ADS через ADPS, который читает и отправляет события из Solr ADPS в централизованный приемник Аудита. Публикует метамодель для событий Аудита ADPS.

При установке:

- установку выполняет на один заданный сервер - для одного конкретного инстанса компонента интеграции ADPS (для одного или нескольких кластеров ADH и ADS)
- выкладывает конфиг-файл для компонента интеграции (файл .env.js)
- выкладывает на сервер докер-имидж с компонентом интеграции, публикует его и запускает с использованием сгенерированного docker-compose файла. Имидж включает конфиг-файл для работы с KDC
- публикует метамодель (при выставленном параметре автоустановки метамодели Аудита ADPS).

3.5.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `"/ansible/deploy-typescript-audit-solr-kdc"`.

Название докер-имиджа: `audit_adps`.

3.5.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- подготовить файлы с сертификатом TLS для Аудита
- подготовить файлы-справочник с событиями аудита (по умолчанию размещается в Оркестраторе ./files/audit-adps/audit_adps_dict.json)
- подготовить keytab-файл для подключения к Solr под Kerberos
- выполнить запуск плейбука для выполнения установки на всех хостах.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На хосте установки DEPLOYMENT работает докер-контейнер на основе имиджа audit_adqm
- В логе контейнера отсутствуют ошибки
- В логе компонента отсутствуют ошибки (файл на хосте в каталоге установки, который включает ID сервисам и ID компонента, например /opt/solr_14_1579/logs/cron_solr_14_1579.log)
- С задержкой (штатная задержка по умолчанию - до 90 секунд) должны начать отображаться события от технологий Hadoop в WebUI тех.сервиса Аудит.

3.5.4 Публикация метамодели Аудита ADPS

Для публикации метамодели Аудита ADPS в ручном режиме необходимо выполнить вызов скрипта:

```
# войти в контейнер:  
docker exec -it audit_adps sh  
# сменить рабочую директорию:  
cd /usr/src/app/build  
# запустить скрипт метамодели:  
node metamodel-app.js
```

Ожидаемый ответ успешного выполнения в консоли аналогичен следующему:

```
{"id":"71146465b7b1f82d11e80b5ab798f7fc"}
```

3.6 Плейбук Мониторинг - метрики Heart beat

3.6.1 Цель

Устанавливает на DEPLOYMENT хост компоненты интеграции с Мониторингом, который считывает значения heart beat по всем сервисам и компонентам ADCM, и отправляет их как метрики во внутренний мониторинг ADCM (для дальнейшей пересылки на внешние приемники).

При установке:

- установку выполняет на один заданный сервер - один на инстанс ADCM
- выкладывает конфиг-файл для компонента интеграции (файл .env.js)
- выкладывает на сервер докер-имидж с компонентом интеграции, публикует его и запускает с использованием сгенерированного docker-compose файла.

3.6.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `"/.ansible/deploy-mon-adcm-hball"`.

Название докер-имиджа: `heartbeat`.

3.6.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- выполнить запуск плейбука для выполнения установки на всех хостах.

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На хосте установки DEPLOYMENT работает докер-контейнер на основе имиджа `heartbeat`
- В логе контейнера отсутствуют ошибки
- В логе компонента отсутствуют ошибки (файл на хосте в каталоге установки, по умолчанию `/opt/heartbeat/logs/`).

С задержкой (штатная задержка по умолчанию - порядка 90 секунд) должны начать сниматься и доставляться метрики мониторинга `heartbeat`, и при развернутом компоненте интеграции "Мониторинг - пересылка"

метрик" метрики с значениями должны отобразиться в WebUI тех.сервиса Мониторинг (в Grafana).

3.7 Плейбук Мониторинг - ADB метрики количества запросов

3.7.1 Цель

Устанавливает на хосты ADCC компонент интеграции с Мониторингом, который считывает значения количественных метрик по SQL запросам СУБД (принятым / выполняемым / обработанным), и отправляет их как метрики во внутренний мониторинг ADCM (для дальнейшей пересылки на внешние приемники).

При установке:

- установку выполняет на каждый сервер ADCC, для которого прописан хотя бы один подключенный кластер ADB
- выкладывает исполняемый bash-скрипт с заполненными значениями параметров работы, и используемый SQL скрипт для обращения с СУБД PostgreSQL ADCC.

3.7.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог `./ansible/deploy-mon-adcm-hball`. Docker имидж не используется - скрипты выполняются непосредственно на хосте.

3.7.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- выполнить запуск плейбука для выполнения установки на всех хостах

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- В логе компонента отсутствуют ошибки (файл на хосте в каталоге установки, `/home/gpadmin/arenadata_configs/gosteh_adb_metrics.log`)
- С задержкой (штатная задержка по умолчанию - порядка 90 секунд) должны начать сниматься и доставляться метрики мониторинга ADB, и при развернутом компоненте интеграции "Мониторинг -

пересылка метрик" метрики с значениями должны отобразиться в WebUI тех.сервиса Мониторинг (в Grafana).

3.8 Плейбук Мониторинг - пересылка метрик

3.8.1 Цель

Устанавливает на хост с штатным приемником метрик мониторинга Arenadata компонент интеграции с Мониторингом, который принимает и пересылает метрики мониторинга на внешний приемник / приемники. При пересылке в Prometheus - трансформирует названия метрик в формате Prometheus (с "лейблами").

При установке:

- генерирует и выкладывает конфиг-файл Logstash при приемки / отправки метрик, выкладывает файл используемой библиотеки трансформирования названий метрик на языке Ruby
- установку выполняет сервер, на котором согласно метаданным ADCM установлен внутренний мониторинг Arenadata
- выкладывает на сервер докер-имидж Logstash, публикует его и запускает с использованием сгенерированного docker-compose файла.

3.8.2 Дистрибутив

В дистрибутиве плейбук размещается по умолчанию в каталог "./ansible/deploy-mon-adcm-resender".

Название докер-имиджа: logstash-oss-with-opensearch-output-plugin.

3.8.3 Порядок развертывания

Порядок развертывания по умолчанию (все шаги выполняются автоматизированного Оркестратором):

- подготовить inventory-файл
- выполнить запуск плейбука для выполнения установки

```
ansible-playbook -i {название inventory файла} main.yml -v --vault-password-file {название файла с vault паролем}
```

Результатом успешного выполнения playbook-а является:

- Отсутствие ошибок в консоли при работе плейбука
- На хосте установки работает докер-контейнер на основе имиджа logstash-oss-with-opensearch-output-plugin
- В логе контейнера отсутствуют ошибки

- Значения метрик от продуктов Arenadata с задержкой (штатная задержка - до 10 секунд) отображаются в WebUI тех.сервиса Мониторинг (Grafana) в случае отправки в Prometheus.