



itsumma
emergency consulting

ИНСТРУКЦИЯ
по установке
Программного решения
для обработки и хранения данных
ITSumma Data Processing Platform

Москва

2023

СОДЕРЖАНИЕ

Назначение документа	3
Общие сведения	3
Определения, обозначения и сокращения	3
1. Описание дистрибутива	4
2. Требования к аппаратному и программному обеспечению	5
ITS DPP.MW	5
3. Установка необходимого ПО	8
4. Копирование и распаковка дистрибутива ПО	9
5. Установка ITS DPP.MW	13
6. Установка ITS DPP.ETL	17
7. Установка ITS DPP.MPP DB	19
8. Установка ITS DPP.DataLake	20
9. Установка ITS DPP.Analytics DB	21
10. Техническая поддержка	28

Назначение документа

Настоящий документ содержит инструкцию по установке Программного решения для обработки и хранения данных “ITSumma Data Processing Platform” (далее – ITS DPP, программное решение, ПО).

Общие сведения

Программное решение предназначено для работы с большими объемами потоков данных. Оно позволяет получать данные из различных внешних источников, обрабатывать их, хранить в структурированном и неструктурированном виде, а также раскрывать данные для внешних систем.

Определения, обозначения и сокращения

В настоящем документе применяются следующие термины с соответствующими определениями, а также обозначения и сокращения представлены ниже:

ОС	операционная система
ПО	программное обеспечение
ITS DPP	ITSumma Data Processing Platform

1. Описание дистрибутива

Дистрибутив представляет собой директорию с набором файлов для установки ПО:

Наименование файла с образом	Описание содержимого
ITS DPP.MW (Managed Workflows)	<ul style="list-style-type: none"> - файл <code>playbook.yml</code> содержащий набор инструкций для установки модуля - файл <code>requirements.yml</code> содержащий набор ansible ролей для установки модуля - файлы содержащие примеры конфигураций для установки модуля - инструкция по установке модуля - инструкция по эксплуатации модуля
ITS DPP.ETL	<ul style="list-style-type: none"> - файл <code>playbook.yml</code> содержащий набор инструкций для установки модуля - файл <code>requirements.yml</code> содержащий набор ansible ролей для установки модуля - файлы содержащие примеры конфигураций для установки модуля - инструкция по установке модуля - инструкция по эксплуатации модуля
ITS DPP.MPP DB	<ul style="list-style-type: none"> - файл <code>playbook.yml</code> содержащий набор инструкций для установки модуля - файл <code>requirements.yml</code> содержащий набор ansible ролей для установки модуля - файлы содержащие примеры конфигураций для установки модуля - инструкция по установке модуля - инструкция по эксплуатации модуля
ITS DPP.Analytics DB	<ul style="list-style-type: none"> - файл <code>playbook.yml</code> содержащий набор инструкций для установки модуля - файл <code>requirements.yml</code> содержащий набор ansible ролей для установки модуля - файлы содержащие примеры конфигураций для установки модуля - инструкция по установке модуля

	<ul style="list-style-type: none">- инструкция по эксплуатации модуля
ITS DPP.DataLake	<ul style="list-style-type: none">- файл <code>playbook.yml</code> содержащий набор инструкций для установки модуля- файл <code>requirements.yml</code> содержащий набор <code>ansible</code> ролей для установки модуля- файлы содержащие примеры конфигураций для установки модуля- инструкция по установке модуля- инструкция по эксплуатации модуля

2. Требования к аппаратному и программному обеспечению ITS DPP.MW

Для установки ПО требуется физическая или виртуальная машина (сервер) с обязательным свободным доступом к сети интернет со следующими характеристиками:

Наименование характеристики	Минимально допустимое значение	Рекомендуемое значение
Процессор, ядер	4	8
Оперативная память, Гб	16	32
Жесткий диск (тип)	HDD	SSD
Жесткий диск (объем), Гб	10	40

ITS DPP.ETL

Для установки ПО требуется кластер из трех или более физических или виртуальных машин (серверов) с обязательным свободным доступом к сети интернет со следующими характеристиками:

Наименование характеристики	Минимально допустимое значение	Рекомендуемое значение
Процессор, ядер	8	8
Оперативная память, Гб	32	32
Жесткий диск (тип)	HDD	SSD
Жесткий диск (объем), Гб	10	40

ITS DPP.MPP DB

Для установки ПО требуется кластер из трех или более физических или виртуальных машин (серверов) с обязательным свободным доступом к сети интернет со следующими характеристиками:

Наименование характеристики	Минимально допустимое значение	Рекомендуемое значение
Процессор, ядер	8	8
Оперативная память, Гб	32	32
Жесткий диск (тип)	HDD	SSD
Жесткий диск (объем), Гб	10	40

ITS DPP.Analytics DB

Для установки ПО требуется кластер из трех или более физических или виртуальных машин (серверов) с обязательным свободным доступом к сети интернет со следующими характеристиками:

Наименование характеристики	Минимально допустимое значение	Рекомендуемое значение
Процессор, ядер	4	8
Оперативная память, Гб	16	32
Жесткий диск (тип)	HDD	SSD
Жесткий диск (объем), Гб	10	40

ITS DPP.DataLake

Для установки ПО требуется кластер из трех или более физических или виртуальных машин (серверов) с обязательным свободным доступом к сети интернет со следующими характеристиками:

Наименование характеристики	Минимально допустимое значение	Рекомендуемое значение
Процессор, ядер	8	8
Оперативная память, Гб	32	32
Жесткий диск (тип)	HDD	SSD
Жесткий диск (объем), Гб	20	80

К программному обеспечению, необходимому для установки и настройки ПО предъявляются следующие требования:

ITS DPP.MW

Перечень необходимого ПО	Минимальная версия	Рекомендуемая версия
ОС	<ul style="list-style-type: none"> - Ubuntu - CentOS - Astra Linux - РЕД ОС 	Ubuntu 20.4

ITS DPP. ETL

Перечень необходимого ПО	Минимальная версия	Рекомендуемая версия
ОС	<ul style="list-style-type: none"> - Ubuntu - CentOS - Astra Linux - РЕД ОС 	Ubuntu 20.4

ITS DPP.MPP DB

Перечень необходимого ПО	Минимальная версия	Рекомендуемая версия
ОС	<ul style="list-style-type: none"> - Ubuntu - CentOS - Astra Linux - РЕД ОС 	Ubuntu 18.4

ITS DPP.DataLake

Перечень необходимого ПО	Минимальная версия	Рекомендуемая версия
ОС	<ul style="list-style-type: none">- Ubuntu- CentOS- Astra Linux- РЕД ОС	Ubuntu 20.4

ITS DPP.Analytics DB

Перечень необходимого ПО	Минимальная версия	Рекомендуемая версия
ОС	<ul style="list-style-type: none">- Ubuntu- CentOS- Astra Linux- РЕД ОС	Ubuntu 20.4

3. Установка необходимого ПО

Для установки модулей платформы ITS DPP необходимо подготовить сервер, с которого будет производиться установка модулей, на остальную инфраструктуру выделенную под платформу.

На данный сервер необходимо установить Ansible.

Python 3.10

1. `sudo apt update`
2. `sudo apt install software-properties-common`
3. `sudo add-apt-repository ppa:deadsnakes/ppa`
4. `sudo apt update`
5. `sudo apt install python3.10`
6. `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
7. `python get-pip.py`

Ansible

Инструкции по установке приведены для ОС Ubuntu.

1. `python3 -m pip install --user ansible==6.6.0`

4. Копирование и распаковка дистрибутива ПО

Дистрибутив ПО поставляется в следующем формате:

ITS DPP.MW

Директория со следующим содержимым:

- директория examples
 - директория cluster
 - директория group_vars
 - airflow.yaml - конфигурационный файл с переменными для Airflow
 - postgresql.yaml - конфигурационный файл с переменными для Postgresql
 - redis.yaml - конфигурационный файл с переменными для Redis
 - директория host_vars - в этой директории хранятся примеры для конфигурации каждого из узлов кластера
 - hosts.yaml - конфигурационный файл, используемый для указания групп серверов для применения Ansible
 - директория standalone
 - директория group_vars
 - airflow.yaml - конфигурационный файл с переменными для Airflow
 - postgresql.yaml - конфигурационный файл с переменными для Postgresql
 - hosts.yaml - конфигурационный файл, используемый для указания групп серверов для применения Ansible
- playbook.yaml - основной конфигурационный файл для запуска Ansible
- requirements.yaml - конфигурационный файл с указанием зависимостей для установки модуля
- директория roles
 - директория common - содержит набор конфигурационных файлов для Ansible, служащих для подготовки сервера к установке модуля
 - node_exporter - содержит набор конфигурационных файлов для Ansible, служащих для установки агентов мониторинга
 - docker - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Docker
 - postgresql - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Postgresql
 - airflow - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Airflow
 - redis - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Redis

ITS DPP.ETL

Директория со следующим содержимым:

- директория `examples/cluster`
 - директория `group_vars`
 - `kafka.yaml` - конфигурационный файл с переменными для Kafka
 - `redis.yaml` - конфигурационный файл с переменными для Redis
 - директория `host_vars` - в этой директории хранятся примеры для конфигурации каждого из узлов кластера
 - `hosts.yaml` - конфигурационный файл, используемый для указания групп серверов для применения Ansible
- `playbook.yaml` - основной конфигурационный файл для запуска Ansible
- `requirements.yaml` - конфигурационный файл с указанием зависимостей для установки модуля
- директория `roles`
 - директория `common` - содержит набор конфигурационных файлов для Ansible, служащих для подготовки сервера к установке модуля
 - `node_exporter` - содержит набор конфигурационных файлов для Ansible, служащих для установки агентов мониторинга
 - `java` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Java
 - `zookeeper` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Zookeeper
 - `docker` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Docker
 - `kafka` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Kafka
 - `redis` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Redis

ITS DPP.MPP DB

Директория со следующим содержимым:

- директория `examples/cluster`
 - директория `group_vars`
 - `greenplum_master.yaml` - конфигурационный файл с переменными для Greenplum
 - `hosts.yaml` - конфигурационный файл, используемый для указания групп серверов для применения Ansible
- `playbook.yaml` - основной конфигурационный файл для запуска Ansible

- requirements.yaml - конфигурационный файл с указанием зависимостей для установки модуля
- директория roles
 - директория common - содержит набор конфигурационных файлов для Ansible, служащих для подготовки сервера к установке модуля
 - node_exporter - содержит набор конфигурационных файлов для Ansible, служащих для установки агентов мониторинга
 - java - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Java
 - docker - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Docker
 - greenplum - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Greenplum

ITS DPP.Analytics DB

Директория со следующим содержимым:

- директория examples
 - директория cluster_v1
 - директория group_vars
 - clickhouse.yaml - конфигурационный файл с переменными для Clickhouse
 - postgresql.yaml - конфигурационный файл с переменными для Postgresql
 - redash.yaml - конфигурационный файл с переменными для Redash
 - redis_for_redash.yaml - конфигурационный файл с переменными для Redis в Redash
 - superset.yaml - конфигурационный файл с переменными для Superset
 - директория host_vars - в этой директории хранятся примеры для конфигурации каждого из узлов кластера
 - hosts.yaml - конфигурационный файл, используемый для указания групп серверов для применения Ansible
 - директория cluster_v2
 - директория group_vars
 - clickhouse.yaml - конфигурационный файл с переменными для Clickhouse
 - postgresql.yaml - конфигурационный файл с переменными для Postgresql
 - redash.yaml - конфигурационный файл с переменными для Redash
 - redis_for_redash.yaml - конфигурационный файл с переменными для Redis в Redash

- redis_for_superset.yaml - конфигурационный файл с переменными для Redis в Superset
- superset.yaml - конфигурационный файл с переменными для Superset
- директория host_vars - в этой директории хранятся примеры для конфигурации каждого из узлов кластера
- hosts.yaml - конфигурационный файл, используемый для указания групп серверов для применения Ansible
- playbook.yaml - основной конфигурационный файл для запуска Ansible
- requirements.yaml - конфигурационный файл с указанием зависимостей для установки модуля
- директория roles
 - директория common - содержит набор конфигурационных файлов для Ansible, служащих для подготовки сервера к установке модуля
 - node_exporter - содержит набор конфигурационных файлов для Ansible, служащих для установки агентов мониторинга
 - docker - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Docker
 - java - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Java
 - zookeeper - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Zookeeper
 - clickhouse - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Clickhouse
 - postgresql - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса PostgreSQL
 - redis - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Redis
 - superset - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Superset
 - redash - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Redash

ITS DPP.DataLake

Директория со следующим содержимым:

- директория examples/cluster
 - hosts.yaml - конфигурационный файл, используемый для указания групп серверов для применения Ansible
- playbook.yaml - основной конфигурационный файл для запуска Ansible
- requirements.yaml - конфигурационный файл с указанием зависимостей для установки модуля
- директория roles

- директория `common` - содержит набор конфигурационных файлов для Ansible, служащих для подготовки сервера к установке модуля
- `docker` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Docker
- `java` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Java
- `hadoop` - содержит набор конфигурационных файлов для Ansible, служащих для установки сервиса Hadoop

Необходимо разместить директорию на сервере с которого будет производиться установка. Сервер должен иметь SSH доступ до целевых серверов, на которых будет размещаться платформа.

5. Установка ITS DPP.MW

Установка

1. Установить необходимые зависимости

```
ansible-galaxy install -r requirements.yml
```

2. Оформить inventory и переменные окружения

Примеры inventory.yml и group/host-переменных доступны в examples-директории. Для запуска standalone версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/standalone/hosts.yml) указав вместо примера ip, ip сервера, на который необходимо установить модуль.
- создать в директории модуля директорию group_vars (пример содержимого директории ./examples/standalone/group_vars) указав в файлах переменных необходимые параметры:

group_vars/airflow.yml

```
airflow_cluster_enabled: false Airflow запущен в режиме standalone
airflow_database_postgresql_name: airflow_db имя базы данных для Airflow
в СУБД Postgresql
airflow_database_postgresql_user: airflow_user логин для пользователя в
базе данных для Airflow в СУБД Postgresql
airflow_database_postgresql_password: 13Zz1MgwM2ID9Q4l пароль для
пользователя в базе данных для Airflow в СУБД Postgresql
airflow_database_postgresql_host: 127.0.0.1 ip сервера, на котором
расположена база данных Postgresql для Airflow
```

Для запуска cluster версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/cluster/hosts.yml) указав вместо примера ip, ip серверов, на которых необходимо установить модуль.
- создать в директории модуля директорию group_vars (пример содержимого директории ./examples/cluster/group_vars) указав в файлах переменных необходимые параметры:

group_vars/airflow.yml

```
airflow_cluster_enabled: true Airflow запущен в режиме cluster
airflow_database_postgresql_name: airflow_db имя базы данных для Airflow
в СУБД Postgresql
airflow_database_postgresql_user: airflow_user логин для пользователя в
базе данных для Airflow в СУБД Postgresql
```



```
airflow_database_postgresql_password: 13Zz1MgwM2ID9Q41 пароль для
пользователя в базе данных для Airflow в СУБД Postgresql
airflow_database_postgresql_host: 127.0.0.1 ip сервера, на котором
расположена база данных Postgresql для Airflow, по умолчанию
airflow_celery_broker_sentinel_hosts:
- sentinel://:zaAtnK1cOwwGDkwO@10.1.23.101:26379
- sentinel://:zaAtnK1cOwwGDkwO@10.1.23.102:26379
- sentinel://:zaAtnK1cOwwGDkwO@10.1.23.103:26379
здесь указаны данные для синхронизации узлов кластера Airflow через Redis
```

group_vars/postgresql.yml

```
postgresql_listen_addresses: 0.0.0.0
postgresql_create_users_bases_grants: true
postgresql_databases:
- name: airflow_db имя базы данных для Airflow
postgresql_users:
- name: airflow_user
  password: 13Zz1MgwM2ID9Q41 логин и пароль для пользователя в базе данных
для Airflow
postgresql_grants:
- database: airflow_db
  roles: airflow_user
  grant_option: true
  privs: ALL
  type: database
postgresql_hba_entries:
- {type: local, database: all, user: postgres, auth_method: peer}
- {type: local, database: all, user: all, auth_method: md5}
- {type: local, database: all, user: all, auth_method: peer}
- {type: host, database: all, user: postgres, address:
'127.0.0.1/32', auth_method: trust}
- {type: host, database: all, user: all, address: '10.1.23.0/24',
auth_method: md5} # Instances network указание на сеть, в которой
расположен кластер
- {type: host, database: all, user: all, address: '127.0.0.1/32',
auth_method: md5}
- {type: host, database: all, user: all, address: '::1/128',
auth_method: md5}
- {type: local, database: replication, user: all, auth_method:
peer}
- {type: host, database: replication, user: all, address:
'127.0.0.1/32', auth_method: md5}
- {type: host, database: replication, user: all, address:
'::1/128', auth_method: md5}
```

group_vars/redis.yml

```
redis_password: zaAtnK1cOwwGDkwO
redis_sentinel_enabled: true Redis запущен в режиме кластера
redis_replication_master_ip: 10.1.23.101 ip сервера на котором запущен
master для кластера
redis_sentinel_master_host: 10.1.23.101 ip сервера на котором запущен
master для кластера
```

- создать в директории модуля директорию host_vars (пример содержимого директории ./examples/cluster/host_vars)

host_vars/airflow-master.yml

```
redis_replication_role: master
airflow_cluster_role: master
```

host_vars/airflow-worker.yml

```
redis_replication_role: slave
airflow_cluster_role: worker
```

3. Запустить плейбук для установки модуля

```
ansible-playbook -i inventory.yml playbook.yml -D
```

Проверка

Для проверки работы модуля необходимо выполнить успешную авторизацию в Airflow, для этого:

1. Зайти по адресу <http://IP:8080>
2. Ввести логин и пароль для базового пользователя Airflow:
admin/Awa1jljZCiSdQvDA

6. Установка ITS DPP.ETL

Установка

1. Установить необходимые зависимости

```
ansible-galaxy install -r requirements.yml
```

2. Оформить inventory и переменные окружения

Примеры inventory.yml и group/host-переменных доступны в examples-директории. Для запуска cluster версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/cluster/hosts.yml) указав вместо примера ip, ip серверов, на которых необходимо установить модуль.
- создать в директории модуля директорию group_vars (пример содержимого директории ./examples/cluster/group_vars) указав в файлах переменных необходимые параметры обязательно необходимо указать:

group_vars/kafka.yml

```
kafka_zookeeper_hosts:  
  - 10.1.23.101:2181  
  - 10.1.23.102:2181  
  - 10.1.23.103:2181  
kafka_burrow_zookeeper_hosts:  
  - 10.1.23.101:2181  
  - 10.1.23.102:2181  
  - 10.1.23.103:2181  
kafka_burrow_kafka_hosts:  
  - 10.1.23.101:9092  
  - 10.1.23.102:9092  
  - 10.1.23.103:9092
```

в данном конфигурационном файле указываем ip для серверов кластера

group_vars/redis.yml

```
redis_password: MZYvoG4tWuboiMDW  
redis_sentinel_enabled: true Redis запущен в режиме кластера  
redis_replication_master_ip: 10.1.23.101 ip сервера на котором запущен  
master для кластера  
redis_sentinel_master_host: 10.1.23.101 ip сервера на котором запущен  
master для кластера
```

group_vars/spark.yml

```
spark_master_enabled: true
```

```
spark_worker_enabled: true

spark_worker_connect_url:
spark://dpp-etl-vm1:7077,dpp-etl-vm2:7077,dpp-etl-vm3:7077

spark_env_options:
  SPARK_MASTER_PORT: 7077
  SPARK_MASTER_WEBUI_PORT: 8081
  SPARK_WORKER_WEBUI_PORT: 8082
  SPARK_DAEMON_JAVA_OPTS: '"-Dspark.deploy.recoveryMode=ZOOKEEPER
-Dspark.deploy.zookeeper.url=10.1.23.101:2181,10.1.23.102:2181,10.1
.23.103:2181 -Dspark.deploy.zookeeper.dir=/spark"'
в данной переменной указываем ip серверов кластера, на которых развернут zookeeper.
  SPARK_WORKER_CORES: 1
  SPARK_WORKER_MEMORY: 1g
```

- создать в директории модуля директорию `host_vars` (пример содержимого директории `./examples/cluster/host_vars`) и указать там актуальные ip для каждого сервера.

3. Запустить плейбук для установки модуля

```
ansible-playbook -i inventory.yml playbook.yml -D
```

Проверка

Для проверки создадим в Kafka тестовый топик

```
./bin/kafka-topics.sh --create --topic test --bootstrap-server
127.0.0.1:9092
```

Ожидаемый ответ

```
Created topic registrations.
```

7. Установка ITS DPP.MPP DB

Установка

1. Установить необходимые зависимости

```
ansible-galaxy install -r requirements.yml
```

2. Оформить inventory и переменные окружения

Примеры inventory.yml и group/host-переменных доступны в examples-директории. Для запуска cluster версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/cluster/hosts.yml) указав вместо примера ip, ip серверов, на которых необходимо установить модуль.
- создать в директории модуля директорию group_vars (пример содержимого директории ./examples/cluster/group_vars) указав в файлах переменных необходимые параметры.

group_vars/greenplum.yml

```
greenplum_database_users: Описание набора пользователей для СУБД
  - name: "user1"
    db: "db"
    password: "22FQkaKkAgqmv4cc"
    priv: "ALL"
  - name: "user2"
    db: "db"
    password: "0ez1PqrF6SLzn6it"
    priv: "ALL"
greenplum_pg_hba:
  - users: "all"
    source: "10.1.23.0/24" указание на сеть, в которой расположен кластер
    databases: "all"
    method: "trust"
    contype: "host"
```

3. Запустить плейбук для установки модуля

```
ansible-playbook -i inventory.yml playbook.yml -D
```

Проверка

1. su - gpadmin -s /bin/bash
2. psql postgres

8. Установка ITS DPP.DataLake

Установка

1. Установить необходимые зависимости

```
ansible-galaxy install -r requirements.yml
```

2. Оформить inventory

Примеры inventory.yml и group/host-переменных доступны в examples-директории. Для запуска cluster версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/cluster/hosts.yaml) указав вместо примера ip, ip серверов, на которые необходимо установить модуль.

3. Запустить плейбук для установки модуля

```
ansible-playbook -i inventory.yml playbook.yml -D
```

Проверка

Проверить работу модуля, можно подключившись с любого из серверов кластера к Apache Hadoop.

```
su - hadoop
echo teststring > tst.file
hdfs dfs -put tst.file /
hdfs dfs -ls /
```

Ожидаемый результат

```
Found 1 items
-rw-r--r--      1 hadoop supergroup          11 2023-06-01 07:18
/tst.file
```

9. Установка ITS DPP.Analytics DB

Установка

1. Установить необходимые зависимости

```
ansible-galaxy install -r requirements.yml
```

2. Оформить inventory

Примеры inventory.yml и group/host-переменных доступны в examples-директории. Для запуска cluster версии необходимо:

- создать в директории модуля файл inventory.yml (пример содержимого ./examples/cluster/hosts.yml) указав вместо примера ip, ip серверов, на которые необходимо установить модуль.

3. Выбрать подходящую версию конфигурации для кластера:

- все необходимое ПО устанавливается на 3 сервера. Zookeeper, Clickhouse - кластерные. Superset, Postgresql, Redash, Redash - нет. Кластеры можно разделить по разным серверам по аналогии.
- все необходимое ПО устанавливается на 4 сервера. Zookeeper, Clickhouse, Superset, Redis для Superset - кластерные, Postgresql, Redash и Redis для него - нет. Кластеры можно разделить по разным серверам по аналогии.

4. Подготовить переменные окружения, в зависимости от выбранного типа кластера:

- для первой версии кластера примеры конфигураций находятся в директории examples/cluster_v1/group_vars

group_vars/clickhouse.yml

```
---
clickhouse_zookeeper_hosts: в данной переменной указываем ip серверов
кластера, на которых развернут zookeeper
- host: 10.1.23.101
- host: 10.1.23.102
- host: 10.1.23.103

clickhouse_clusters: в данной переменной указываем ip серверов кластера, на
которых будет развернут clickhouse
test_cluster:
  shard:
    - host: 10.1.23.101
    - host: 10.1.23.102
    - host: 10.1.23.103

clickhouse_users: описание пользователя для СУБД
user:
  password_sha256_hex:
"2923fd1ee2ff7e60268bb353e53dbffa9e18a5f677ef60599cf0138082b13dba"
  profile: "default"
  networks:
```

- 0.0.0.0

group_vars/postgresql.yml

```
postgresql_listen_addresses: 0.0.0.0
postgresql_create_users_bases_grants: true
postgresql_databases:
  - name: superset_db имя для базы данных для Superset
  - name: redash_db имя для базы данных для Redash
postgresql_users: описание пользователей для СУБД
  - name: superset_user
    password: PL5lyFceU62xVZl0
  - name: redash_user
    password: Cfoe3uDEbOr0MO2m
postgresql_grants:
  - database: superset_db
    roles: superset_user
    grant_option: true
    privs: ALL
    type: database
  - database: redash_db
    roles: redash_user
    grant_option: true
    privs: ALL
    type: database
postgresql_hba_entries:
  - {type: local, database: all, user: postgres, auth_method: peer}
  - {type: local, database: all, user: all, auth_method: md5}
  - {type: local, database: all, user: all, auth_method: peer}
  - {type: host, database: all, user: postgres, address:
'127.0.0.1/32', auth_method: trust}
  - {type: host, database: all, user: all, address: '10.1.23.0/24',
auth_method: md5}          # Instances network указание на сеть, в которой
расположен кластер

  - {type: host, database: all, user: all, address:
'172.16.0.0/12', auth_method: md5}          # Docker network (for
exporter)
  - {type: host, database: all, user: all, address: '127.0.0.1/32',
auth_method: md5}
  - {type: host, database: all, user: all, address: '::1/128',
auth_method: md5}
  - {type: local, database: replication, user: all, auth_method:
peer}
  - {type: host, database: replication, user: all, address:
'127.0.0.1/32', auth_method: md5}
  - {type: host, database: replication, user: all, address:
```



```
'::1/128', auth_method: md5}
```

group_vars/redash.yml

```
redash_postgresql_database: redash_db название базы данных для Redash в СУБД Postgresql  
redash_postgresql_user: redash_user описание пользователя для Redash в СУБД Postgresql  
redash_postgresql_password: Cfoe3uDEbOr0MO2m  
redash_postgresql_host: 10.1.23.101 ip сервера, на котором развернута СУБД Postgresql для Redash  
redash_redis_host: 10.1.23.101:6379 ip сервера, на котором развернута СУБД Redis для Redash  
redash_redis_password: cFSpbYUYGqcKG4Xj пароль для доступа к СУБД Redis
```

group_vars/redis_for_redash.yml

```
---  
redis_password: cFSpbYUYGqcKG4Xj пароль для доступа к СУБД Redis
```

group_vars/superset.yml

```
---  
superset_database_postgresql_name: superset_db название базы данных для Superset в СУБД Postgresql  
superset_database_postgresql_user: superset_user описание пользователя для Superset в СУБД Postgresql  
superset_database_postgresql_password: PL5lyFceU62xVZ10  
superset_database_postgresql_host: 10.1.23.101 ip сервера, на котором развернута СУБД Postgresql для Superset
```

- для второй версии кластера примеры конфигураций находятся в директории `examples/cluster_v2/group_vars`:

group_vars/clickhouse.yml

```
---
clickhouse_zookeeper_hosts: в данной переменной указываем ip серверов
кластера, на которых развернут zookeeper
- host: 10.1.23.101
- host: 10.1.23.102
- host: 10.1.23.103

clickhouse_clusters: в данной переменной указываем ip серверов кластера, на
которых будет развернут clickhouse
test_cluster:
  shard:
    - host: 10.1.23.101
    - host: 10.1.23.102
    - host: 10.1.23.103

clickhouse_users: описание пользователя для СУБД
user:
  password_sha256_hex:
"2923fd1ee2ff7e60268bb353e53dbffa9e18a5f677ef60599cf0138082b13dba"
  profile: "default"
  networks:
    - 0.0.0.0
```

group_vars/postgresql.yml

```
postgresql_listen_addresses: 0.0.0.0
postgresql_create_users_bases_grants: true
postgresql_databases:
  - name: superset_db имя для базы данных для Superset
  - name: redash_db имя для базы данных для Redash
postgresql_users: описание пользователей для СУБД
  - name: superset_user
    password: PL5lyFceU62xVZl0
  - name: redash_user
    password: Cfoe3uDEbOr0MO2m
postgresql_grants:
  - database: superset_db
    roles: superset_user
    grant_option: true
    privs: ALL
    type: database
  - database: redash_db
    roles: redash_user
    grant_option: true
```

```
privs: ALL
type: database
postgresql_hba_entries:
- {type: local, database: all, user: postgres, auth_method: peer}
- {type: local, database: all, user: all, auth_method: md5}
- {type: local, database: all, user: all, auth_method: peer}
- {type: host, database: all, user: postgres, address:
'127.0.0.1/32', auth_method: trust}
- {type: host, database: all, user: all, address: '10.1.23.0/24',
auth_method: md5}          # Instances network указание на сеть, в которой
расположен кластер

- {type: host, database: all, user: all, address:
'172.16.0.0/12', auth_method: md5}          # Docker network (for
exporter)
- {type: host, database: all, user: all, address: '127.0.0.1/32',
auth_method: md5}
- {type: host, database: all, user: all, address: '::1/128',
auth_method: md5}
- {type: local, database: replication, user: all, auth_method:
peer}
- {type: host, database: replication, user: all, address:
'127.0.0.1/32', auth_method: md5}
- {type: host, database: replication, user: all, address:
 '::1/128', auth_method: md5}
```

group_vars/redash.yml

```
redash_postgresql_database: redash_db название базы данных для Redash в
СУБД Postgresql
redash_postgresql_user: redash_user описание пользователя для Redash в
СУБД Postgresql
redash_postgresql_password: Cfoe3uDEbOr0MO2m
redash_postgresql_host: 10.1.23.101 ip сервера, на котором развернута СУБД
Postgresql для Redash
redash_redis_host: 10.1.23.101:6379 ip сервера, на котором развернута СУБД
Redis для Redash
redash_redis_password: cFSpbYUYGqcKG4Xj пароль для доступа к СУБД Redis
```

group_vars/redis_for_redash.yml

```
---
redis_password: cFSpbYUYGqcKG4Xj пароль для доступа к СУБД Redis
```

group_vars/redis_for_superset.yml

```
redis_password: aULfe6aqLEpp0Ycl пароль для доступа к СУБД Redis
redis_sentinel_enabled: true Redis запущен в режиме кластера
redis_replication_master_ip: 10.1.23.101 ip сервера на котором запущен
master для кластера
redis_sentinel_master_host: 10.1.23.101 ip сервера на котором запущен
master для кластера
```

group_vars/superset.yml

```
---
superset_database_postgresql_name: superset_db название базы данных для
Superset в СУБД Postgresql
superset_database_postgresql_user: superset_user описание пользователя
для Superset в СУБД Postgresql
superset_database_postgresql_password: PL5lyFceU62xVZl0
superset_database_postgresql_host: 10.1.23.101 ip сервера, на котором
развернута СУБД Postgresql для Superset
superset_celery_broker_sentinel_hosts: ip серверов на которых развернут
кластер Redis для Superset
- sentinel://:aULfe6aqLEpp0Ycl@10.1.23.101:26379
- sentinel://:aULfe6aqLEpp0Ycl@10.1.23.102:26379
- sentinel://:aULfe6aqLEpp0Ycl@10.1.23.103:26379
```

- создать в директории модуля директорию host_vars (пример содержимого директории ./examples/cluster/host_vars) и указать там актуальные ip для каждого сервера.

3. Запустить плейбук для установки модуля

```
ansible-playbook -i inventory.yml playbook.yml -D
```

Проверка

Для Superset

Для проверки работы модуля необходимо выполнить успешную авторизацию в Airflow, для этого:

1. Зайти по адресу <http://IP:8088>
2. Ввести логин и пароль для базового пользователя : admin/Awa1jJZCiSdQvDA

Для Redash

Для проверки работы модуля необходимо выполнить успешную авторизацию в Airflow, для этого:

1. Зайти по адресу <http://IP:80>
2. Выполнить процедуру регистрации для пользователя admin

10. Техническая поддержка

В случае возникновения ошибок при инсталляции или настройке ПО, а также при необходимости получения дополнительных консультаций следует связаться со службой технической поддержки, направив запрос по электронной почте dpp-support@itsumma.ru.