

# **AVELANA**

Руководство по базовой установке Продукта

## СОДЕРЖАНИЕ

1	Аннотация.....	3
2	История изменений.....	4
3	Состав дистрибутива.....	5
4	Подготовка к установке.....	6
4.1	Подготовка управляющего сервера.....	7
4.1.1	Проверка версии и установка Python.....	7
4.1.2	Установка Ansible.....	8
4.2	Подготовка управляемых серверов.....	8
4.2.1	Доступ к серверам по публичному ключу.....	8
4.2.2	Отключение необходимости ввода пароля для повышения прав.....	9
4.2.3	Подключение сетевого хранилища.....	10
4.3	Заполнение inventory файла.....	11
4.3.1	Возможные ошибки.....	14
4.4	Заполнение файла с переменными.....	15
5	Установка Продукта.....	19
5.1	Выполнение основного плейбука.....	19
5.1.1	Проверка результата.....	22
5.2	Выполнение плейбука iptables.....	23
5.3	Проверка служб после установки.....	23

# 1 АННОТАЦИЯ

Документ описывает процесс установки Продукта AVELANA.

Перед началом процесса установки необходимо запросить у менеджера логин и пароль для возможности скачивания дистрибутива и доступа к приватному registry.

Публичная версия руководства по базовой установке не содержит чувствительных данных, необходимых для полноценной процедура установки продукта, и предназначена для ознакомления с общим процессом развертывания.

В случае необходимости вы можете запросить полную версию инструкции по развертыванию, сформированную для вашей инфраструктуры и окружения у своего менеджера, либо на портале технической поддержки продукта.

Также полная версия данного руководства доступна в комплекте документов, предоставляемых вам в рамках взаимодействия с компанией: Контракт, Пресейл, Пакет для реестра отечественного ПО и т.д.

## 2 ИСТОРИЯ ИЗМЕНЕНИЙ

Версия документа: 1.0

Версия продукта:

Дата изменения документа: 11.06.2024

1. Перенос документа.
  2. Переход на новую структуру документа.
-

## 3 СОСТАВ ДИСТРИБУТИВА

Дистрибутив содержит следующие компоненты Продукта:

- **ansible** – набор playbook для автоматической подготовки серверов и развертывания Продукта;
- **chatwidget** – набор файлов для подключения клиентской части чата (КЧЧ) на сайте Заказчика;

### ВАЖНО

Перед началом установки убедитесь, что окружение соответствует [минимальным системным требованиям](#). Данные требования представлены в Сайзинге, либо на сайте в разделе [документация](#)

## 4 ПОДГОТОВКА К УСТАНОВКЕ

Перед началом установки Продукта необходимо:

- иметь учётную запись доступа к репозиториям, в которых размещены компоненты Продукта для установки (доступ запрашивается у менеджера проекта)
- убедиться в том, что вся инфраструктура для установки подготовлена согласно требованиям Продукта и выполненному сайзингу:
  - Управляющий сервер с Ansible (рабочее место специалиста внедрения);
  - Сервер(ы) аутентификации и авторизации;
  - Сервер(ы) внутренних сервисов;
  - Сервер(ы) публичных сервисов;
  - Сервер(ы) баз данных PostgreSQL;
  - Сервер(ы) шины данных RabbitMQ;
  - Сервер(ы) шины данных и кэша Продукта;
  - Сервер(ы) отчетов;
  - Сетевое хранилище;
  - Сетевое оборудование (балансировщик нагрузки);
  - Рабочие места пользователей (ПК операторов/супервизоров);
- выполнить подготовительные действия:
  - подготовить рабочее место специалиста внедрения - управляющий сервер Ansible;
  - подготовить управляемые серверы - серверы, где будут размещены компоненты Продукта согласно их ролям;
  - заполнить inventory;
  - заполнить переменные.

## 4.1 ПОДГОТОВКА УПРАВЛЯЮЩЕГО СЕРВЕРА

Управляющим сервером может быть любой сервер с ОС семейства linux (включая macos), а также ОС Windows с установленной [Windows Subsystem for Linux WSL](#), с сетевой доступностью к управляемым серверам, подготовленным для установки Продукта (подробнее см. [требования к рабочему месту специалиста внедрения](#)).

### 4.1.1 ПРОВЕРКА ВЕРСИИ И УСТАНОВКА PYTHON

Обязательным условием использования ansible является наличие python на управляющем и управляемых серверах.

#### Примечание

Плейбук адаптирован под: `ansible version >= 9.4.0 python version >= 3.6`

- Проверка установленного python на управляющем сервере:

Необходимо убедиться, что версия Python не ниже 3.6:

```
python3 --version
```

Если не установлен или установленная версия ниже 3.6, требуется выполнить установку:

```
sudo ____ -y install python3
```

Вместо пропуска указать пакетный менеджер системы, на которой происходит установка - apk/apt/yum/dnf.

- Проверка установленного pip менеджера:

```
python3 -m pip --version
```

Если не установлен:

```
sudo ____ -y install python3-pip
```

Вместо пропуска указать пакетный менеджер системы, на которой происходит установка - apk/apt/yum/dnf.

### 4.1.2 УСТАНОВКА ANSIBLE

Установка Продукта выполняется с помощью инструмента [Ansible](#), который выполняет несколько наборов (плейбуков) заранее настроенных автоматических шагов по установке Продукта.

- Установка ansible на управляющий сервер:

```
sudo pip3 install ansible
```

- Для проверки установки необходимо ввести:

```
ansible --version
```

- Если проверка установки не проходит, либо в момент установки возникают иные ошибки - необходимо обратиться к официальной документации [Ansible](#)

## 4.2 ПОДГОТОВКА УПРАВЛЯЕМЫХ СЕРВЕРОВ

Перед запуском процесса установки Продукта на управляемых серверах необходимо убедиться в том, что обеспечены:

- [требования к предустановленным пакетам и базовой настройке](#);
- [требования к организации доступа к серверам](#);
- [требования к сетевому хранилищу файлов](#)

Если какие-либо требования не обеспечены, но есть доступ по логину-паролю на сервера с правами `sudo`, остальные требования, исключая информацию об ntp-серверах, возможно обеспечить при подготовке к установке согласно подразделам ниже.

### 4.2.1 ДОСТУП К СЕРВЕРАМ ПО ПУБЛИЧНОМУ КЛЮЧУ

Для того, чтобы Ansible мог выполнить шаги плейбука на управляемых серверах необходимо обеспечить возможность доступа на управляемые серверы по публичному ключу.



1. Необходимо на управляющем сервере сгенерировать пару ssh ключей, выполнив команду:

```
ssh-keygen
```

Далее можно поменять или оставить название файлов ключей (если вам действительно нужна эта инструкция, настоятельно рекомендуется оставлять умолчания - просто четыре раза нажмите enter после ввода команды).

НЕ вводить парольную фразу!

2. Скопировать абсолютные пути до файлов ключей, они потребуются на этапе заполнения inventory файла и файла с переменными.

### Примечание

Чтобы узнать текущую директорию на управляющем сервере, введите в командной строке `pwd`. Посмотреть содержимое директории, например, чтобы узнать название ключа, введите `ls -lah`

## 4.2.2 ОТКЛЮЧЕНИЕ НЕОБХОДИМОСТИ ВВОДА ПАРОЛЯ ДЛЯ ПОВЫШЕНИЯ ПРАВ

Так как действия по установке на управляемых серверах осуществляются автоматически и требуют повышения привелегий, для корректной работы Ansible необходимо отключить подтверждение пароля при повышении привелегий.

Для этого на каждом сервере необходимо:

1. Выполнить вход с предоставленной учетной записью:

```
ssh username*@*ipadress  
#переключиться на суперюзера:  
sudo su  
#(ввести пароль)
```

2. Разрешить редактирование файла sudoers:

```
chmod +w /etc/sudoers
```

3. Добавить запись:

```
echo "%username% ALL=NOPASSWD: ALL" >> /etc/sudoers
```

заменив `%username%` на логин учетной записи входа.

4. Запретить дальнейшие изменения файла `sudoers`:

```
chmod -w /etc/sudoers
```

### 4.2.3 ПОДКЛЮЧЕНИЕ СЕТЕВОГО ХРАНИЛИЩА

Для работы Продукта необходимо заранее убедиться в том, что сетевое хранилище создано и подключено к серверам внешнего и внутреннего контуров [согласно требованиям](#).

Если сетевое хранилище есть, но не подключено, его необходимо подключить к серверам внутренних и публичных сервисов.

Для этого на каждом сервере внутренних и публичных сервисов необходимо:

1. Выполнить вход с предоставленной учётной записью: `ssh username@ipadress`.
2. Выполнить команду установки утилиты `cifs-utils`, обеспечивающей работы с файловым хранилищем по протоколу SMB:

```
sudo ____ -y install cifs-utils
```

Вместо пропуска указать пакетный менеджер системы, на которой происходит установка - `apk/apt/yum/dnf`.

3. Выполнить команду подключения файлового хранилища:

```
sudo echo -e 'username=%username%\npassword=%password%' >> /.smbclient
sudo echo '//%IP_or_FQDN%/%share_name%/ /mnt/sharedfiles cifs
user,rw,vers=2.0,uid=0,noforceuid,gid=0,noforcegid,file_mode=0666,dir_mode=0777,credentials=/.smbcli
ent 0 0' >> /etc/fstab
```

где:

- `%username%` - имя пользователя(сервисной УЗ), имеющего доступ к сетевому хранилища
- `%password%` - пароль пользователя(сервисной УЗ), имеющего доступ к сетевому хранилища
- `%IP_or_FQDN%`- IP адрес или полное имя сетевого хранилища

- `%share_name%` - имя сетевого хранилища

## 4.3 ЗАПОЛНЕНИЕ INVENTORY ФАЙЛА

1. Скачать архив `avelana_deploy_master.zip`, используя ссылку на скачивание и логин/пароль полученные от менеджера проекта.
2. Скопировать архив в любую директорию на управляющем сервере, откуда будет производиться установка Продукта, и разархивировать его, выполнив команду:

```
cd /opt/example_dir
unzip avelana_deploy_master.zip
ls -lah
```

На этом этапе будут отображены директории:

- **ansible** – набор `playbook` для автоматической подготовки серверов и развертывания Продукта;
  - **chatwidget** – набор файлов для подключения клиентской части чата (КЧЧ) на сайте Заказчика;
3. Перейти в директорию, содержащую набор `playbook`:

```
cd ansible/
```

4. Заполнить данные об окружении отредактировав файл `inventory`.
5. Находясь в директории распакованного архива, в файл `hosts.yml` внести данные об `ip` адресах серверов, согласно их предполагаемой роли в соответствии с сайзингом.

Если какие-то элементы будут установлены на один и тот же сервер, необходимо продублировать адреса.

6. Изменить значение переменных:
  - `ansible_user` - логин для входа на управляемые сервера;
  - `ansible_ssh_pass` - пароль для входа на управляемые сервера (если не был обеспечен вход по `ssh` ключу, в ином случае, эту строку нужно закомментировать, добавив знак `#` в начало строки);

- `ansible_ssh_private_key_file` - абсолютный путь до приватного ssh ключа, который будет использоваться для подключения.

Пример:

```
all:
  hosts:
    int_app:
      ansible_host: 10.10.10.1 # - сервер для внутренних приложений
    ext_app:
      ansible_host: 10.10.10.2 # - сервер для приложений dmz
    load_balancer:
      ansible_host: 10.10.10.3 # - сервер для размещения балансировщика
    coturn:
      ansible_host: 10.10.10.5 # - сервер для размещения turn сервера
    redis:
      ansible_host: 10.10.10.6 # - сервер для размещения redis
    rabbit:
      ansible_host: 10.10.10.6 # - сервер для размещения rabbitMQ
    pentaho:
      ansible_host: 10.10.10.7 # - сервер для размещения сервера отчетности pentaho
    keycloak:
      ansible_host: 10.10.10.8 # - сервер для размещения сервера аутентификации и
авторизации keycloak
    databases:
      ansible_host: 10.10.10.9 # - сервер для размещения сервера баз данных postgresql
  vars:
    ansible_user: developer # - укажите имя пользователя с правами sudo на целевых
серверах
    ansible_ssh_pass: SomePassword # - укажите пароль пользователя с правами sudo на
целевых серверах
    ansible_ssh_private_key_file: /home/user/.ssh/id_rsa # - укажите абсолютный путь до
файла приватного ключа
```

## Примечание

Если какие-либо компоненты не требуют установки - необходимо закомментировать строку, добавив знак `#` в начало строки.

Например:

```
# load_balancer:
# ansible_host: 10.10.10.3
```

### Важно!

Ниже строки `#Don't change anything else` вносить изменения **не нужно!** Служебные данные.

### Примечание

Комментарии в примере указаны для удобства ориентирования и в итоговом файле инвентори не используются.

После выполнения указанных выше действий необходимо команду проверки доступности управляемых серверов:

```
ansible -i hosts.yml all -m ping
```

Ожидаемый ответ:

```
● → ansible_requirements git:(test) ✖ ansible -i inventory/hosts.yml all -m ping
freeswitch | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
int_app | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
ext_app | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
rabbit | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
redis | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
keycloak | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
databases | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
pentaho | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
}
```

#### 4.3.1 ВОЗМОЖНЫЕ ОШИБКИ

- Ошибка интерпретатора:  
Необходимо проверить версию python.
- Сервер недоступен:

Проверить корректность заполнения inventory файла (верно ли указан IP).

Проверить доступность сервера, подключившись на него с управляющего сервера по ssh:

```
ssh username@ipaddress
```

Заменяв `username` и `ipaddress` на логин и ip адрес сервера соответственно.

## 4.4 ЗАПОЛНЕНИЕ ФАЙЛА С ПЕРЕМЕННЫМИ

Перед запуском плейбука необходимо отредактировать конфигурационный файл с переменными, указав специфичные для каждого окружения данные.

В файле `group_vars/all/vars.yml` необходимо внести значения переменных:

- Указать значение `timezone`, согласно своему часовому поясу, в формате `Area/Location` (Подробнее в [таблице](#) столбец `TZ Identifier`):

```
Enter your Time Zone (default is Europe/Moscow)
timezone: "Europe/Moscow"
```

По умолчанию (если значение не указано), будет установлен часовой пояс `Europe/Moscow`.

- Указать ip адреса или dns имена ntp серверов (при наличии):

```
Enter your NTP if you have some
first_ntp: "time1.google.com"
second_ntp: 'time2.google.com'
```

В противном случае по умолчанию будут использованы ntp google (`time1.google.com`, `time2.google.com`).

В итоговом конфигурационном файле всего 4 сервера, изменить можно таким образом первые два ntp сервера.

Два последних указаны `time3.google.com` и `time4.google.com` для надежности.

### Важно!

Если предварительно была произведена настройка ntp серверов, во избежание повторной настройки необходимо закомментировать в основном плейбуке `playbook.yml` строки `name: Set timezone`, поставив в начало каждой строки знак `#`:

```
- name: Copy chrony conf file
  template:
    src: templates/chrony.conf.j2
    dest: "{{ '/etc/chrony.conf' if ansible_facts['os_family'] == 'RedHat' else
```

```
'/etc/chrony/chrony.conf' }}"  
notify: Chrony restart
```

- Указать путь точки монтирования сетевого файлового хранилища, если он отличается от указанного:

```
# Укажите путь до общего файлового хранилища на управляемых серверах  
sharedpath: "/mnt/sharedfiles"
```

- Указать путь до файла публичного ключа, который будет использоваться для подключения к управляемым серверам (если был ранее создан), например `/home/user/.ssh/id_rsa.pub`:

```
# Укажите абсолютный путь до файла публичного ключа, который вы хотели бы использовать для  
подключения к управляемым серверам  
ssh_pub_key_file: ""
```

- Указать пароль для postgresql, полученный от Заказчика, либо сгенерированный самостоятельно и переданный Заказчику:

```
Enter password for postgres and pgagent  
pgpassword: "pgpassword"
```

- Указать пароли для Redis и RabbitMQ, полученные от Заказчика, либо сгенерированные самостоятельно и переданные Заказчику:

```
# Укажите пароль пользователя redis  
redispass: "redispass"  
# Укажите пароль пользователя rabbit  
rabbitpass: "rabbitpass"
```

- Указать имена виртуальных хостов RabbitMQ для внешнего и внутреннего контуров, если требуется использование значения, отличного от значения по умолчанию:

```
# Укажите имена виртуальных хостов rabbit для внешнего и внутреннего контуров  
local_vhost: "internal"  
dmz_vhost: "external"
```

- Указать `hub_name` и `hub_pass` логин пароль для доступа к приватному registry, полученные от менеджера:

```
Enter your login and password to hub.avelana.ru  
hub_name: ""  
hub_pass: ""
```



- Если планируется установка TURN сервера, указать `realmurl` - dns имя реалма для настройки coturn и `coturn_user`, `coturn_pass` - соответственно логин и пароль для coturn:

```
realmurl: ""
coturn_user: ""
coturn_pass: ""
```

- Требуется ввести относительные или абсолютные пути к файлам сертификатов и ключей ssl для внутреннего и внешних контуров, а так же для RabbitMQ. В случае использования одних и тех же файлов - пути указываются одинаковыми.

По умолчанию, рекомендуется положить файлы в директорию проекта.

### Важно!

Файл `cert.crt` должен содержать полную цепочку сертификатов и CA.

```
```yaml
# Укажите расположение файлов сертификата (full-chain) и ключа для web интерфейса RabbitMQ
rabbit_ssl_certificate_path: "cert.crt"
rabbit_ssl_certificate_key_path: "key.key"
# Путь расположения файлов сертификатов и ключей для внутреннего
local_ssl_certificate_path: "cert.crt"
local_ssl_certificate_key_path: "key.key"
# и внешнего контуров
dmz_ssl_certificate_path: "cert.crt"
dmz_avelana.russl_certificate_key_path: "key.key"
```
```

Указать пути до файлов сертификата и ключа ssl для управляемых серверов. Если используется самоподписанный сертификат для локального контура, требуется оставить путь `./untrusted_certs/название_файла`:

```
#Введите директорию расположения SSL на управляемой ноде, например, /etc/ssl/cert.crt
(оставьте untrusted_certs для самоподписанных)
CERTPATH: "/opt/product/untrusted_certs/cert.crt"
KEYPATH: "/opt/product/untrusted_certs/key.key"
```

### Важно!

Если используется самоподписанный сертификат для локального контура, перед запуском потребуется раскомментировать следующие строки плейбука `deploy/initial_local.yml`:

```

...yaml
# Раскомментировать строки ниже в случае использования самоподписанных сертификатов
- name: Sed if untrusted certs
  ansible.builtin.shell: sed -i "s|\#  -
\${LOCAL_TRUSTED_CERTS_PATH}\:\${CONTAINER_TRUSTED_CERTS_PATH}|  -
\${LOCAL_TRUSTED_CERTS_PATH}\:\${CONTAINER_TRUSTED_CERTS_PATH}|g"
- name: Create dir ./untrusted_certs if not exist
  ansible.builtin.file:
    path: /opt/product/untrusted_certs
    state: directory
...

```

- Указать логин и пароль для входа на web интерфейс сервера(ов) аутентификации:

```

# Укажите пользователя системы аутентификации
keycloak_admin: "admin"
# Укажите пароль пользователя системы аутентификации
keycloak_admin_password: "avelana"

```

- Заполнить поля согласно доменным именам окружения:

```

#Initial. Заполните DNS names (Все имена должны быть заполнены в нижнем регистре):

#Укажите FQDN домена локального контура (например, avelana.ru):
full_domain: ""
#Укажите DNS имя, включая домен, сервера модуля авторизации и аутентификации (KeyCloak):
auth_server_name: ""
#Укажите DNS имя, включая домен, сервера RABBITMQ:
rmq_dns_name: ""
#Укажите A-name сервера приложений или балансировщика нагрузки (без указания домена):
localapp_server_name: ""
#Укажите DNS имя сервера приложений DMZ-зоны (например, dmz.avelana.ru):
dmz_server_name: ""
#Укажите DNS имя сервера модуля единой очереди (например, pbx.avelana.ru). Если роль не
предусмотрена - оставить поле пустым:

# Если у вас есть балансировщик, раскомментируйте строку, удалив #, укажите dns имя или ip,
заскомментируйте строку ниже, добавив #:
#LBDNS: ""
LBDNS: '{{ localapp_server_name }}.{{ full_domain }}'

```

## 5 УСТАНОВКА ПРОДУКТА

Установка Продукта после подготовки заключается в:

- выполнении основного плейбука;
- выполнении плейбука iptables.

### 5.1 ВЫПОЛНЕНИЕ ОСНОВНОГО ПЛЕЙБУКА

Находясь в директории, куда распакован архив, необходимо выполнить команду:

```
ansible-playbook -i hosts.yml playbook.yml
```

В первую очередь, плейбук в автоматическом режиме выполнит следующие действия:

- Проверит входит ли использующиеся на окружении операционные системы в список поддерживаемых Продуктом;
- Запросит подтверждение о прочтении информации о поддержке на управляемых серверах ОС, и согласие продолжить проверки (ввести **yes** или **no**);
- Проверит соответствие серверов **минимальным системным требованиям** и выведет подтверждение соответствия (зеленым цветом) или ошибки (красным цветом);
- Проверит добавлен ли ssh ключ пользователю, и, если нет - добавит;
- Проверит установленную версию **python**, если ниже ожидаемой **3.6**, попыбует обновить (результат зависит от доступности более новой версии на репозиториях используемой ОС);
- Отключит фаерволы и **SELinux**, перезагрузит VM, если отключение **SELinux** произойдет;
- Проверит доступность сетевого хранилища на основании предоставленных данных на серверах внешнего и локального контура, проверит возможность записать файл;
- Запросит подтверждение, что после выданных проверками результатов, Вы готовы начать установку (ввести **yes** или **no**).

## Важно!

Перед выбором ответа на запрос продолжения внимательно ознакомиться с выведенной в консоль информацией при выполнении проверок.

Тестирование Продукта выполняется только на тех операционных системах, которые входят в список поддерживаемых, поэтому отсутствует гарантия корректной установки и работу Продукта на какой-либо операционной системе, не входящей в этот список. Продолжая установку на ОС не входящих в список поддерживаемых Продуктом, необходимо осознавать все возможные риски и проблемы разного уровня сложности в ходе установки и эксплуатации Продукта без помощи со стороны службы технической поддержки Продукта.

Установка Продукта на оборудование, не соответствующее минимальным аппаратным требованиям, может привести к некорректной работе или низкой скорости работы Продукта. Это может быть приемлемо при развертывании среды разработки или тестовой среды, но не рекомендуется для развертывания промышленных сред.

Не рекомендуется продолжать установку, если в процессе проверки возникли сложности с:

- + проверкой соответствия минимальным аппаратным требованиям;
- + проверкой соответствия списку поддерживаемых ОС;
- + добавлением ssh ключа;
- + отключением фаерволов;
- + проверкой доступности сетевого файлового хранилища;
- + установкой\обновлением `python`.

После проверок, при получении подтверждения продолжения установки, будет произведена подготовка окружения и установка Продукта.

Ожидаемые результаты:

В ходе выполнения плейбука будут отдаваться статусы по каждой выполняемой задаче. Нормальные ответы - `OK`, `Changed`, `Included`, `Skipping`, `Ignoring`:

```

+ ansible_requirements git:(test) x ansible-playbook -i inventory/hosts.yml playbook.yml
PLAY [Update and Upgrade] *****
TASK [Gathering Facts] *****
ok: [ext_app]
ok: [int_app]
ok: [rabbit]
ok: [redis]
ok: [freeswitch]
ok: [pentaho]
ok: [keycloak]
ok: [databases]

TASK [Update cache and upgrade] *****
included: /Users/jullmaksimova/ansible_requirements/tasks/update/yum.yml for int_app, ext_app, redis, rabbit, pentaho, keycloak, databases
included: /Users/jullmaksimova/ansible_requirements/tasks/update/apt.yml for freeswitch

TASK [Install epel] *****
changed: [int_app]
changed: [ext_app]
changed: [pentaho]
changed: [rabbit]
ok: [redis]
changed: [keycloak]
ok: [databases]

TASK [Update packages] *****
changed: [ext_app]
changed: [int_app]
changed: [pentaho]
changed: [redis]
ok: [rabbit]
changed: [keycloak]
ok: [databases]

```

```

TASK [Reboot if kernel updated] *****
skipping: [int_app]
skipping: [ext_app]
skipping: [redis]
skipping: [rabbit]
skipping: [pentaho]
skipping: [keycloak]
skipping: [databases]
changed: [freeswitch]

```

Информационные (предупреждающие) ответы - **Warning** - могут сообщать о каких-то незначительных событиях, например, в случае отсутствия какой-то роли, не будут использоваться часть плейев:

```

[WARNING]: Could not match supplied host pattern, ignoring: coturn

```

Ошибки, нуждающиеся в изучении и исправлении, обозначаются как **failed**:

```

PLAY RECAP *****
databases      : ok=48  changed=21  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
ext_app        : ok=28  changed=17  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
freeswitch     : ok=22  changed=13  unreachable=0  failed=1  skipped=1  rescued=0  ignored=0
int_app        : ok=28  changed=17  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
keycloak       : ok=34  changed=22  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
pentaho        : ok=33  changed=21  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
rabbit         : ok=32  changed=15  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
redis          : ok=36  changed=18  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0

```

В случае возникновения ошибок, описание события можно найти в истории выполнения задач (будет выделено красным шрифтом). Описание, обычно содержит необходимую информацию, достаточную для исправления ошибки. Если очевидные рекомендации по исправлению не сработали, требуется обратиться к сопровождающей установке организации.

### Важно!

Известные ошибки:

Ошибки **failed** при выполнении плей "Create extentions" будут всегда, так как при выполнении команды непустой stdout. Внимательно ознакомьтесь с содержимым сообщения: оно может включать в себя оповещение об успешном добавлении extentions или вывести ошибки добавления. Например, если такие extentions уже существуют.

Ошибки **failed** при выполнении плеев "Initial local" также содержат информацию из stdout. Внимательно ознакомьтесь с выводом, он должен завершаться строками, содержащими "Инициализация базы данных успешно выполнена"

Пример непустого вывода:

```
PL/pgSQL function inline_code_block line 7 at SQL statement: "psql:MakePostgresJob.sql:119: ERROR: relation \"pgagent.pga_job\" does not exist", LINE 1
: INSERT INTO pgagent.pga_job(" ^", "QUERY: INSERT INTO pgagent.pga_job(" jobclid, jobname, jobdesc, jobhostagent, jo
benabled", " ) VALUES (" , " 1::integer, 'J_LOAD_REVIEW_EXISTS':text, "':text, "':text, true" " ) RETURNING
jobid", "CONTEXT: PL/pgSQL function inline_code_block line 7 at SQL statement", "psql:MakePostgresJob.sql:171: ERROR: relation \"pgagent.pga_job\" does n
ot exist", "LINE 1: INSERT INTO pgagent.pga_job(" ^", "QUERY: INSERT INTO pgagent.pga_job(" jobclid, jobname, jobdesc
, jobhostagent, jobenabled", " ) VALUES (" , " 1::integer, 'J_CLEAR_LOG':text, "':text, "':text, true" " ) RE
TURNING jobid", "CONTEXT: PL/pgSQL function inline_code_block line 7 at SQL statement", "[19.08.2024 10:19:22.486] Apply sql-script MakePostgresJob comple
te", "\u001b(B\u001b[m[19.08.2024 10:19:22.486] Applying database changes completed", "\u001b(B\u001b[m Инициализация базы данных успешно выполнена", "\u00
1b(B\u001b[m]]
...ignoring
TASK [Up web-cm-managing] *****
```

## 5.1.1 ПРОВЕРКА РЕЗУЛЬТАТА

После выполнения всех шагов плейбука, Продукт установлен и требуется проверка перед дальнейшей настройкой и введением Продукта в эксплуатацию.

Аналогичным образом можно проверить корректность установки Keycloak, для этого указать fqdn-адрес сервера аутентификации без указания порта по протоколу **https** в браузере.

## 5.2 ВЫПОЛНЕНИЕ ПЛЕЙБУКА IPTABLES

Выполнение данного плейбука позволяет дополнительно обеспечить безопасность управляемых серверов, ограничив набор открытых портов на каждом сервере до перечня необходимых для работы Продукта.

Для запуска необходимо использовать команду:

```
ansible-playbook -i hosts.yml iptables.yml
```

Если после применения какие-то элементы перестали работать, можно откатить изменения на всех серверах, запустив плейбук `flush_iptables.yml`:

```
ansible-playbook -i hosts.yml flush_iptables.yml
```

Для отмены изменений на определенной группе серверов, внести изменения в первую строку плейбука `flush_iptables.yml`, указав перечень ролей серверов из inventory файла, для которых необходимо выполнить операцию, перед запуском, например:

```
- hosts: int_app, load_balancer
```

## 5.3 ПРОВЕРКА СЛУЖБ ПОСЛЕ УСТАНОВКИ

Для проверки запуска и работы службу после установки, на серверах внутренних и публичных сервисов необходимо выполнить команду:

```
docker ps | grep Res
```

и убедиться в том, что контейнеры с сервисами не находятся в состоянии постоянного перезапуска.