



ТЕРН ETL

Версия 1.0

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

Листов 16

2023

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ.....	3
1. Системные требования	4
2.1. Установка Docker.....	5
2.2. Установка Docker Compose.....	8
3. Установка и настройка ПО «Терн ETL	9
4. Устранение неисправностей	15

ГЛОССАРИЙ

№ п/п	Термин	Описание
1.	Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.
2.	Docker compose	Инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов.

ВАЖНО: Так будут выделены важные замечания.

1. Системные требования

Таблица 1. Системные требования

Процессоры	x64 представленная сборка оптимизирована под Intel, не менее 16 ядер 3 ГГц, (для оптимальной производительности предлагается использовать модель i9-13900KF)
Оперативная память (RAM)	Не менее 32 ГБ DDR5 (частота оперативной памяти не менее 4 ГГц)
Дисковая память	Не менее 500 ГБ SSD (зависит от объема данных участвующих в ETL процессах)
Сетевая карта	Не менее 100 Мбит/с
Операционная система	ОС Linux ALT Linux Workstation 10 или Astra Linux. (Представленная сборка оптимизирована под ALT Linux Workstation 10).

2. Формирование системного окружения для ПО «Tern ETL»

Перед началом работ по установке ПО «Tern ETL» необходимо установить и настроить следующие программные продукты:

- а) Docker;
- б) Docker compose;

ВАЖНО: Все действия необходимо выполнять в указанной последовательности.

Развертывание выполняется на машине под ОС Alt Linux Workstation. Также возможна установка на ОС Astra Linux с некоторыми изменениями, указанными в процессе установки.

2.1. Установка Docker.

Для развертывания проекта требуется установленный в Системе Docker.

Установка производится из официального репозитория Docker.

Далее будет указана последовательность действий для установки Docker на машину с ОС Linux.

Последовательность действий при установке Docker:

1. Первым делом обновляем список пакетов с помощью следующей команды:

```
sudo apt-get update.
```

На скриншоте указан вывод команды (может отличаться).

```
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64 release [4223B]
Получено: 2 http://ftp.altlinux.org p10/branch/x86_64-i586 release [1665B]
Получено: 3 http://ftp.altlinux.org p10/branch/noarch release [2844B]
Получено 8732B за 0s (44,0kB/s).
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic pkglist [23,3MB]
Получено: 2 http://ftp.altlinux.org p10/branch/x86_64/classic release [137B]
Получено: 3 http://ftp.altlinux.org p10/branch/x86_64-i586/classic pkglist [17,2M
B]
Получено: 4 http://ftp.altlinux.org p10/branch/x86_64-i586/classic release [142B]
Получено: 5 http://ftp.altlinux.org p10/branch/noarch/classic pkglist [6444kB]
Получено: 6 http://ftp.altlinux.org p10/branch/noarch/classic release [137B]
Получено 46,9MB за 11s (4202kB/s).
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
```

2. Устанавливаем Docker последней версии в Систему:

```
sudo apt-get install docker-ce
```

```
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Выбрано docker-engine для 'docker-ce'
Следующие дополнительные пакеты будут установлены:
  containerd docker-buildx docker-cli docker-engine docker-proxy runc tini
Следующие НОВЫЕ пакеты будут установлены:
  containerd docker-buildx docker-cli docker-engine docker-proxy runc tini
0 будет обновлено, 7 новых установлено, 0 пакетов будет удалено и 467 не будет обновлено.
Необходимо получить 80,4MB архивов.
После распаковки потребуется дополнительно 310MB дискового пространства.
Продолжить? [Y/n] Y
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic docker-buildx 0.10.2-alt1:p10+3
15926.2000.6.1@1679487170 [10,7MB]
Получено: 2 http://ftp.altlinux.org p10/branch/x86_64/classic docker-cli 23.0.1-alt1:p10+3159
26.1500.6.1@1679487052 [13,1MB]
Получено: 3 http://ftp.altlinux.org p10/branch/x86_64/classic docker-proxy 23.0.1-alt1:p10+31
5926.1400.6.1@1679486986 [674kB]
Получено: 4 http://ftp.altlinux.org p10/branch/x86_64/classic runc 1.1.5-alt1:p10+318394.100.
3.1@1681923544 [5775kB]
Получено: 5 http://ftp.altlinux.org p10/branch/x86_64/classic tini 0.19.0-alt1:sisyphus+27895
7.100.1.1@1626303581 [263kB]
Получено: 6 http://ftp.altlinux.org p10/branch/x86_64/classic containerd 1.6.17-alt1:p10+3159
26.1300.6.1@1679486906 [31,4MB]
```

```
Получено: 7 http://ftp.altlinux.org p10/branch/x86_64/classic/docker-engine-23.0.1-alt1:p10+3
15926.1400.6.1@1679486986 [18,5MB]
Получено 80,4МВ за 17с (4721кВ/с).
Совершаем изменения...
Подготовка... ##### [100%]
Обновление / установка...
1: runc-1.1.5-alt1 ##### [14%]
2: containerd-1.6.17-alt1 ##### [29%]
3: tini-0.19.0-alt1 ##### [43%]
4: docker-proxy-23.0.1-alt1 ##### [57%]
5: docker-buildx-0.10.2-alt1 ##### [71%]
6: docker-cli-23.0.1-alt1 ##### [86%]
7: docker-engine-23.0.1-alt1 ##### [100%]
Завершено.
```

На данном этапе при установке на ОС Astra Linux может возникнуть ошибка следующего вида: «**docker-ce**» не найден кандидат на установку. Для решения проблемы необходимо перейти в пункт №3 «Устранение проблем» раздела № 4 «Устранение неисправностей».

3. Проверяем корректность установки Docker с помощью следующей команды:

```
sudo docker -v
```

```
Docker version 23.0.1, build e92dd87
```

На данный момент актуальная версия ПО Docker 23.0.1.

4. В настоящий момент использование команды docker доступно только от имени пользователя sudo. Для удобства дальнейшей работы добавим текущего пользователя в группу docker:

```
sudo usermod -aG docker ${USER}
```

В случае успешного выполнения данная команда ничего не выведет.

ВАЖНО: После ввода данной команды необходимо перезайти в Систему от текущего пользователя, либо перезагрузить Систему полностью. Иначе изменения по внесению пользователя в группу не будут применены.

5. Проверяем запуск процесса Docker:

```
sudo systemctl status docker
```

```
o docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
 TriggeredBy: o docker.socket
   Docs: https://docs.docker.com
```

В нашем случае процесс не активен - Active: inactive (dead).

Запустим его вручную:

Терн ЕТЛ. Инструкция по установке. Версия 1.0
`sudo systemctl start docker`

Повторный запуск первой команды из данного пункта отобразит параметр Active с зеленым статусом Active.

ПО Docker было успешно установлено на Систему.

ВАЖНО: При перезапуске компьютера необходимо проделывать пункт №5 данного раздела.

2.2. Установка Docker Compose.

На данном этапе необходимо выполнить загрузку и установку ПО Docker Compose.

Последовательность действий при установке Docker Compose:

1. Следующая команда загружает версию 2.18.1 и сохраняет исполняемый файл в каталоге /usr/local/bin/docker-compose, вследствие чего ПО глобально будет доступно под именем docker compose:

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/download/v2.18.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %         Dload  Upload  Total  Spent    Left   Speed
  0     0     0     0     0     0      0     0  --:--:--  --:--:--  --:--:--    0
100 52.0M 100 52.0M    0     0 11.7M     0  0:00:04  0:00:04  --:--:-- 16.2M
```

2. Устанавливаем разрешение на запуск файла для всех пользователей:

```
sudo chmod +x /usr/local/bin/docker-compose
```

При успешном выполнении данная команда ничего не выведет.

3. Проверяем версию и корректность установки ПО Docker compose:

```
docker-compose version
```

```
Docker Compose version v2.18.1
```

ПО Docker Compose было успешно установлено на целевую систему.

ВАЖНО: Если вы используете другой дистрибутив Linux – обратитесь к официальной документации Docker для установки (<https://docs.docker.com/engine/install/>, <https://docs.docker.com/compose/install/>)

3. Установка и настройка ПО «Tern ETL»

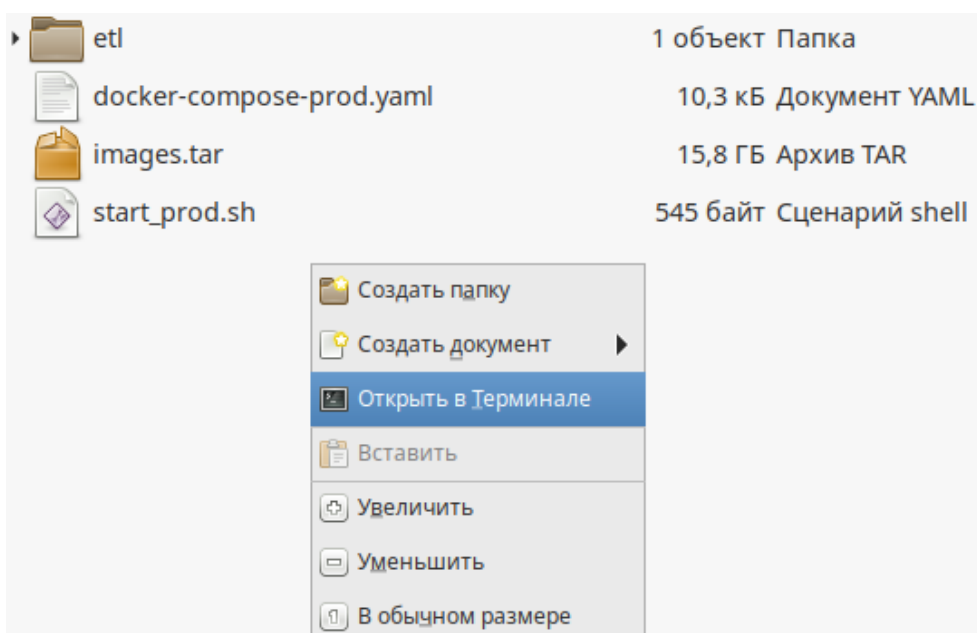
В состав дистрибутива входят следующие файлы:

1. etl – дирекория с файлом переменных среды.
2. Images.tar – архив с Docker образами.
3. Docker-compose.yaml – файл конфигурации для сборки docker образов.
4. Start_prod.sh – исполняемый скрипт для развертывания проекта.

Имя	Размер	Тип
etl	1 объект	Папка
docker-compose-prod.yaml	10,3 кБ	Документ YAML
images.tar	15,8 ГБ	Архив TAR
start_prod.sh	545 байт	Сценарий shell

Последовательность действий при развертывании проекта:

1. После загрузки комплекта поставки ПО на целевую рабочую машину необходимо открыть терминал в директории с файлами установки. Сделать это можно с помощью нажатия правой кнопкой мыши в директории с комплектом поставки и выбрав пункт «Открыть в терминале»:



2. В консоли открытого терминала нужно ввести команду запуска скрипта развертывания:

```
sudo ./start_prod.sh
```

Перед вами появится информационное сообщение о том, что начался процесс извлечения Docker образов из архива:

```
Извлекаем образы из архива
f5bb4f853c84: Loading layer 75.17MB/75.17MB
ff4c2aaaf1a0: Loading layer 44.26MB/44.26MB
e1c3d4a22b54: Loading layer 7.969MB/7.969MB
be5ca82501f6: Loading layer 71.79MB/71.79MB
2a4e41218384: Loading layer 71.79MB/71.79MB
4078e98a60cc: Loading layer 290.1MB/290.1MB
26a99b2246dd: Loading layer 1.536kB/1.536kB
06ea2431bae9: Loading layer 2.56kB/2.56kB
757df55f2417: Loading layer 2.674GB/2.674GB
27edacc198c0: Loading layer 192.9MB/192.9MB
661bb4c5a4f0: Loading layer 16.95MB/16.95MB
6fefef29be077: Loading layer 797.7MB/797.7MB
97102a611405: Loading layer 725.5kB/725.5kB
Loaded image: neural_network:latest
61581d479298: Loading layer 121.3MB/121.3MB
a72216901005: Loading layer 49.56MB/49.56MB
2f66f3254105: Loading layer 181.4MB/181.4MB
28218ecd8008: Loading layer 596.9MB/596.9MB
```

ВАЖНО: Дальнейших действий не требуется. Процесс может занять некоторое время. В случае, если процесс развертывания не начался, перейдите к пункту 2 в разделе «Устранение проблем»

3. Спустя некоторое время в терминале появятся сообщение о запуске контейнеров:

```
Запускаем контейнеры
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
[+] Building 0.0s (0/0)
[+] Running 18/18
✓ Network etl_services_full Created
✓ Volume "etl_files" Created
✓ Volume "etl_airflow_logs" Created
✓ Volume "etl_pg_custom_volume" Created
✓ Volume "etl_django_static" Created
✓ Volume "etl_django_logs" Created
✓ Container sf_shared-neural_network-1 Started
✓ Container sf_shared-neural_ocr-1 Started
✓ Container sf_shared-redis-1 Healthy
✓ Container sf_shared-expert_system-1 Started
✓ Container sf_shared-postgres-1 Started
✓ Container sf_shared-etl-1 Started
✓ Container sf_shared-airflow-init-1 Exited
✓ Container sf_shared-frontend-1 Started
✓ Container sf_shared-airflow-worker-1 Started
✓ Container sf_shared-airflow-webserver-1 Started
✓ Container sf_shared-airflow-scheduler-1 Started
✓ Container sf_shared-airflow-triggerer-1 Started
```

ВАЖНО: Никаких действий от пользователя не требуется.

4. Далее автоматически запускается процесс применения миграций к базе данных:

```
Применяем миграции
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
[+] Building 0.0s (0/0)
[+] Creating 1/0
✓ Container sf_shared-postgres-1 Running
[+] Building 0.0s (0/0)
```

5. На данном шаге автоматически создается супер-пользователь для входа в проект «Терн ETL»:

```
Создаем супер-пользователя
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
WARN[0000] The "AIRFLOW_UID" variable is not set. Defaulting to a blank string.
[+] Building 0.0s (0/0)
[+] Creating 1/0
✓ Container sf_shared-postgres-1 Running
[+] Building 0.0s (0/0)
```

6. В процессе выполнения скрипта будет создан тестовый пользователь с готовыми переносом из таблицы в таблицу.
7. Об успешном окончании всего процесса развертывания будет свидетельствовать информационное сообщение со следующей надписью:

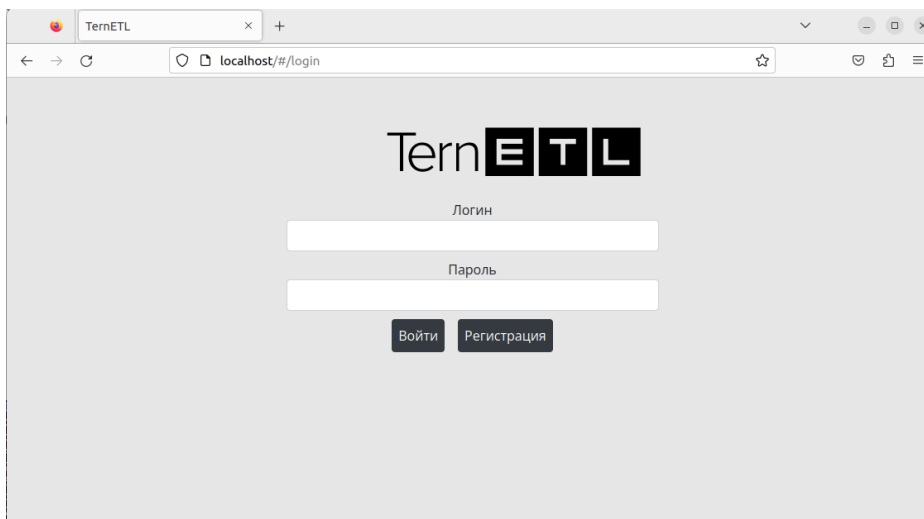
```
Развертывание проекта завершено!
```

8. Для первичной проверки запуска проекта можно ввести в консоли команду:

`docker ps`

Перед вами должен появиться список запущенных Docker контейнеров в количестве 13 штук. Статус каждого контейнера должен быть «Up».

9. Для входа в интерфейс «Tern ETL» необходимо открыть браузер и перейти по адресу localhost:



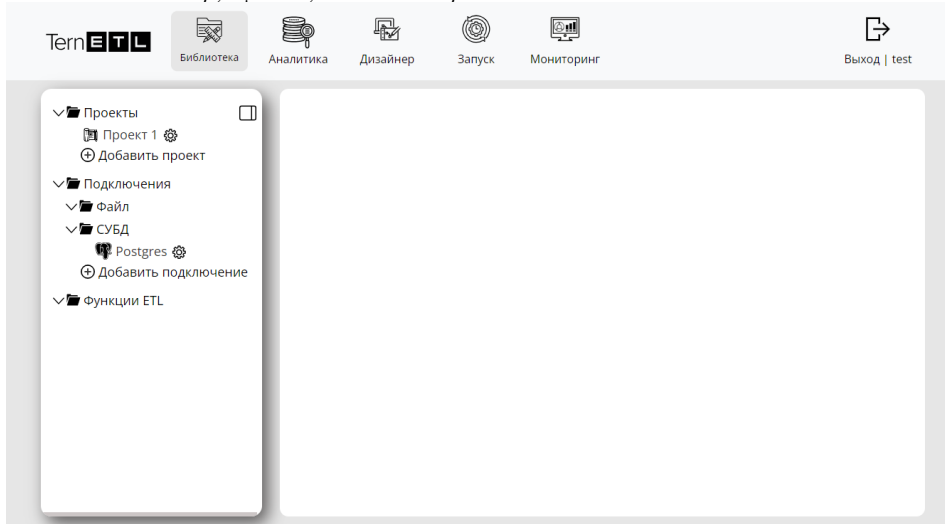
Вас будет приветствовать окно авторизации в интерфейс «Tern ETL». Во время разворачивания автоматически был создан тестовый аккаунт с переносом. С помощью него можно выполнить вход в Систему.

Данные для входа:

Логин: test

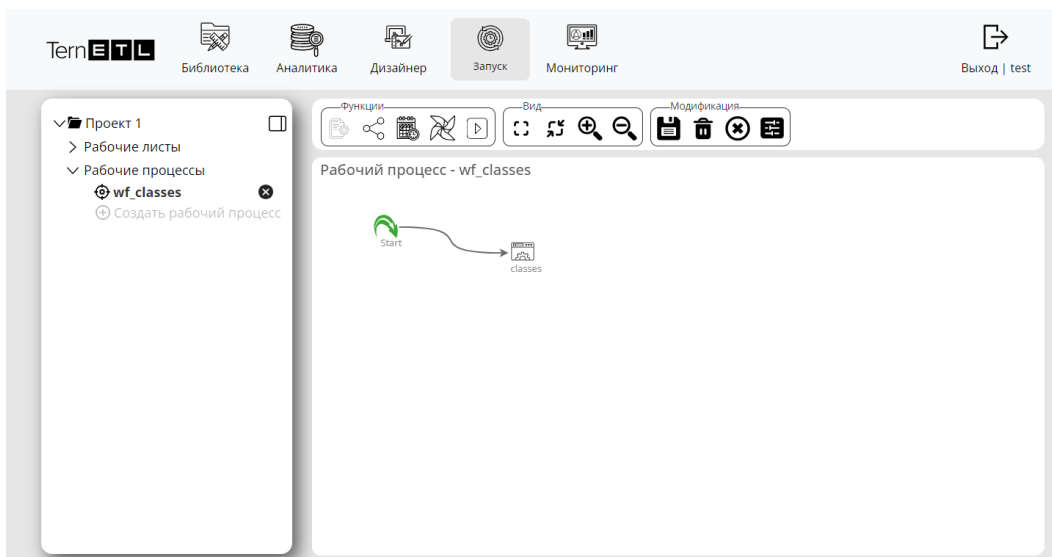
Пароль: 123

10. После авторизации будет выполнено автоматическое перенаправление в интерфейс системы «Tern ETL».

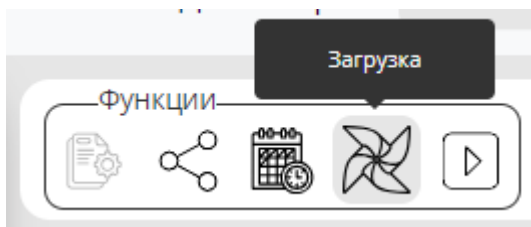


11. Сверху переходим по вкладке «Запуск».

Раскрывая «Проект 1» и «Рабочие процессы» нажимаем на единственный существующий рабочий процесс.

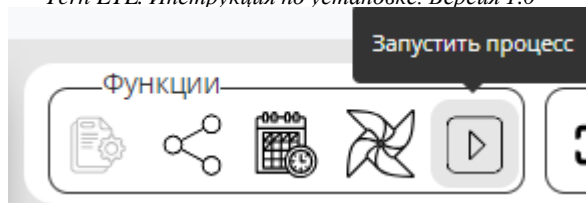


12. Нажимаем на кнопку «Загрузка» в блоке «Функции».



Следует подождать 10 секунд, пока рабочий процесс будет загружен в систему планирования.

Нажимаем кнопку «Запустить процесс»:



13. Переходим во вкладку «Мониторинг», открываем существующий «Проект 1», нажимаем на наш рабочий процесс и, раскрыв его, увидим успешно выполненные задачи.

▼ wf_classes2	2023-07-07 07:47:10.38	2023-07-07 07:47:51.79	success
⊗ create_parquet_folder	2023-07-07 07:47:35.28	2023-07-07 07:47:36.27	success
⊗ SQ_classes	2023-07-07 07:47:40.21	2023-07-07 07:47:43.26	success
⊗ classes_res	2023-07-07 07:47:46.14	2023-07-07 07:47:47.43	success
⊗ delete_parquet_folder	2023-07-07 07:47:49.27	2023-07-07 07:47:49.72	success
⊗ validate_dag	2023-07-07 07:47:50.69	2023-07-07 07:47:50.69	success

В случае возникновения проблем при развертывании следует обратиться к перечню неисправностей и путей их решения в разделе №4 «Устранение неисправностей».

ВАЖНО: В случае нештатного функционирования Системы для получения консультации или технической поддержки необходимо направить обращение по электронному адресу tech@tern.ru в службу технической поддержки.

4. Устранение неисправностей

В данном разделе описаны возможные неисправности и действия по их устранению.

1. Ошибка «<имя_пользователя> is not in the sudoers file. this incident will be reported».

Данная ошибка возникает в случае, когда у текущего пользователя недостаточно прав для выполнения команды с приставкой sudo.

Для решения нужно ввести команду su

Далее потребуется пароль суперпользователя.

Открываем файл sudoers на редактирование:

```
nano /etc/sudoers
```

Необходимо убрать решетку перед строчкой WHEEL_USERS
ALL=(ALL) ALL

```
##  
## Runas alias specification  
##  
##  
## User privilege specification  
##  
# root ALL=(ALL) ALL  
  
## Uncomment to allow members of group wheel to execute any command  
WHEEL_USERS ALL=(ALL) ALL  
  
## Same thing without a password  
# WHEEL_USERS ALL=(ALL) NOPASSWD: ALL  
  
## Uncomment to allow members of group sudo to execute any command  
# SUDO_USERS ALL=(ALL) ALL  
  
## Uncomment to allow any user to run sudo if they know the password  
## of the user they are running the command as (root by default).  
# Defaults targetpw # Ask for the password of the target user  
# ALL ALL=(ALL) ALL # WARNING: only use this together with 'Defaults targetpw'  
  
## Read drop-in files from /etc/sudoers.d  
@includedir /etc/sudoers.d
```

Для сохранения нажимаем сочетание клавиш CTRL + O.

Подтверждаем сохранение клавишей ENTER.

CTRL + X чтобы выйти из редактора.

Следующей командой выходим из режима суперпользователя:

```
Exit
```

Проблема устранена.

2. Команда `sudo ./start_prod.sh` не начала процесс развертывания. Для решения данной проблемы можно запустить выполнение скрипта в интерпретаторе shell. Делается этой следующей командой:

```
Sudo sh ./start_prod.sh
```

3. Ошибка «Для пакета «docker-ce» не найден кандидат на установку».

Данная проблема имеет место быть в случае установки Системы на ОС Astra Linux. Для решения необходимо последовательно ввести следующие команды:

```
sudo apt install apt-transport-https ca-certificates curl gnupg2  
software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo  
apt-key add -
```

```
sudo printf "deb [arch=amd64]
```

```
https://download.docker.com/linux/debian stretch stable \n" >  
/etc/apt/sources.list.d/docker.list
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

ВАЖНО: Для выполнения третьей команды потребуются права суперпользователя (su).

Данными действиями проблема будет решена, можно перейти к следующему пункту при развертывании.