

АРМ «С3000»

Установка образов Docker в ОС Linux



2025

Оглавление

Введение.....	3
Соглашения и условные обозначения	3
Сокращения	3
Системные требования	3
Установка ПО Docker.....	4
Astra Linux.....	4
Ubuntu Linux	4
Проверка работоспособности ПО Docker.....	6
Подготовка контейнера.....	6
Запуск контейнера.....	7
Перенаправление портов UDP	8
Работа с ключом защиты	8
Использование встроенных портов RS-232 и RS-485	9
Использование преобразователей USB в RS-232 и RS-485	9
Остановка и удаление АРМ «C3000».....	11
Восстановление и сброс паролей	11
Приложение А	12

Введение

Руководство предназначено для системных администраторов, выполняющих установку и начальную настройку АРМ «С3000» с использованием Docker (<https://www.docker.io>) в операционных системах семейства Linux.

Соглашения и условные обозначения

- **Полужирным** выделяются названия программных продуктов и аппаратных средств.
- *Курсив* применяется для обозначения технических терминов и в иных случаях для выделения частей текста.
- Моноширинный шрифт применяется для имен файлов, команд и их параметров, а также для примеров выполнения и вывода команд.
- В примерах выполнения команд, символ приглашения командной строки (command prompt) \$ говорит о том, что команда выполняется от имени непrivилегированного пользователя. # используется для команд, выполняемых суперпользователем (root, администратором системы).
- В соответствии с принятыми в документации для систем семейства **UNIX** соглашениями, имена команд записываются с указанием в скобках соответствующего номера раздела страниц руководства (man pages), например: lsusb(8), dmesg(1).
- **Примечание:** – краткие аннотации к основному тексту.

Сокращения

- ОС – операционная система
- ПО – программное обеспечение

Системные требования

Данная инструкция применима для следующих систем:

- Astra Linux Special Edition 1.7, 1.8 («Орел», «Воронеж», «Смоленск»);
- Ubuntu Linux 20.04 LTS («Focal Fossa»), 22.04 LTS («Jammy Jellyfish»), 24.04 LTS («Noble Numbat»);
- Debian 12 («Bookworm»).

Для каждой из ОС предоставляется свой образ **Docker**. Устанавливаемый образ **Docker** должен соответствовать ОС. Если для используемой ОС образ **Docker** отсутствует, следует использовать образ, предназначенный для **Debian**.

Образ **Docker** может поставляться в составе ZIP-архива (файл с расширением .zip). Перед использованием файл образа **Docker** следует извлечь из архива.

Установка ПО Docker

Astra Linux

Примечание: Для установки АРМ «С3000» в Astra Linux должен быть отключен «Мандатный контроль целостности».

Примечание: Для версий Astra Linux 1.7 и 1.8 используются разные образы Docker. Следует убедиться, что в названии Docker-образа версия Astra Linux соответствует версии ОС Astra Linux, на которую производится установка. Например, в названии образа для Astra Linux 1.8 содержится «astra-smolensk-1.8».

Установить пакет docker.io:

```
$ sudo apt install docker.io
```

Запустить службу **Docker**:

```
$ sudo systemctl start docker
```

Включить автоматический запуск службы:

```
$ sudo systemctl enable docker
```

При необходимости, разрешить работу с **Docker** непrivилегированным пользователям. Например, для пользователя *USER_NAME*:

```
$ sudo usermod -a -G docker USER_NAME
```

Для использования **Docker** в *непривилегированном (rootless)* режиме в **Astra Linux** (служба **Docker** запускается без прав суперпользователя, *root*):

- Установить пакет *rootless-helper-astra*:

```
$ sudo apt install rootless-helper-astra
```
- Запустить службу **Docker** от имени пользователя *USER_NAME*:

```
$ sudo systemctl start rootless-docker@USER_NAME
```
- Включить автоматический запуск службы от имени пользователя *USER_NAME*:

```
$ sudo systemctl enable rootless-docker@USER_NAME
```

Для запуска команд **Docker** в непривилегированном режиме следует использовать *rootlessenv(1)* из пакета *rootless-helper-astra*, например: *rootlessenv docker run ...*, *rootlessenv docker volume create ...*, и т.д.

Официальная документация: Установка и администрирование Docker в Astra Linux 1.7 (<https://wiki.astralinux.ru/pages/viewpage.action?pageId=158601444>).

Ubuntu Linux

Предпочтительно устанавливать **Docker** из официального репозитория.

Прежде всего, следует удалить пакеты **Docker**, установленные из репозиториев Ubuntu:

```
$ sudo apt purge      \
  docker-ce          \
  \
```

```
docker-ce-cli      \
containerd.io      \
docker-buildx-plugin \
docker-compose-plugin
```

Установить необходимые пакеты:

```
$ sudo apt install \
ca-certificates \
curl \
gnupg \
lsb-release
```

Загрузить и добавить GPG-ключ репозитория:

```
$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Добавить репозиторий:

```
$ echo "deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" \
| sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

После выполнения команды будет создан файл `/etc/apt/sources.list.d/docker.list` следующего вида (приведён пример для Ubuntu 22.04 LTS «Jammy Jellyfish» на машине с архитектурой `amd64`):

```
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian jammy stable
```

Обновить список пакетов:

```
$ sudo apt update
```

Установить пакеты **Docker**:

```
$ sudo apt install \
docker-ce      \
docker-ce-cli   \
containerd.io   \
docker-compose-plugin
```

В случае успешной установки, служба `docker` будет запущена и добавлена в автоматическую загрузку. Для проверки следует выполнить следующие команды и убедиться в соответствии их вывода приведённому ниже:

```
$ systemctl status docker
```

```
docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; ...)
Active: active (running) since Thu 2023-08-10 22:35:32 MSK; ...
TriggeredBy: docker.socket
...

```

```
$ systemctl list-unit-files | grep -i docker

docker.service      enabled      enabled
docker.socket       enabled      enabled
```

При необходимости, разрешить работу с **Docker** непrivилегированным пользователям. Например, для пользователя *USER_NAME*:

```
$ sudo usermod -a -G docker USER_NAME
```

Проверка работоспособности ПО Docker

Прежде чем приступить к работе с образами АРМ «С3000», рекомендуется произвести проверку **Docker** с использованием специально предназначенного для этой цели контейнера *hello-world*:

Убедиться в наличии подключения к сети Интернет.

Выполнить команду:

```
$ sudo docker run hello-world
```

В случае правильной установки и настройки **Docker**, вывод должен быть таким:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

...

Подготовка контейнера

Импортировать образ в локальный репозиторий **Docker**:

```
$ sudo docker load --input arm-s3000__REPOSITORY_VERSION.tar.gz
```

Здесь *REPOSITORY* в имени файла следует заменить на имя системы, для которой предназначен образ. *VERSION* здесь и далее в имени файла следует заменить на номер версии образа, с которым фактически происходит работа. Например, для образа Astra Linux 1.7 и версии 1.01.654.182, имя файла будет выглядеть как *arm-s3000_astra-smolensk-1.7_1.01.654.182.tar.gz*.

Примечание (для Astra Linux): если при выполнении команды возникла ошибка с текстом «contains vulnerabilities», как на рис. 1, то следует скачать новую версию АРМ «С3000» на сайте НВП «Болид». При повторном возникновении ошибки или другом сообщении следует обратиться в техподдержку.

The screenshot shows a terminal window titled '1 – Терминал' (Terminal 1). The window has a red border. The terminal menu bar includes 'Файл', 'Правка', 'Вид', 'Закладки', 'Модули', 'Настройка', and 'Справка'. Below the menu are icons for copy, paste, cut, and search. The terminal output is as follows:

```
astral@astra:~$ docker load --input arm-s3000-astra-smolens
k_1.7-1.01.697.218.tar.xz
c16fcc329201: Loading layer 856.6MB/856.6MB
directory '/var/lib/docker/overlay2/2c55986070f291d79ab875d
daf58de561a36afc820d3fa7936c939fe3b24881d/diff' contains vu
lnerabilities! Count of vulnerabilities - 128. Image: 2c559
86070f291d79ab875ddaf58de561a36afc820d3fa7936c939fe3b24881d
. Vulnerabilities:
low - oval:astra:def:4293543816087711297194590295619 [BDU:2
023-07117], [CVE-2023-4016], Astra Linux - уязвимость в pr
ocps;
low - oval:astra:def:3820041427189976741948718143043 [BDU:2
023-07904], [CVE-2023-5870], Astra Linux - уязвимость в ro
stgresql-11;
low - oval:astra:def:1079419409311606847105405416724035 [BD
```

Рисунок 1. Ошибка загрузки устаревшего контейнера

Создать том **Docker** для хранения данных контейнера. `arm-s3000-volume` в команде – произвольное имя тома (должно быть уникальным в пределах локальной ОС):

```
$ sudo docker volume create arm-s3000-volume
```

Запуск контейнера

Запуск производится командой:

```
$ sudo docker run \
    --name arm-s3000 \
    --volume arm-s3000-volume:/persist \
    --restart=always \
    --publish 20080:80 \
    --publish 20043:443 \
    arm-s3000 __REPOSITORY:VERSION
```

Команде `docker run` передаются следующие параметры:

- `--name arm-s3000` Произвольное имя контейнера для использования в командах `docker(1)`.
- `--volume arm-s3000-volume:/persist` Имя тома, созданного командой `docker volume create`. `/persist` — папка в контейнере, где будет смонтирован том.
- `--restart=always` Автоматический перезапуск контейнера в случае завершения его работы.

- `--publish 20080:80 --publish 20043:443` Перенаправление портов TCP. Соединение с портом, указанным до `:`, на локальной системе будет перенаправлено на порт, указанный после `:`, в контейнере.
- `arm-s3000__REPOSITORY:VERSION` Имя образа Docker. Про VERSION см. раздел «Подготовка контейнера» выше.

После успешного запуска контейнера соединение с системой АРМ «С3000» возможно на всех сетевых интерфейсах и заданных портах, например: <http://127.0.0.1:20080> или <https://127.0.0.1:20043>.

Перенаправление портов UDP

Перенаправление портов UDP может потребоваться:

- При подключении приборов к АРМ «С3000» через устройство **C2000-Ethernet** в том случае, если в настройках **C2000-Ethernet** отключен параметр «Использовать один UDP-порт на чтение и запись».
- В случае возникновения проблем при использовании NAT.

Для этого необходимо передать команде `docker run` параметр вида `--publish 20500:60500/udp`, где до `:` указан порт на локальной системе, а после `:` – порт в контейнере.

Номера портов на локальной системе могут принимать значения от 2048 до 65535.

Работа с ключом защиты

При запуске системы АРМ «С3000» происходит поиск ключа защиты, подключенного к порту USB. Без ключа АРМ «С3000» будет работать в демонстрационном режиме. Для того чтобы ОС, запущенная в контейнере, смогла найти это устройство, команде `docker run` необходимо передать параметр `--device`, например:

```
--device=/dev/bus/usb/002/003
```

Здесь `/dev/bus/usb/002/003` указывает путь к файлу устройства ключа защиты. Определить этот путь позволяет утилиты `lsusb(8)`:

```
$ lsusb -v
```

```
Bus 002 Device 003: ID 04d8:053f Microchip Technology, Inc.
```

```
Device Descriptor:
```

```
  bLength 18
```

```
  ...
```

```
  iProduct 2 Bolid security dongle
```

```
  ...
```

В выводе команды `lsusb -v` нужно найти запись со значением поля `iProduct`, равным `«Bolid security dongle»`. Поля `Bus` и `Device` этой записи позволяют сформировать путь к файлу устройства. Например, для `Bus 002` и `Device 003` путь будет таким:

```
/dev/bus/usb/002/003
```

При извлечении ключа защиты из ПК или сервера и повторном подключении ключ не будет найден. Следует выполнить повторный поиск с помощью `lsub`.

Для того чтобы поиск ключа защиты происходил автоматически, вместо параметра `device` следует использовать следующий фрагмент:

```
--device-cgroup-rule "c 189:* rwm" -v /dev/bus/usb:/dev/bus/usb
```

Примечание: допустимо использование этой конструкции только при условии запуска только одного контейнера АРМ «С3000» одновременно!

Использование встроенных портов RS-232 и RS-485

Для работы с физически встроенными в компьютер портами RS-232 и RS-485 команде `docker run` необходимо передать параметр `--device`, указывающий путь к файлу используемого устройства.

Для добавления такого порта к системе АРМ «С3000» следует выполнить следующие шаги:

1. При запуске контейнера для каждого используемого порта добавить параметр `--device=/dev/DEVICE`, где `DEVICE` – имя порта RS-232/485. Чтобы узнать верное название порта, следует обратиться к документации поставщика оборудования.
2. Зайти на страницу «Конфигурирование структуры объекта → Приборы → Порты RS» и добавить каждый из них: в поле «Устройство» следует указывать переданные в `--device` значения
3. Дальше можно создавать Линии с их использованием (см. раздел «Добавление линии» Руководства по эксплуатации).

Использование преобразователей USB в RS-232 и RS-485

Для работы с преобразователями USB в RS-232 и RS-485 команде `docker run` необходимо передать параметр `--device`, указывающий путь к файлу используемого устройства. Файл самого устройства назначается по порядку нахождения их на USB-шине, и при перезапуске компьютера имя может быть изменено операционной системой.

Все преобразователи USB-RS производства **НВП «Болид»** при подключении имеют пути `/dev/ttyUSB{номер}`.

Для определения необходимого номера следует выполнить следующие действия:

1. В Терминале ввести команду:

```
$ ls -al /dev/serial/by-id
```

2. Найти префикс своего устройства (см. Приложение А) в появившемся списке, в конце строки с нужным префиксом будет отображаться требуемое устройство `ttyUSB` (см. рис. 2).

Примечание: если при подключении преобразователя не появилось устройство, т.е. нет нового /dev/ttyUSB{номер}, обратитесь в техподдержку для получения инструкций по поддержке.

Преобразователи вносят задержку в передачу данных, и ее необходимо компенсировать увеличением параметров «Таймаут ответа на команду» и «Таймаут запроса новых событий» используемой Линии согласно приложению А (см. раздел «Параметры линий» Руководства по эксплуатации).

```
astra1@astra:~$ ls -al /dev/serial/by-id/  
итого 0  
drwxr-xr-x 2 root root 60 фев 17 13:22 .  
drwxr-xr-x 4 root root 80 фев 17 13:22 ..  
lrwxrwxrwx 1 root root 13 фев 17 13:22 usb-Exar_Corp._XR21B  
1411_N1574158351-if00-port0 -> ../../ttyUSB0
```

Рисунок 2. Определение номера ttyUSB

Для получения пары VID:PID воспользуйтесь командой lsusb (рис. 3):

```
astra1@astra:~$ lsusb  
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root  
hub  
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root  
hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root  
hub  
Bus 002 Device 002: ID 0403:6001 Future Technology Devices  
International, Ltd FT232 Serial (UART) IC  
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root  
hub
```

Рисунок 3. Получение VID:PID с помощью lsusb

Остановка и удаление АРМ «С3000»

Остановить контейнер:

```
# docker stop arm-s3000
```

Удалить контейнер:

```
# docker rm arm-s3000
```

Удалить том **Docker** (`arm-s3000-volume` – имя тома):

```
# docker volume rm arm-s3000-volume
```

Удалить образ **Docker**:

```
# docker image rm arm-s3000__REPOSITORY:VERSION
```

Восстановление и сброс паролей

В случае утери пароля для встроенной учетной записи, а также при необходимости изменения паролей других пользователей без использования web-интерфейса, используется команда `password-reset`, запускаемая в контейнере.

Вызванная без параметров, она восстанавливает пароль по умолчанию (`armS3000`) для пользователя `admin`. При вызове с ключом `-u` команда меняет пароль для пользователя с указанным именем учетной записи.

Сначала нужно остановить контейнер (`arm-s3000` – имя контейнера):

```
# docker stop arm-s3000
```

Восстановить пароль пользователя `admin`:

```
# docker run \
  --volume arm-s3000-volume:/persist \
  --rm \
  arm-s3000__REPOSITORY:VERSION \
  password-reset
```

Описание параметров команды `docker run` приведены в разделе «Запуск контейнера».

Задать новый пароль `new_password` для пользователя `user_name`:

```
# docker run \
  --volume arm-s3000-volume:/persist \
  --rm \
  arm-s3000__REPOSITORY:VERSION \
  password-reset -u "user_name" "new_password"
```

Примечание:

Если пользователь с именем, переданным команде, не существует, он будет создан; роль новой учетной записи – `service`.

При последующем запуске контейнера вступят в действие новые пароли.

Приложение А

Таблица А.1. Характеристика преобразователей USB в RS-232 и RS-485 НВП «Болид»

Название	VID:PID	Наличие уникального номера	Префикс в папке /dev/serial/by-id/	Увеличить на X мс				
				Орион				
				9600	9600	19200	57600	
C2000-USB или USB-RS485 (RS-485)	1a86:7523	Нет	недоступно	32	32	16	0	0
	10c4:ea60	Да	usb-Silicon_Labs_CP2104_USB_to_UART_Bridge_Controller	72	128	40	5	0
	04e2:1411	Да	usb-Exar_Corp._XR21B14	Не поддерживается в Linux!				
	0403:6001	Да	usb-FTDI_USB-RS485	0	0	0	0	0
USB-RS232 (RS-232)	1a86:7523	Нет	usb-1a86_USB_Serial	32	32	16	0	0
	10c4:ea60	Да	недоступно	72	128	40	5	0
	04e2:1411	Да	usb-Exar_Corp._XR21B14	0	0	0	0	0
USB-RS	10c4:ea60	Да	недоступно	72	128	40	5	0