

Как развернуть Astra Linux в Docker: подробный гайд

Хотите попробовать Astra Linux без установки на жёсткий диск?

Docker позволяет запустить Astra Linux в изолированном контейнере — как легковесную песочницу.

Мы описали, как это сделать на Ubuntu с помощью командной строки.

- [Шаг 1. Установить Docker](#)
- [Шаг 2. Запустить Docker](#)
- [Шаг 3. Проверить установку](#)
- [Шаг 4. Выбрать образ Astra Linux](#)
- [Шаг 5. Загрузить выбранный образ](#)
- [Шаг 6. Запустить контейнер с Astra Linux](#)
- [Что делать после запуска](#)
- [Основные команды Docker](#)
- [Дополнительно](#)

Шаг 1. Установить Docker

Выполните команду.

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

Шаг 2. Запустить Docker

Запустите Docker вручную.

```
$ sudo systemctl start docker
```

Если требуется, чтобы Docker автоматически запускался при старте Ubuntu, включите автозапуск.

```
$ sudo systemctl enable docker
```

Шаг 3. Проверить установку

Проверьте установленную версию Docker.

```
$ docker --version
```

При корректной установке Docker отобразится его версия.

Пример вывода:

```
Docker version 24.0.5, build 1234567
```

Также можно проверить Docker, запустив в нем тестовый образ `hello-world`.

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

Если Docker успешно развернёт тестовый образ, отобразится приветственное сообщение.

Пример вывода:

```
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
...
```

Шаг 4. Выбрать образ Astra Linux

Выберите для загрузки необходимый `образ` Astra Linux — готовый шаблон системы, по которому Docker создаст контейнер с ней.

Официальные базовые образы Astra Linux доступны в каталоге <https://registry.astralinux.ru/browse/library/> в разделе `astra`.

В названиях образов используется шаблон: `astra/ubi<major-version>[-dev]:<tag>`.

Параметры шаблона:

- `<major-version>` — основная версия Astra Linux (например, `18` для версии Astra Linux 1.8).
- `[-dev]` — (опционально) указывает на предустановленную среду выполнения (например, `python311` для Python 3.11).
- `<tag>` — тег образа (`latest` для последней версии).

Примеры:

- `astra/ubi18:1.8.0`: образ, основанный на Astra Linux 1.8.
 - `astra/ubi17-openjdk110:latest`: образ на базе Astra Linux 1.7 с предустановленной средой для выполнения OpenJDK версии 11.0, помеченный как последний доступный.
-

Шаг 5. Загрузить выбранный образ

Допустим, выбран образ самой новой версии Astra Linux 1.8 без предустановленных сред выполнения кода.

Для его загрузки в Docker выполните следующую команду.

```
$ docker pull registry.astralinux.ru/library/astra/ubi18:latest
```

Пример вывода:

```
latest: Pulling from library/astra/ubi18
1ac5ef385c4d: Pulling fs layer
e8d586ab4014: Pulling fs layer
2d96c61df80c: Pulling fs layer
...
Digest: sha256:d0d2f68b1546c29dbf3b30f053292d5e7df1d8b6d6db320af865c4205c6cbd0b
Status: Downloaded newer image for
registry.astralinux.ru/library/astra/ubi18:latest
registry.astralinux.ru/library/astra/ubi18:latest
```

Шаг 6. Запустить контейнер с Astra Linux

Воспользуйтесь командой, чтобы запустить Astra Linux и командную строку в нем:

```
$ docker run -it --name astra_container
registry.astralinux.ru/library/astra/ubi18:latest /bin/bash
```

В случае успешного запуска вы окажетесь в командной строке контейнера и увидите мигающий курсор.

Пример вывода:

```
[root@f1d3c2a4b7e8 /]
```

Готово! Образ Astra Linux успешно запущен в Docker. Теперь вы можете работать в его командной строке.

Что делать после запуска

Проверьте версию Astra Linux

```
$ cat /etc/os-release
```

Пример вывода:

```
PRETTY_NAME="Astra Linux"  
NAME="Astra Linux"  
ID=astra  
ID_LIKE=debian  
ANSI_COLOR="1;31"  
HOME_URL="https://astralinux.ru"  
SUPPORT_URL="https://astralinux.ru/support"  
LOGO=astra  
VERSION_ID=1.8_x86-64  
VERSION_CODENAME=1.8_x86-64
```

Обновите Astra Linux внутри контейнера

```
$ apt-get dist-upgrade
```

Пример вывода:

```
Reading package lists... Done  
Building dependency tree... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Установите пакеты, необходимые для работы

Например, для установки текстового редактора nano:

```
$ apt install -y nano
```

Пример вывода (сокращенный):

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 850 kB of archives.
After this operation, 3,678 kB of additional disk space will be used.
Get:1 http://mirror.astra... nano_5.4-2_amd64.deb [850 kB]
Fetched 850 kB in 1s (1123 kB/s)
Selecting previously unselected package nano.
Preparing to unpack .../nano_5.4-2_amd64.deb ...
Unpacking nano (5.4-2) ...
Setting up nano (5.4-2) ...
Processing triggers for man-db (2.9.4-2) ...
```

Создайте пользователя

Чтобы создать пользователя **myuser**, воспользуйтесь командой:

```
$ adduser myuser
```

Пример вывода:

```
Adding user `myuser' ...
Adding new group `myuser' (1001) ...
Adding new user `myuser' (1001) with group `myuser' ...
Creating home directory `/home/myuser' ...
Copying files from `/etc/skel' ...

New password:
Retype new password:
passwd: password updated successfully
Changing the user information for myuser
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
```

Чтобы предоставить пользователю **myuser** права администратора, добавьте его в группу **sudo** (пользователь сможет запускать команды с **sudo**).

```
$ usermod -aG sudo myuser
```

Команда не выводит сообщений, если выполнена успешно.

Выйдите из контейнера с Astra Linux

```
$ exit
```

После выхода контейнер с Astra Linux остановится, и вы вернётесь в основную систему Ubuntu.

Заново подключитесь к контейнеру и его командной строке

```
$ docker start -ai astra_container
```

После выполнения команды вы снова окажетесь в командной строке контейнера и увидите мигающий курсор.

Основные команды Docker

Обратите внимание, что команды Docker не работают в командной строке Astra Linux. Перед их использованием важно выйти из контейнера (вернуться в Ubuntu) с помощью команды **exit**.

Посмотрите запущенные контейнеры

```
$ docker ps
```

В выводе отобразится список запущенных контейнеров. Astra Linux среди них не будет, так как он не работает в фоновом режиме и отключается сразу после выхода из него.

Удалите контейнер с Astra Linux

```
$ docker rm astra_container
```

Если контейнер успешно удален отобразится подтверждение того, что удалён контейнер с именем **astra_container**.

```
astra_container
```

```
$ docker rmi registry.astralinux.ru/library/astra/astra/ubi18:latest
```

[illegible]

- [Официальная документация Docker](#)
- [Документация образов Astra Linux](#)