



PICODATA

*Распределенный сервер приложений со встроенной  
распределенной базой данных*

## **Руководство по установке**

# 1. Оглавление

О данном руководстве.....	3
1. Системные требования.....	4
1.1. Спецификация основных ресурсов.....	4
1.2. Методика расчета ресурсов для эксплуатации распределенной системы.....	4
1.3. Требования к операционной системе.....	4
2. Установка из исходного кода.....	5
2.1. Предварительные действия для RHEL/CentOS 8.....	5
2.2. Предварительные действия для Ubuntu 22.04.....	5
2.3. Предварительные действия для Alt Workstation p10.....	6
2.4. Получение исходного кода и сборка.....	6
3. Установка готовых пакетов.....	7
3.1. Установка в RHEL/CentOS.....	7
3.2. Установка в РЕД ОС/RED OS 7.3 «Муром».....	8
3.3. Установка в Astra Linux 1.7 SE.....	8
3.4. Установка в Debian.....	9
3.5. Установка в Ubuntu 20.04 и 22.04.....	9
3.6. Установка в Alt Linux.....	9
4. Проверка установки.....	11

## **О данном руководстве**

Документ «Руководство по установке» содержит сведения, которые должны помочь пользователям и системным администраторам установить программное обеспечение Picodata для последующего ознакомления, запуска и эксплуатации.

Информация о работе уже установленного программного обеспечения приведена в отдельном документе «Руководство по эксплуатации».

Информация о внутреннем устройстве распределенной системы (кластера) приведена в отдельном документе «Руководство администратора».

В текущем документе содержится описание последовательности действий для сборки ПО Picodata из исходного кода, а также для установки готовых, заранее собранных пакетов Picodata в поддерживаемых операционных системах.

Сведения в данном документе относятся к текущей публично доступной версии ПО Picodata 22.11.0, вышедшей в ноябре 2022 г. Информация в этом руководстве будет обновляться для наиболее полного соответствия фактической функциональности ПО Picodata на момент публикации.

# **1. Системные требования**

## **1.1. Спецификация основных ресурсов**

**Системные требования** к программному обеспечению Picodata являются спецификацией основных ресурсных потребностей данного ПО. В частности, для эксплуатации ПО требуется наличие Intel-совместимого ПК со следующими характеристиками:

- ЦП Intel/AMD с частотой не менее 2 ГГц и минимум двумя ядрами;
- 8 ГБ ОЗУ или более;
- основной накопитель емкостью не менее 30 ГБ;

## **1.2. Методика расчета ресурсов для эксплуатации распределенной системы**

Методика расчета требуемых аппаратных ресурсов на основании целевых показателей нагрузки состоит в планировании числа узлов для распределенной системы (кластера), а также выделению требуемых вычислительных ресурсов на каждый элемент/узел кластера.

При развертывании распределенной системы на нескольких отдельных серверах, следует учитывать следующие требования к каждому серверу/узлу такой системы:

- совместимость с архитектурой x86\_64
- одно ядро CPU на один экземпляр/инстанс Сервиса
- сетевая связность между экземплярами

## **1.3. Требования к операционной системе**

ОС на основе Linux x86\_64 с glibc 2.17 или новее.

## 2. Установка из исходного кода

Приведенные ниже инструкции относятся к сборке исходного кода в окружении ОС Linux x86\_64. Рекомендуется использовать современный дистрибутив Linux с актуальными версиями инструментов разработчика. Под этими инструментами подразумевается современный набор пакетов для сборки приложений, написанных на языках программирования C/C++ и Rust.

Убедитесь, что в вашей системе установлены следующие компоненты:

- Rust и Cargo 1.59 или новее
- Cmake 3.16 или новее
- gcc, g++
- libstdc++-static

*Примечание. Названия соответствующих пакетов могут отличаться в зависимости от используемого дистрибутива Linux.*

Далее приведены инструкции для некоторых популярных дистрибутивов Linux.

### 2.1. Предварительные действия для RHEL/CentOS 8

Используйте следующие команды для установки необходимых зависимостей для сборки Picodata из исходного кода. Обратите внимание, что потребуется установить актуальные версии Rust и Cargo, используя официальные инструкции с сайта [rustup.rs](https://rustup.rs):

```
sudo dnf config-manager --set-enabled powertools
sudo dnf install -y gcc gcc-c++ make cmake git patch libstdc++-static
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
source "$HOME/.cargo/env"
```

### 2.2. Предварительные действия для Ubuntu 22.04

Используйте следующие команды для установки необходимых зависимостей для сборки Picodata из исходного кода. Обратите внимание, что Ubuntu 22.04 предоставляет актуальные версии Rust и Cargo в стандартном репозитории, поэтому для их установки следует использовать `apt-get`:

```
sudo apt-get install build-essential cargo git cmake -y
```

## 2.3. Предварительные действия для Alt Workstation p10

Используйте следующие команды для установки необходимых зависимостей для сборки Picodata из исходного кода. Обратите внимание, что потребуется установить актуальные версии Rust и Cargo, используя официальные инструкции с сайта [rustup.rs](https://rustup.rs):

```
su -  
  
apt-get install gcc gcc-c++ cmake git patch libstdc++10-devel-  
static libgomp10-devel-static -y && exit  
  
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh  
  
source "$HOME/.cargo/env"
```

## 2.4. Получение исходного кода и сборка

Выполните в терминале следующие команды:

```
git clone https://git.picodata.io/picodata/picodata/picodata.git  
cd picodata  
git submodule update --init --recursive
```

Сборка проекта производится следующей командой:

```
cargo build
```

Таким образом будет собрана debug-версия. Если вам нужна релизная версия, используйте следующую команду при сборке:

```
cargo build --release
```

Бинарные файлы приложения Picodata появятся в директории `target/debug` или `target/release`.

### 3. Установка готовых пакетов

Picodata.io поддерживает установку программного обеспечения «Пикодата», а также нашего форка СУБД Tarantool на следующих операционных системах:

- RHEL 7-8 и совместимые деривативы, включая CentOS 7-8
- РЕД ОС 7.3 «Муром»
- Astra Linux 1.7 Special Edition
- Debian 11 «Bullseye»
- Ubuntu 20.04 (Focal Fossa), 22.04 (Jammy Jellyfish) и совместимые деривативы
- Семейство Alt Linux на основе платформы p10.

Picodata поддерживает актуальные репозитории с готовыми пакетами Tarantool и Picodata для RHEL (Yum/DNF) и Ubuntu LTS (Apt). Для Alt Linux предоставляется отдельные RPM-пакеты с Picodata и нашим форком СУБД Tarantool.

Форк Tarantool, развиваемый совместно с Picodata, имеет ряд технологических отличий от стандартного Tarantool, в том числе поддержку кластерных запросов SQL, дополнительные Rust bindings и т.д.

#### 3.1. Установка в RHEL/CentOS

Подключите репозиторий Picodata, используя следующие команды в терминале:

```
sudo rpm --import https://download.picodata.io/tarantool-picodata/el/RPM-GPG-KEY-kdy
# для RHEL/CentOS 7:
sudo yum install -y
https://download.picodata.io/tarantool-picodata/el/7/x86_64/picodata-release-1.1.0.11-1.el7.x86_64.rpm
# для RHEL/CentOS 8:
sudo yum install -y
https://download.picodata.io/tarantool-picodata/el/8/x86_64/picodata-release-1.1.0.11-1.el8.x86_64.rpm
```

---

После успешного выполнения команды в вашей системе появится дополнительный репозиторий в /etc/yum.repos.d/picodata.repo.

Установите пакет Picodata, скопировав и вставив в терминал следующие команды:

```
sudo yum clean all
sudo yum install -y picodata
```

Для того чтобы установить только пакет с нашей версией Tarantool, введите следующую команду:

```
sudo yum --disablerepo="*" --enablerepo="picodata" install -y tarantool-picodata
```

Вы также можете вручную скачать наши пакеты по адресу <https://download.picodata.io/tarantool-picodata>, не подключая репозиторий, и установить их самостоятельно.

## 3.2. Установка в РЕД ОС/RED OS 7.3 «Муром»

Подключите репозиторий Picodata, используя следующие команды в терминале:

```
rpm --import https://download.picodata.io/tarantool-picodata/el/RPM-GPG-KEY-kdy
yum install -y
https://download.picodata.io/tarantool-picodata/redos/7/x86_64/picodata-release-
1.1.1.0-1.el7.x86_64.rpm
```

---

После успешного выполнения команды в вашей системе появится дополнительный репозиторий в /etc/yum.repos.d/picodata.repo.

Установите пакет Picodata, скопировав и вставив в терминал следующие команды:

```
sudo yum clean all
sudo yum install -y picodata
```

Для того чтобы установить только пакет с нашей версией Tarantool, введите следующую команду:

```
sudo yum --disablerepo="*" --enablerepo="picodata" install -y tarantool-picodata
```

## 3.3. Установка в Astra Linux 1.7 SE

Подключите репозиторий Picodata, используя следующие команды в терминале:

```
su -i
apt-get install gpg curl apt-transport-https
curl -s https://download.picodata.io/tarantool-picodata/ubuntu/picodata.gpg.key
| gpg --no-default-keyring --keyring
gnupg-ring:/etc/apt/trusted.gpg.d/picodata.gpg --import
chmod 644 /etc/apt/trusted.gpg.d/picodata.gpg
echo "deb [arch=amd64] https://download.picodata.io/tarantool-picodata/astra/
orel main" > /etc/apt/sources.list.d/picodata.list
```

Установите пакет Picodata, скопировав и вставив в терминал следующую команду от пользователя root:

```
apt install picodata
```

Для того чтобы установить только пакет с нашей версией Tarantool, введите следующую команду:

```
apt install tarantool-picodata
```

Для работы в режиме Замкнутая программная среда (ЗПС) выполните дополнительную команду от пользователя root:

```
curl -s https://download.picodata.io/tarantool-picodata/astra/picodata_pub.key -
o /etc/digisig/keys/picodata_pub.key
```

После этого потребуется перезагрузка ОС.



### 3.4. Установка в Debian

Подключите репозиторий Picodata, используя следующие команды в терминале:

```
su -
apt-get install gpg software-properties-common
curl -s https://download.picodata.io/tarantool-picodata/ubuntu/picodata.gpg.key
| gpg --no-default-keyring --keyring
gnupg-ring:/etc/apt/trusted.gpg.d/picodata.gpg --import
chmod 644 /etc/apt/trusted.gpg.d/picodata.gpg
add-apt-repository "deb [arch=amd64] https://download.picodata.io/tarantool-
picodata/debian/ $(lsb_release -sc) main"
```

---

Установите пакет Picodata, скопировав и вставив в терминал следующую команду:

```
sudo apt install picodata
```

Для того чтобы установить только пакет с нашей версией Tarantool, введите следующую команду:

```
sudo apt install tarantool-picodata
```

### 3.5. Установка в Ubuntu 20.04 и 22.04

Скопируйте и вставьте следующие команды в терминал:

```
curl -s https://download.picodata.io/tarantool-picodata/ubuntu/picodata.gpg.key
| sudo gpg --no-default-keyring --keyring
gnupg-ring:/etc/apt/trusted.gpg.d/picodata.gpg --import
sudo chmod 644 /etc/apt/trusted.gpg.d/picodata.gpg
sudo add-apt-repository "deb [arch=amd64]
https://download.picodata.io/tarantool-picodata/ubuntu/ $(lsb_release -sc) main"
sudo apt install picodata
```

Для того чтобы установить только пакет с нашей версией Tarantool, введите следующую команду:

```
sudo apt install tarantool-picodata
```

### 3.6. Установка в Alt Linux

Скопируйте и вставьте следующие команды в терминал.

Для установки Picodata:

```
su -
curl https://download.picodata.io/tarantool-picodata/altlinux/p10/picodata-
release-1.0.2.7-1.p10.x86_64.rpm -o picodata-release-1.0.2.7-1.p10.x86_64.rpm
apt-get install ./picodata-release-1.0.2.7-1.p10.x86_64.rpm
apt-get update
apt-get install picodata
```

Для установки Tarantool следует добавить наш репозиторий (см. выше) и затем ввести следующие команды:

```
su -
```

```
apt-get install tarantool-picodata
```

## 4. Проверка установки

Когда программное обеспечение Picodata установлено, то можно проверить наличие в системе основного исполняемого файла `picodata`, используя следующую команду:

```
which picodata
```

Ответом на неё должно быть значение `/usr/bin/picodata`, либо — если вы устанавливали ПО вручную с другим префиксом — иное расположение, включенное в `$PATH`.

Чтобы убедиться в работоспособности ПО, а также посмотреть его версию, используйте следующую команду:

```
picodata --help
```

В состав ПО также включены юнит-тесты, позволяющие проверить работоспособность основных функций. Юнит-тесты можно запустить следующей командой:

```
picodata test
```

Пример вывода команды:

```
running 6 tests
test test_traft_pool ... ok
test test_storage_peers ... ok
test test_storage_state ... ok
test test_storage_log ... ok
test test_mailbox ... ok
test test_version ... ok

test result: ok. 6 passed; 0 failed; finished in 0.88s
```