

Node.js на Ubuntu 20.04: как установить и настроить

 cloud.ru/blog/node-js-na-ubuntu-24-04

Александра Гонтарева

Продукты из этой статьи:

Evolution Object Storage

Evolution Managed Kubernetes

Evolution Compute

Кроссплатформенная среда для разработки клиентских приложений Node.js была разработана в 2009 году, значительно упростив работу с написанными на JavaScript приложениями. Благодаря ей, писать на этом языке стало можно не только в браузере, но и на сервере.

В техническом смысле Node.js — это приложение на C++, которое получает и выполняет код JavaScript, превращая его в язык общего назначения с помощью собственного интерфейса.

Использовать Node.js можно при разработке программ для разных операционных систем (Linux, OS X и Windows), кроссплатформенных приложений, а также API. Эта среда идеальна для создания сервисов с постоянным обменом информацией с пользователем — соцсетей, чатов, онлайн-игр и тому подобных.



Логотип Node.js

В числе преимуществ Node.js для разработки:

- Открытый исходный код. Node.js на Ubuntu 24.04 можно пользоваться бесплатно в коммерческих и некоммерческих целях, что делает ее удобным инструментом для стартапов, небольших компаний и разработчиков-энтузиастов.
- Высокая производительность. Мощность платформы базируется на асинхронной модели ввода-вывода и событийно-ориентированной архитектуре.
- Широкий набор инструментов. Node.js располагает большим числом модулей, фреймворков и инструментов, которые доступны через дефолтный пакетный менеджер npm.
- Мультиплатформенность. Node.js позволяет создавать кроссплатформенные приложения, которые одинаково хорошо работают на операционных системах Windows, macOS и Linux.
- Простая автоматизация. Ubuntu 24.04 широко используется в DevOps-процессах благодаря встроенным средствам автоматизации: Ansible, Terraform и Bash-скрипты. А Node.js позволяет автоматизировать и ускорить сборку, тестирование и развертывание приложений.

Перед установкой новой версии Node.js важно убедиться, что на компьютере нет предыдущих установок, которые могут вызвать конфликты, особенно если старое ПО устанавливалось вручную или из устаревших репозиториях.

Для этого следуйте инструкции:

1. Введите:

Команда удалит старую версию программного обеспечения со встроенным пакетным менеджером APT.

2. Далее введите команду:

Это очистит конфигурации, связанные со старой версией Node.js.

3. Воспользуйтесь командой:

Так вы очистите уже неиспользуемые зависимости с файлами прежней версии.

В результате старая версия ПО будет удалена, а на жестком диске освободится место для новой.

```
wardah@wardah:~$ sudo apt remove nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp javascript-common libc-ares2 libfprint-2-tod1 libjs-inherits
  libjs-is-typedarray libjs-psl libjs-typedarray-to-buffer libllvm9
  libnode-dev libnode64 libpython2-stdlib libpython2.7-minimal
  libpython2.7-stdlib libssl-dev libuv1-dev linux-image-5.8.0-48-generic
  linux-modules-5.8.0-48-generic linux-modules-extra-5.8.0-48-generic
  nodejs-doc python-pkg-resources python2 python2-minimal python2.7
  python2.7-minimal
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  node-abbrev node-ajv node-ansi node-ansi-align node-ansi-regex
  node-ansi-styles node-ansistyles node-aproba node-archy
  node-are-we-there-yet node-asap node-asn1 node-assert-plus node-asynckit
  node-aws-sign2 node-aws4 node-balanced-match node-bcrypt-pbkdf node-bl
  node-bluebird node-boxen node-brace-expansion node-builtin-modules
  node-builtins node-cacache node-call-limit node-camelcase node-caseless
  node-chalk node-chownr node-ci-info node-cli-boxes node-cliui node-clone
  node-co node-color-convert node-color-name node-colors node-columnify
  node-combined-stream node-concat-map node-concat-stream node-config-chain
  node-configstore node-console-control-strings node-copy-concurrently
```

Интерфейс Node.js в Ubuntu 24.04

Далее важно обновить компоненты операционной системы. Для этого нужно ввести еще несколько команд, чтобы завершить подготовку к установке Node.js:

1. Добавьте команду:

Это позволит загрузить из репозитория информацию о последних версиях ПО.

2. После введите в терминал:

Команда запустит загрузку и установку доступных обновлений для всех устаревших компонентов ОС Ubuntu.

3. Если есть необходимость обновить стандартные пакеты и ядро системы, дополнительно вбейте команду:

Некоторые библиотеки и инструменты обязательны для работы Node.js. Они используются для компиляции исходного кода или выполнения других системных задач. В первую очередь вам стоит установить пакет компиляторов, отладчиков и прочих утилит, необходимых для корректной работы ПО. Для этого воспользуйтесь командой:

Если в системе по каким-то причинам еще не установлена утилита curl, установите ее при помощи команды:

Эта утилита используется для загрузки скриптов из репозитория.

Последним шагом при подготовке к установке Node.js рекомендуем использовать команду:

Она перезагрузит систему — в результате все внесенные изменения будут корректно применены.

Есть как минимум три способа, которыми можно установить Node.js на операционную систему Ubuntu 24.04. Рассмотрим каждый из них.



Node.js в процессе установки

Способ 1: установка через репозитории Ubuntu

Это самый простой метод. Но у него есть недостаток: всегда есть вероятность, что в итоге будет установлена не самая актуальная версия программного обеспечения.

Процесс установки такой:

1. Введите команду:

Она обновит список доступных пакетов.

2. Далее используйте команду:

И дождитесь завершения установки.

Когда установка завершится, введите команду `node -v` для проверки текущей версии ПО. Если она отличается от той, которую вы хотите использовать, проделайте шаги из раздела выше про подготовку системы. Затем воспользуйтесь альтернативными способами установки, которые мы рассмотрим ниже.

Кроме того, можно применить команду `apt` — она позволяет выбрать версию ПО для загрузки. Для этого введите:

Способ 2: установка из репозитория NodeSource

Установка ПО из PPA (Personal Package Archive) — это гарантированный способ получить актуальную версию ПО. Чтобы им воспользоваться, введите в терминал:

С помощью PPA можно не только устанавливать, но и обновлять текущую версию ПО до наиболее актуальной через ввод `sudo apt update`.

Способ 3: установка через Node Version Manager

Метод позволяет не только устанавливать ПО, но еще и гибко управлять несколькими версиями Node.js. Для его использования сперва введите команду для установки NVM, которая выглядит так:

Далее приведите утилиту в действие с помощью команды:

А теперь, чтобы установить актуальную версию ПО, введите в терминал:

И ждите выполнения.

Если по каким-то причинам предложенные выше способы установки Node.js не подходят, есть альтернатива. Один из них — это установка программного обеспечения из бинарных файлов. Скачать их актуальные версии можно на официальном сайте по [ссылке](#). Они представлены в виде архива, который надо распаковать.

После распаковки архива нужно настроить путь к файлам. Для этого введите в терминал команду:

А затем:

При этом укажите путь до файлов.

Еще один альтернативный способ установки предлагает воспользоваться исходными кодами. Это не самый простой вариант, но у него есть большое преимущество — он позволяет установить кастомизированную версию Node.js.

Кастомизированная версия Node.js — это модифицированная сборка платформы, настроенная под конкретные требования. Она может включать или исключать определенные модули, изменять стандартные настройки, добавлять собственные функции или патчи. Используется для оптимизации производительности, уменьшения размера приложения или реализации специфических функций, которые не предусмотрены в стандартной версии.

Для скачивания исходных кодов введите команду:

И дождитесь загрузки.

После этого для сборки Node.js поочередно вводите команды:

1. `cd node`
2. `./configure`
3. `make -j4` (ключ `-j4` задействует 4 потока для ускорения сборки)
4. `sudo make install`

При работе в облаке [Cloud.ru среду для разработки Node.js](#) можно бесплатно скачать на маркетплейсе и запустить прямо на виртуальной машине. Настройка интуитивно понятна, есть подробная инструкция, а техподдержка готова ответить на вопросы.

После установки Node.js вам будет необходимо установить или обновить npm (Node Package Manager) — менеджер пакетов для работы с JavaScript. На этом этапе для начала нужно проверить текущую версию npm. Для этого введите команду `npm -v`. Если команда возвращает номер версии (например, 9.2.0), значит, npm установлен. Если вывод указывает на ошибку, нужно установить npm вручную. Для установки npm введите в консоль команду `sudo apt install npm -y`.

Чтобы получить доступ к последним функциям и обновлениям, рекомендуется обновить npm. Для этого нужно ввести в терминал команду `sudo npm install -g npm@latest`.

Node.js включает встроенную REPL-среду, которая позволяет выполнять JavaScript-код в интерактивном режиме. Это полезно для быстрого тестирования платформы и выполнения простых операций.

Как протестировать установку:

- нужно перейти в интерактивный режим и ввести `node`;
- далее отправить код: например:

При верной установке на экране должно появиться то, что написали в скобках — Node.js готово к работе.

Работу Node.js можно проверить, если создать простое серверное приложение, которое обрабатывает HTTP-запросы. Для этого нужно перейти в рабочую директорию и создать в ней новый тестовый файл, используя команду `nano app.js`.


```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Например, код будет такой

Запускается код командой `node app.js`. Если Node.js работает корректно, на экране появится сообщение `Server running at http://127.0.0.1:3000/`. Далее открываем браузер и вводим в адресную строку адрес `http://127.0.0.1:3000/`. В открывшемся окне должна появиться надпись «Hello World! Node.js работает корректно». Если все действительно так, тестирование можно считать успешным.

Подготовить среду Node.js — значит настроить управление глобальными пакетами и работу с проектами через `package.json`, а также выбрать редактор кода и провести настройку дополнительных инструментов при необходимости.

- Чтобы настроить пакеты `nodemon`, `eslint` и `prettier`, используйте команду `npm install -g nodemon eslint prettier` и дождитесь ее исполнения. Утилиты будут установлены автоматически.
- Для управления проектами и зависимостями через `package.json`, выполните команду `npm init -y`. А если хотите заполнить основные поля по типу названия, версии и автора вручную, используйте аналогичную команду, но без ключа `-y`.
- Для установки зависимостей используйте команды типа `npm install express --save` или `npm install jest --save-dev`. Установленные зависимости автоматически добавляются в созданный ранее файл `package.json`, тогда как все модули перемещаются в папку «`node_modules`».

Node.js — это кроссплатформенная платформа с открытым исходным кодом для разработки веб-сервисов, API и приложений, известная своей производительностью и удобством.

- До установки Node.js важно удалить ее старые версии и обновить систему, чтобы избежать конфликтов версий.
- Node.js можно установить через репозитории, NodeSource, NVM или из исходного кода. Установка из бинарных файлов или исходного кода подходит для настройки кастомизированных версий.
- Npm устанавливается вместе с Node.js, но может быть обновлен до последней версии вручную. Использование REPL и создание тестового HTTP-сервера позволяет проверить корректность установки.
- Настройка глобальных npm-пакетов, package.json и редактора кода улучшает производительность и удобство разработки.
- Для дальнейшего погружения в тему Node.js рекомендуем изучить популярные библиотеки по типу Socket.io и Express, а также внедрить в работу nvm для гибкого управления версиями ПО.