GitLab CI против Jenkins: Различия и сходства [2023]

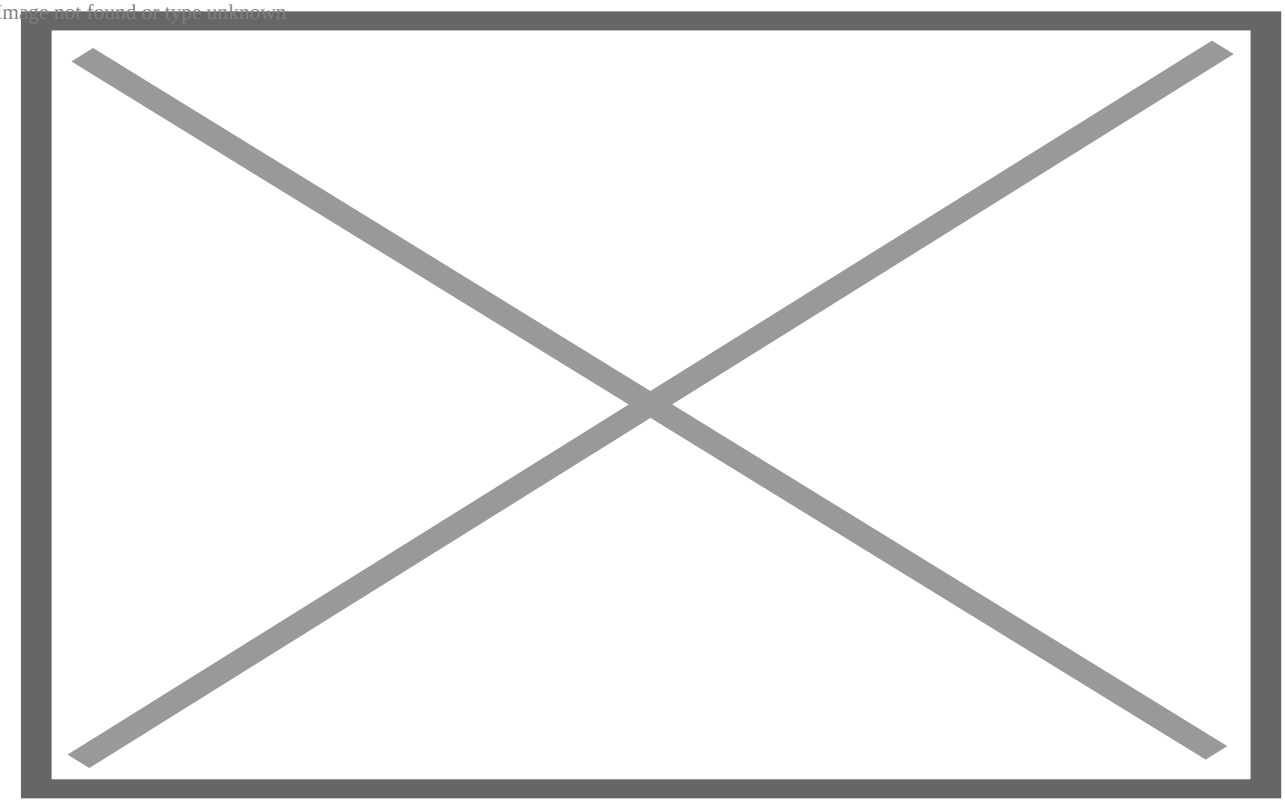
### Описание

Разработка программного обеспечения – это целый путь. Оно начинается с планирования, анализа, проектирования, разработки, тестирования, развертывания и сопровождения. Разработчикам программного обеспечения нужны различные инструменты для использования в этом путешествии. GitLab CI и Jenkins – одни из самых известных в нише непрерывной интеграции и доставки (CI/CD). CI/CD – это набор процессов, автоматизирующих разработку программного обеспечения. В этой статье GitLab vs. Jenkins мы дадим определение каждому инструменту, объясним, как каждый из них работает, и сравним их ключевые особенности.

## Что такое GitLab CI?

GitLab Continuous Integration and Delivery – это инструмент DevOps, который автоматизирует шаги, необходимые для сборки, тестирования и развертывания кода в производственной среде.

Image not found or type unknown



GitLab CI состоит из двух частей: непрерывной интеграции (CI) и непрерывной доставки/развертывания (CD). Задача Continuous Integration – автоматизировать все сборки. Эта функция также обеспечивает обратную связь через обзор кода и автоматизирует тесты безопасности и качества для вашего кода. Наконец, CI создает пакет релиза для развертывания в производственной среде. Бит непрерывного развертывания GitLab CI автоматически обеспечивает инфраструктуру и управляет тикетами, изменениями инфраструктуры и версией релиза. CD предлагает среду для постепенного развертывания кода и позволяет проверять и отслеживать все внесенные изменения. Эта функция также позволяет пользователям откатывать изменения, когда это необходимо. Когда эти две функции объединены, вы можете автоматизировать весь жизненный цикл разработки программного обеспечения (SDLC) с минимальным ручным вмешательством.

В двух словах, GitLab CI используется для:

- Хранения и управления кодом
- Автоматизировать конвейер CI/CD
- Отслеживание проблем
- Защита кода

- Совместная работа

### Плюсы использования GitLab CI

- **Увеличение скорости:** при использовании GitLab CI вам больше не придется выполнять вручную такие процессы, как сборка, тестирование и развертывание кода.
- **Улучшенное качество кода:** С помощью GitLab CI вы можете отлавливать ошибки и погрешности в коде до того, как они перейдут в продакшн.
- **Безопасность:** GitLab CI имеет различные функции безопасности, такие как сканирование уязвимостей, управление секретами и сканирование кода для снижения угроз взлома.
- **Гибкость:** Вы можете настроить GitLab CI в соответствии с потребностями вашей команды разработчиков. Вы можете использовать этот инструмент с основными языками и фреймворками.
- **Автоматизированное тестирование:** Вам не придется запускать тесты вручную, поскольку вы можете написать скрипты для автоматического написания и выполнения тестов.

## Что такое Jenkins?

Jenkins – это расширяемый сервер автоматизации. Этот инструмент с открытым исходным кодом помогает пользователям управлять и создавать конвейеры непрерывной интеграции и непрерывной доставки (CI/CD). Jenkins – любимый инструмент для DevOps и инженеров-программистов, поскольку он помогает улучшить качество, надежность и скорость доставки программного обеспечения.

Jenkins запускает серию “заданий” или этапов в конвейере. Задание состоит из нескольких шагов, которые выполняются в определенной последовательности. Шагом в Jenkins может быть сборка, тестирование, развертывание или любая другая задача, которую можно автоматизировать в жизненном цикле разработки программного обеспечения.

Области применения Jenkins можно описать следующим образом:

- Непрерывная интеграция
- Непрерывная доставка
- Автоматизированное тестирование

- Мониторинг и отчетность
- Сканирование кода
- Планирование заданий

### Плюсы использования Jenkins

- **Масштабируемость:** Вы можете использовать Jenkins как с маленькими, так и с большими приложениями.
- **Простая конфигурация:** Jenkins имеет простой процесс настройки, поддерживаемый достаточным количеством ресурсов.
- **Большое сообщество:** Jenkins существует уже почти два десятилетия и привлек множество последователей.
- **Разнообразие плагинов:** Вы можете улучшить функциональность Jenkins, используя широкий спектр плагинов.
- **Позволяет параллельно выполнять задания:** С помощью Jenkins вы можете выполнять различные задания одновременно, экономя время.

## Сходства GitLab CI и Jenkins

- Оба поддерживают непрерывную интеграцию и непрерывную доставку.
- Оба являются инструментами с открытым исходным кодом.
- Оба имеют большое сообщество и поклонников.
- Оба поддерживают различные языки и фреймворки.
- Оба автоматизируют различные задачи в жизненном цикле разработки программного обеспечения
- Оба имеют большую экосистему плагинов и расширений.

## GitLab CI против Jenkins: Сравнение возможностей

| Особенность | GitLab CI | Jenkins |
|---|---|---|
| **С открытым исходным кодом** | Да | Да |
| **Языковая поддержка** | Многие языки | Многие языки |

| | | |
|---|---|---|
| **Ценообразование** | Большинство функций доступно в бесплатном плане. У инструментатакже есть платные планыс большим количеством функций. | Более новая платформа; была создана в 2014 году. |
| **Зрелость** | Более новая платформа; создана в 2014 году. | Jenkins был запущен как Hudson, но в 2011 году был форкнут и переименован в Jenkins. |
| **Простота использования** | Прост в использовании. | Это может быть сложно для начинающих. |
| **Хостинг** | Внутренние и внешние | Внутренние и внешние |
| **Предусловия** | Node.JS, Git, Ruby, Go | Java Runtime Environment |

# GitLab CI vs. Jenkins: Подробное сравнение

Несмотря на то, что GitLab и Jenkins имеют общие черты, у этих двух программ есть явные различия, которые влияют на то, как они управляют процессом CI/CD. Вот некоторые из основных областей;

## Архитектура

**Jenkins** использует архитектуру мастер-работник для управления сборками.

- Мастер Jenkins планирует сборку заданий и распределяет их между рабочими для фактического выполнения. Эта функция также отслеживает состояние рабочих, собирает и агрегирует все результаты сборки в веб-панель.
- Jenkins worker – это исполняемый файл Java, запущенный на удаленной машине. Также известная как агент сборки, эта функция прослушивает все запросы от мастера и выполняет их. Вы можете иметь до 100+ узлов и добавлять или удалять рабочих по своему усмотрению.

**GitLab CI** является частью GitLab, веб-интерфейса для управления репозиториями,

запросами на слияние и многим другим. GitLab CI состоит из различных компонентов:

- Инструмент GitLab CI/CD позволяет управлять сборками.
- GitLab Runners выполняет все задания CI. Этот легкий процесс может выполняться в облаке или на вашей машине.
- Конфигурации CI/CD-конвейера определяются в файле ***.gitlab-ci.yml***. Этот файл определяет все задания, этапы и шаги в конвейере.

## Плагины

**Jenkins** имеет более 1800 плагинов, поддерживаемых сообществом. Эти плагины охватывают различные области, такие как сборка, развертывание и автоматизация проектов. Пользователи могут настраивать свои CI/CD конвейеры и расширять функциональность Jenkins. Разработчики могут создавать собственные плагины с помощью обширной документации Jenkins. Они также могут создавать плагины и добавлять их в каталог Jenkins. Jenkins имеет большое и активное сообщество, которое помогает создавать плагины.

**GitLab CI** позволяет подключаться/интегрироваться с внешними сервисами для расширения функциональности. Его библиотека плагинов/расширений меньше, чем у Jenkins, но она постоянно растет. Поскольку GitLab CI является частью GitLab, он поставляется с множеством встроенных функций. Пользователи могут настраивать/конфигурировать свои рабочие процессы с помощью файла .gitlab-ci.yml. Вы можете указать все задания, этапы и шаги.

## Трубопроводы

**Jenkins** позволяет использовать как декларативный, так и скриптовый синтаксис конвейеров. Вы можете использовать любой из этих двух подходов с помощью веб-интерфейса или Jenkinsfile. Последний вариант является наиболее предпочтительным. Вы можете хранить конвейеры в виде Jenkins-файлов в репозитории, где хранится исходный код. Платформа поставляется со встроенным веб-интерфейсом GUI, где можно отслеживать и визуализировать выполнение.

**GitLab CI** имеет файл ***.gitlab-ci.yml***, который определяет все конвейеры. Этот файл конфигурации на основе YML хранится в корневом каталоге проекта. ***gitlab-ci.yml*** известен своим простым синтаксисом с набором команд и предопределенных

ключевых слов, охватывающих наиболее распространенные задачи CI/CD. GitLab CI интегрируется с другими функциями GitLab, позволяя управлять исходным кодом, отслеживать проблемы, проводить обзоры кода, запросы на слияние и многое другое.

# Как установить Jenkins

Перед установкой Jenkins на ваш компьютер вам понадобится Java Runtime Environment (JRE). Скачайте Java с официального сайта, в зависимости от вашей операционной системы. Для демонстрации процесса установки я буду использовать Ubuntu. Если вы используете другую операционную систему, ознакомьтесь с нашей статьей об установке Jenkins и выполните перечисленные шаги.

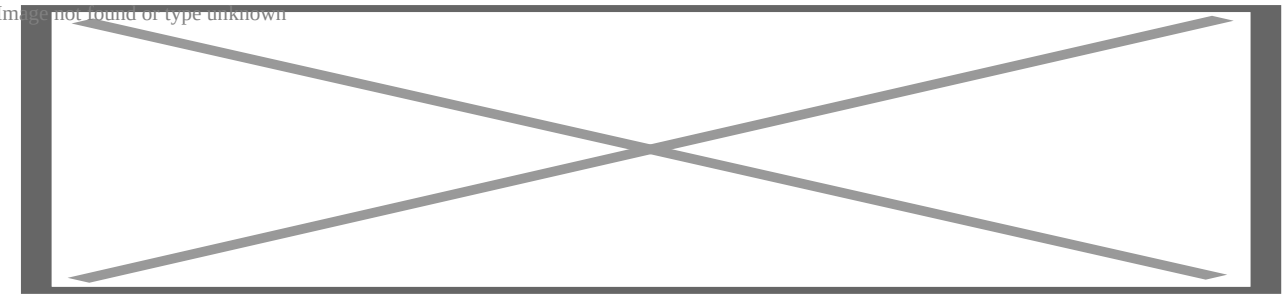
Для Ubuntu выполните следующие шаги:

**Шаг 1:** Проверьте, установлен ли JRE. Выполните эту команду:

```
java -version
```

Если он установлен, у вас будет что-то вроде этого:

Image not found or type unknown



Я установил версию “17.0.6”, но вы можете использовать более высокую версию.

**Шаг 2:** Импортируйте ключ GPG в вашу систему с помощью этой команды:

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

**Шаг 3:** Добавьте репозиторий программного обеспечения Jenkins с помощью этой команды:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

**Шаг 4:** Обновите систему с помощью этой команды:

```
sudo apt update
```

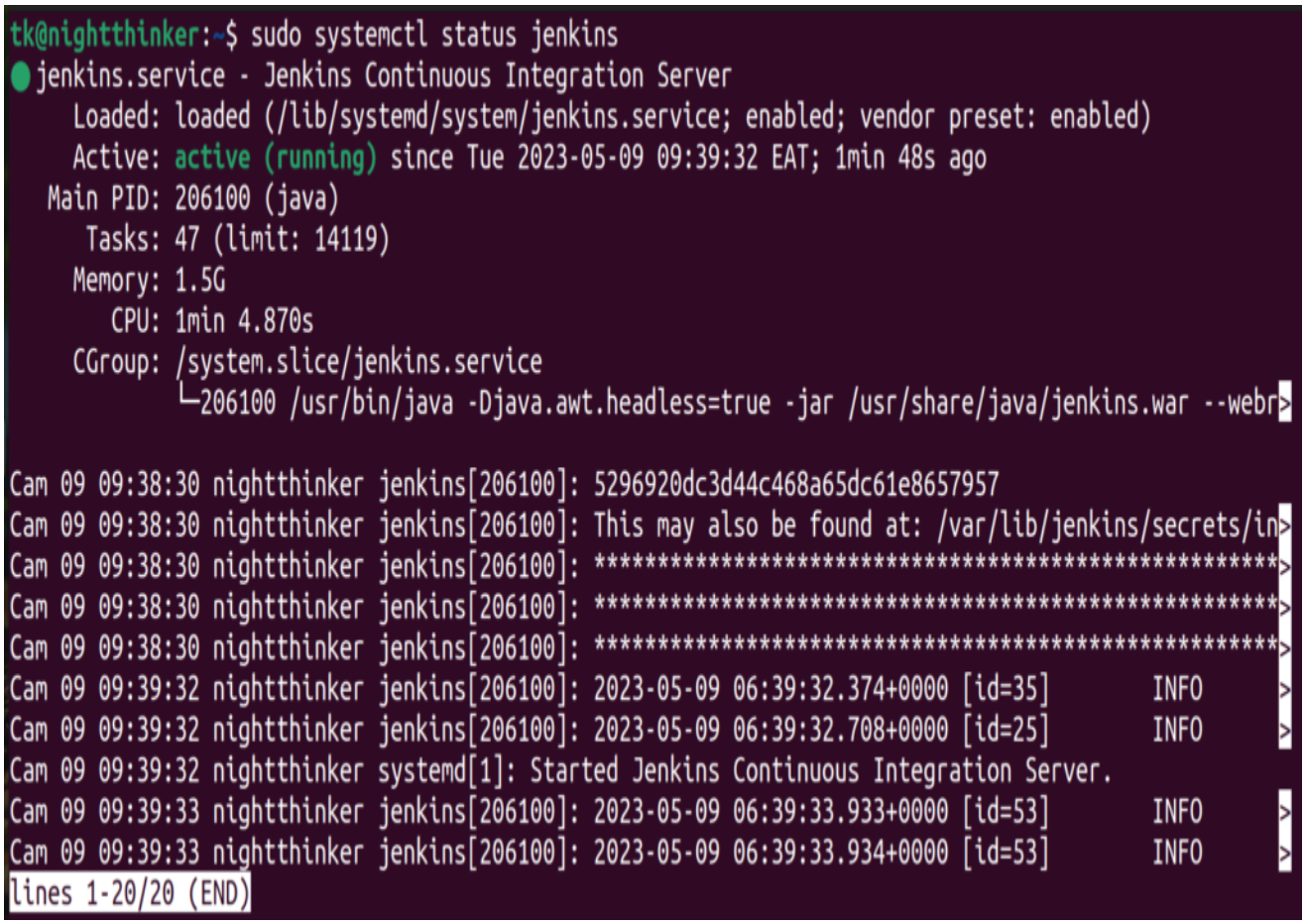
**Шаг 5:** Установите Jenkins с помощью этой команды:

```
sudo apt install jenkins -y
```

**Шаг 6:** Проверьте, установлен ли Jenkins, используя эту команду:

```
sudo systemctl status jenkins
```

Если он установлен, вы увидите что-то похожее на это:




Нажмите ctrl+z на клавиатуре, чтобы выйти и перейти к следующему шагу.

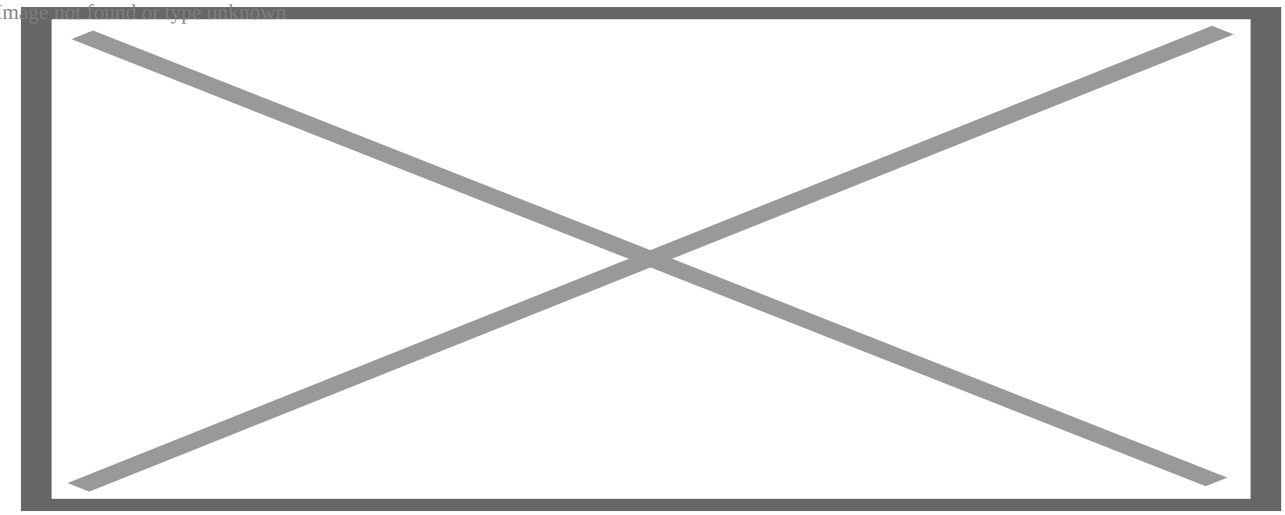
**Шаг 7:** Измените брандмауэр и Jenkins в вашей системе. Используйте эту команду:

```
sudo ufw status
```

**Шаг 8:** Проверьте состояние

```
sudo ufw status
```

Image not found or type unknown



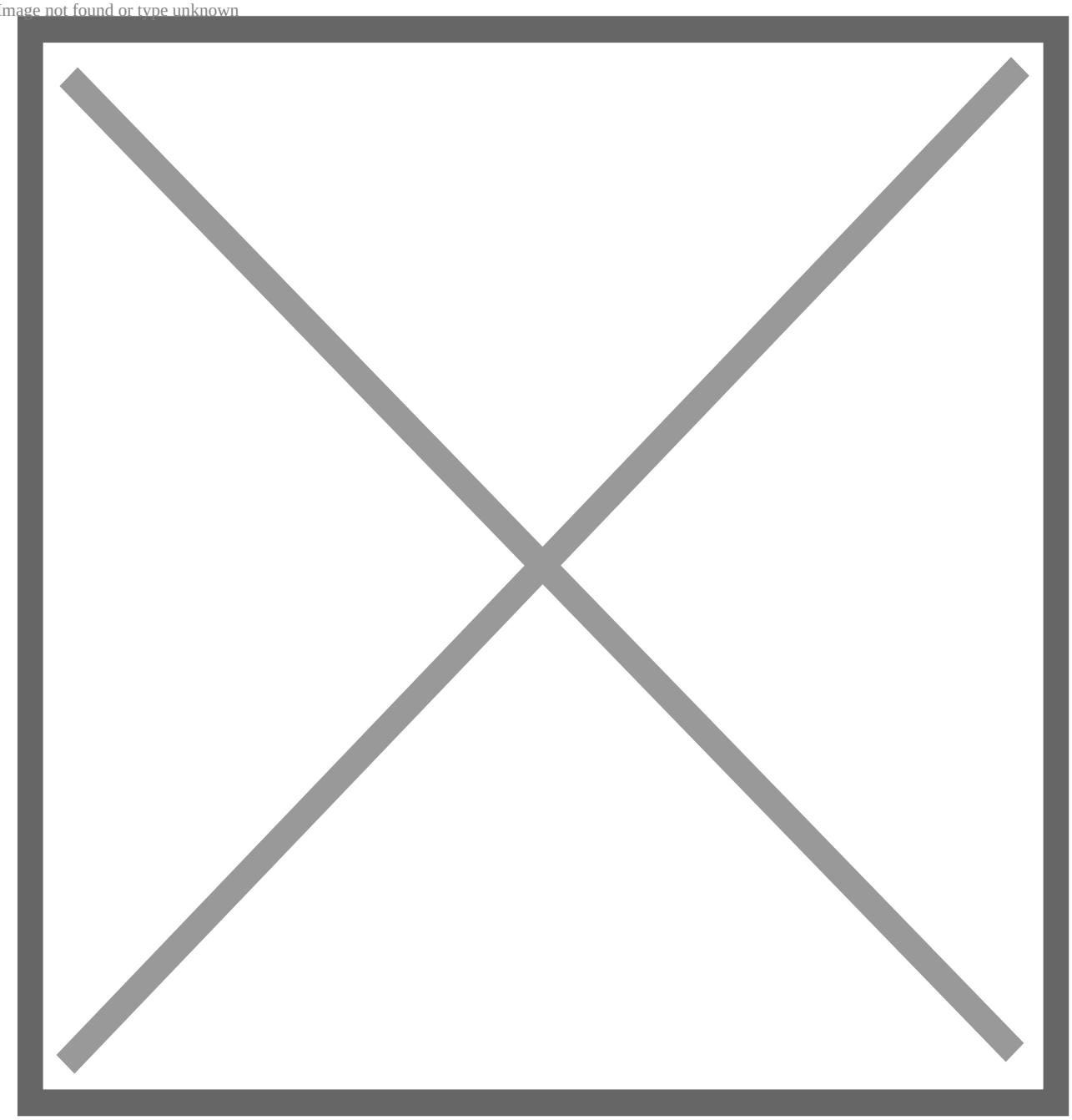
**Шаг 9:** Если на предыдущем шаге статус был “Неактивен”, включите его с помощью этой команды:

```
sudo ufw enable
```

**Шаг 10:** Откройте Jenkins в браузере, набрав в браузере следующее:

http://localhost:8080

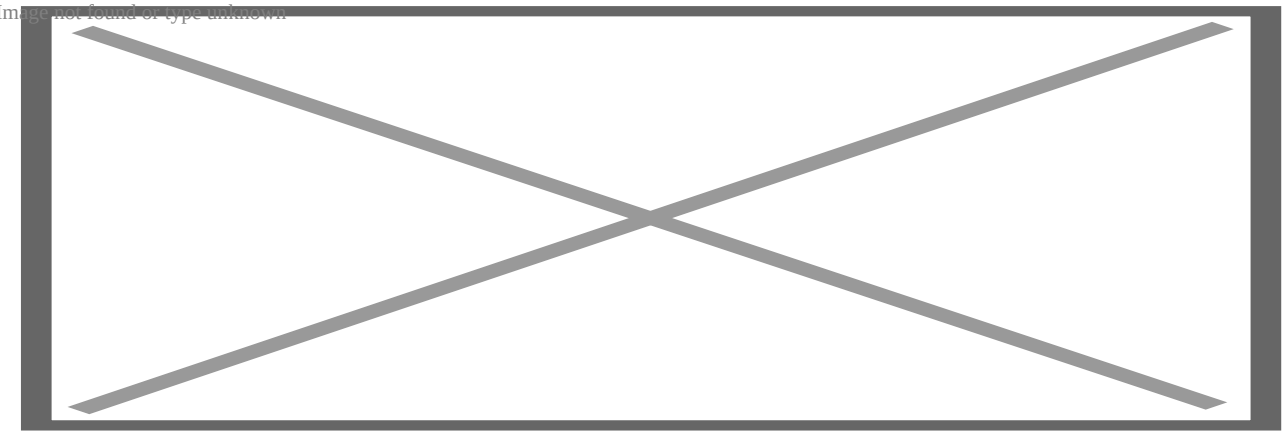
Image not found or type unknown



**Шаг 11:** Получите “Пароль администратора”, выполнив эту команду:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

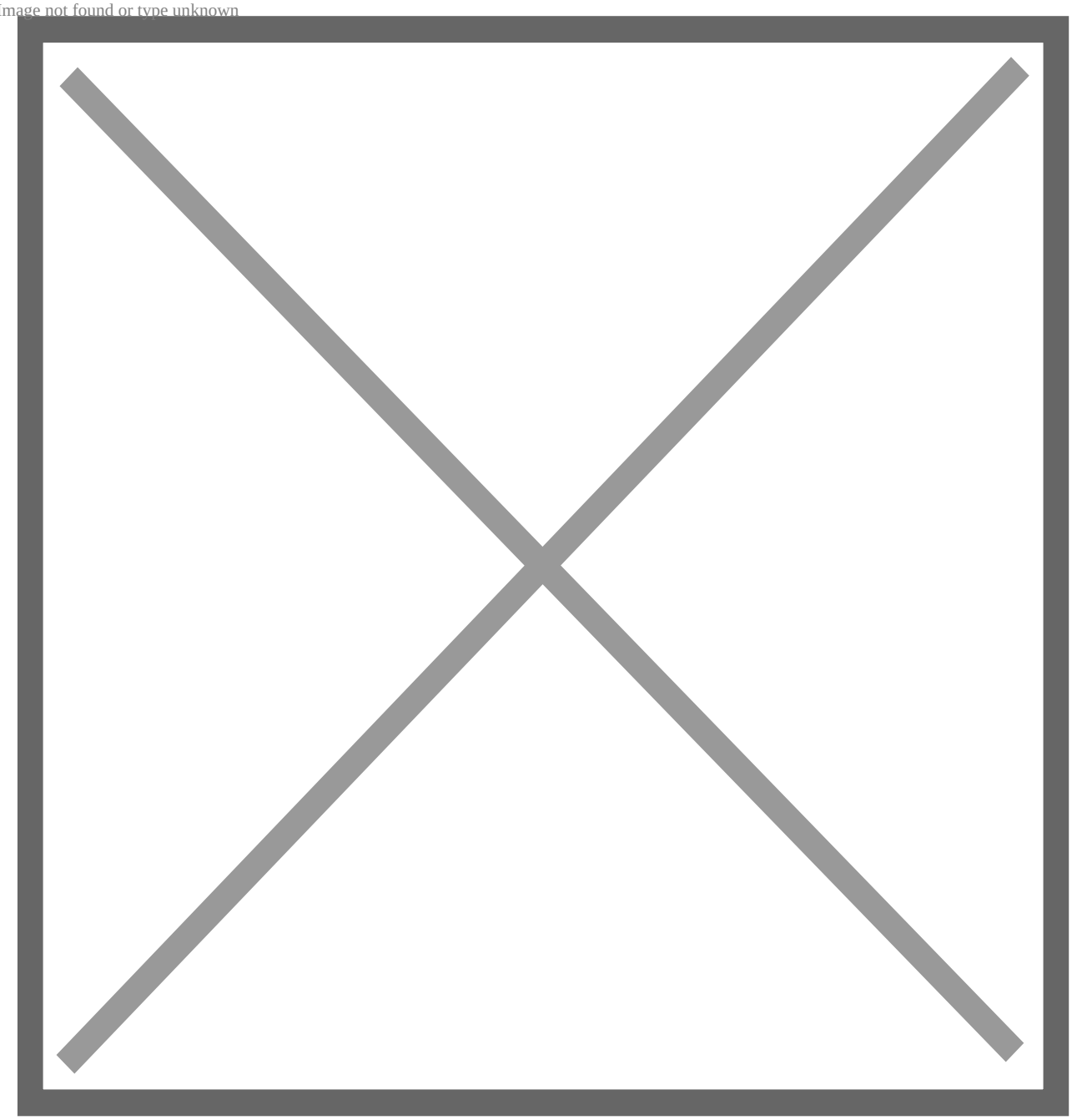
Image not found or type unknown



**Шаг 12:** Разблокируйте Jenkins

Скопируйте пароль, отображаемый на вашем терминале, вставьте в диалоговое окно “Пароль администратора”, которое вы открыли в десятом шаге, и нажмите ‘Continue’.

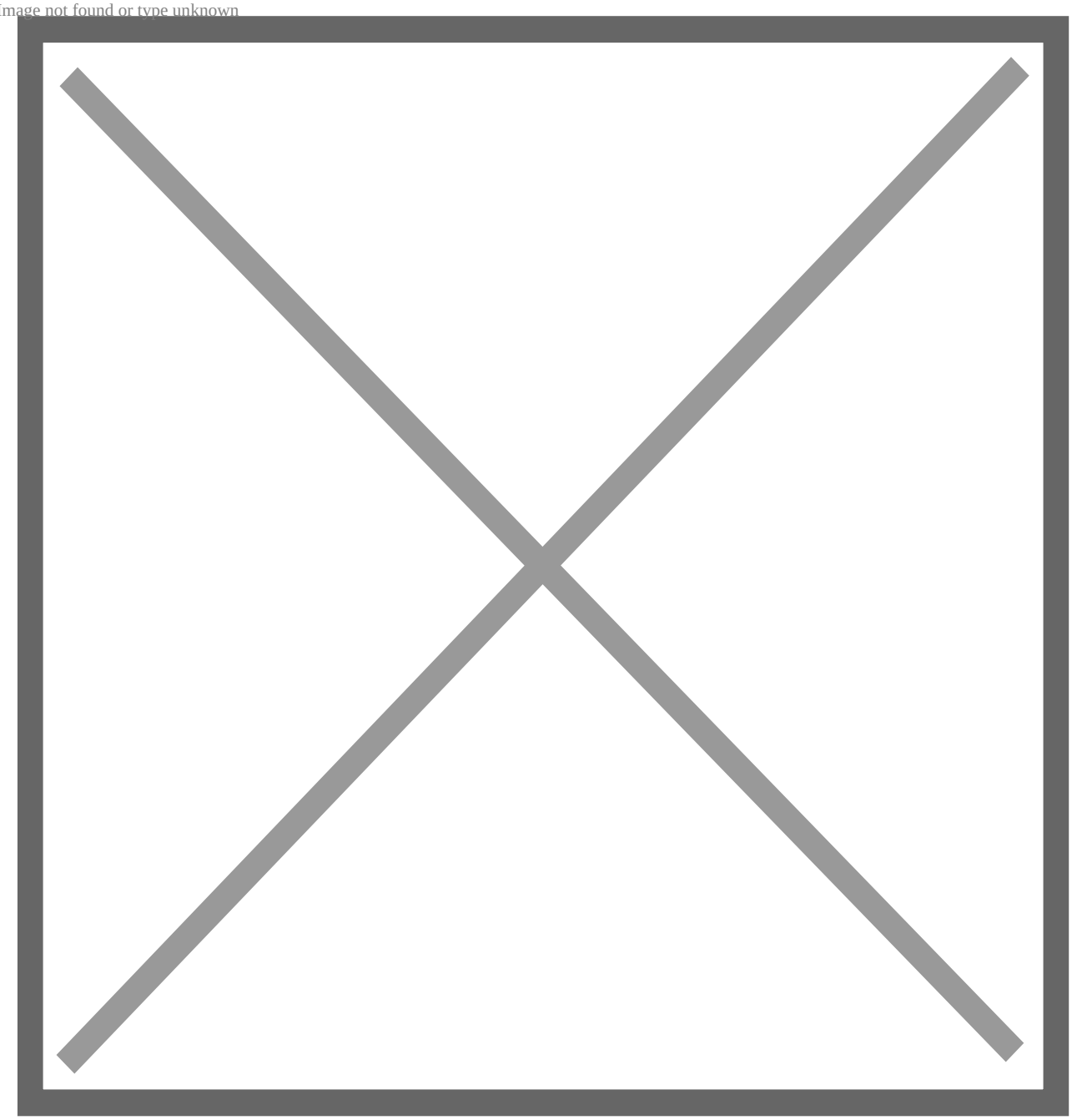
Image not found or type unknown



В новом окне вам будет предложено Установить плагины. Поскольку в данный момент вам не нужно много плагинов, вы можете выбрать плагины по умолчанию и перейти к следующему шагу

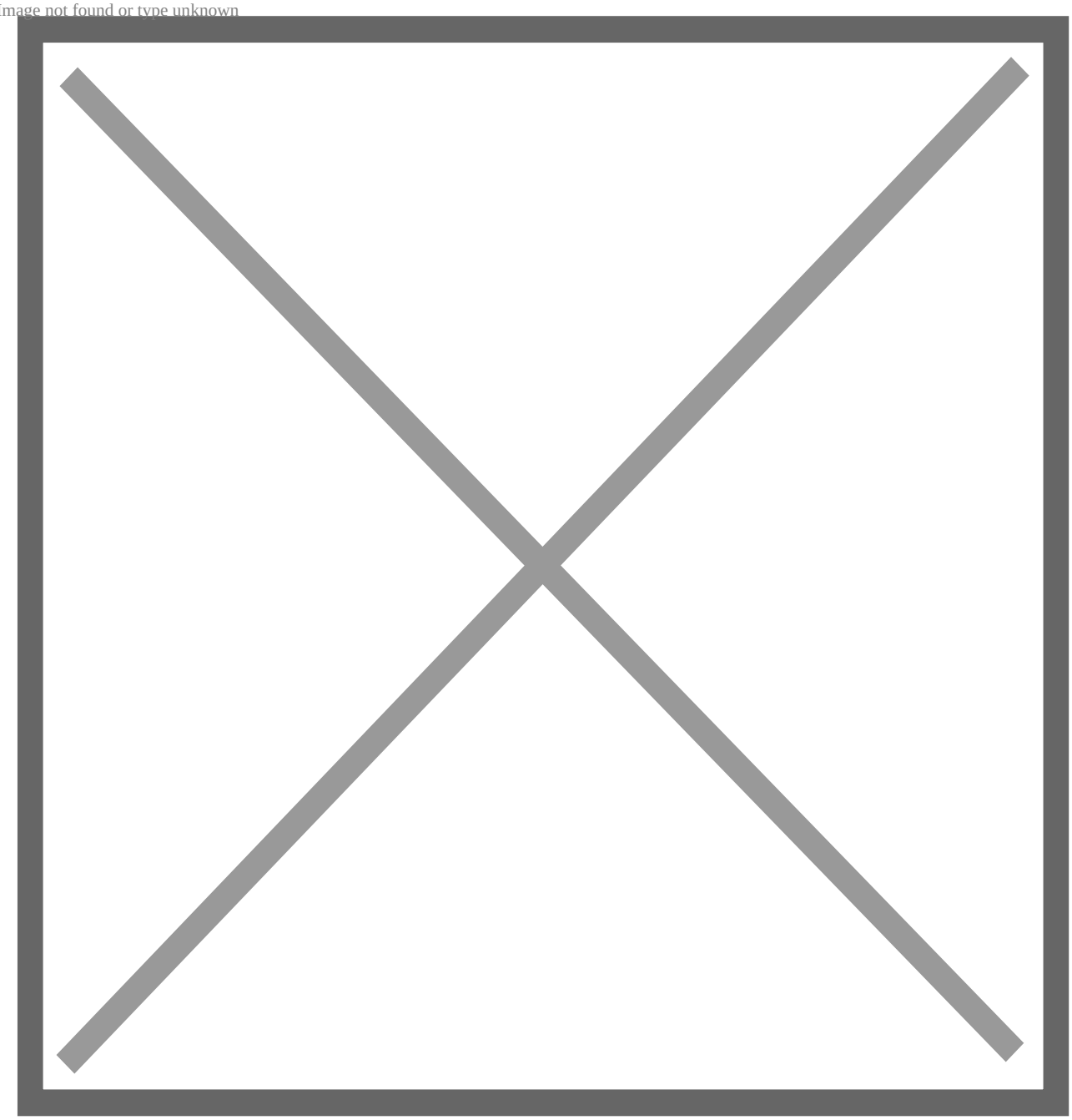
**Шаг 13:** Создайте свой аккаунт, “сохранить и продолжить”.

Image not found or type unknown



**Шаг 14:** Настройте мгновенную конфигурацию и начните использовать Jenkins

Image not found or type unknown



# Как установить GitLab CI

GitLab CI является частью GitLab. Чтобы получить GitLab CI, вы должны сначала установить GitLab Runner, агент, который выполняет все задания перед отправкой их в GitLab. Для демонстрации процесса я буду использовать Ubuntu. Если у вас другая операционная система, ознакомьтесь с официальной документацией. Для Ubuntu выполните следующие шаги:

**Шаг 1:** Обновите и настройте вашу систему:

```
sudo apt-get update
```

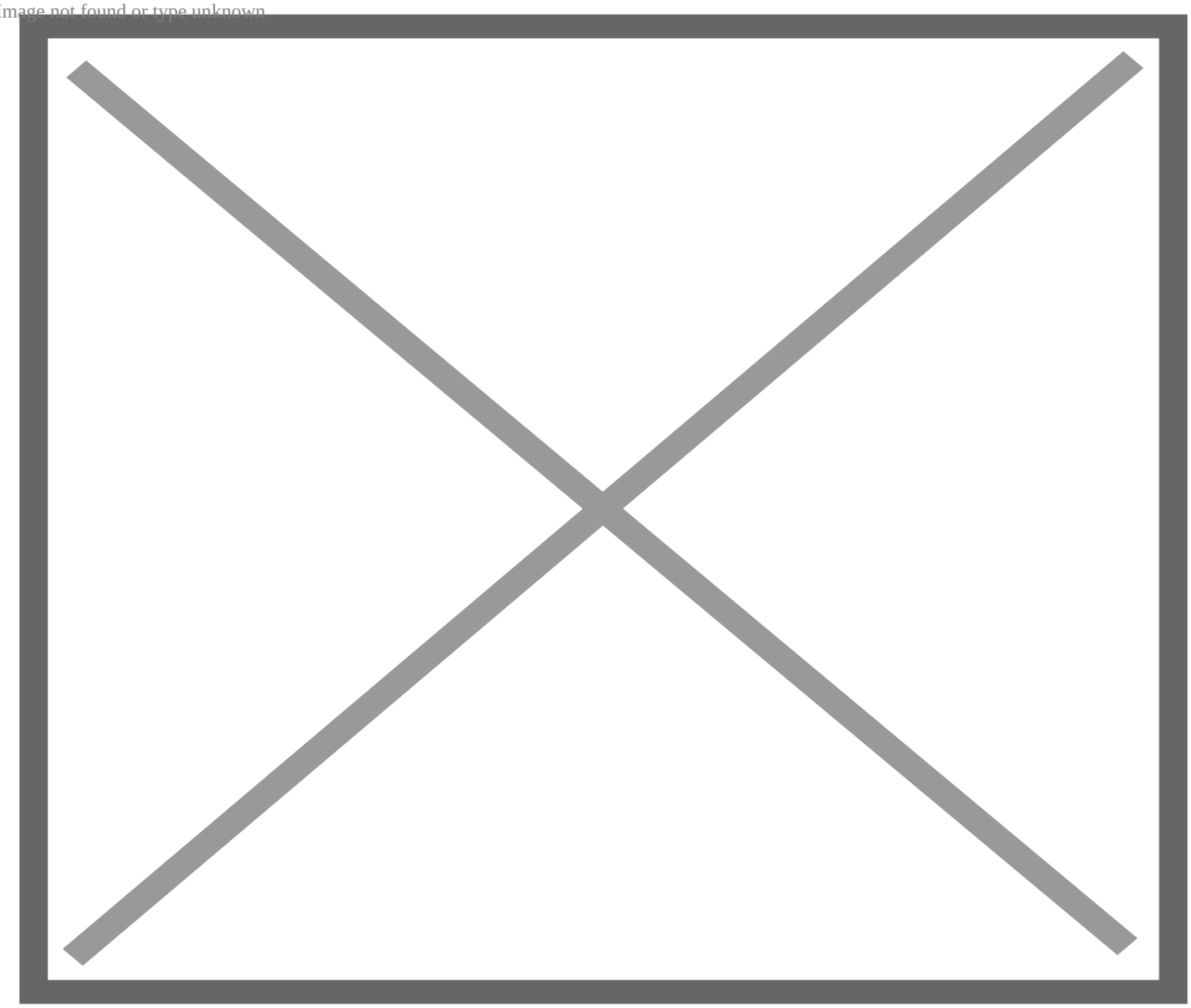
```
sudo apt-get install -y curl openssh-server ca-certificates tzdata perl
```

Вы можете настроить почтовое решение для отправки обновлений или пропустить этот шаг.

**Шаг 2:** Добавьте репозиторий пакетов GitLab

```
curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.
```

Image not found or type unknown



**Шаг 3:** Настройте свою учетную запись GitLab.

Существуют различные варианты хостинга GitLab. Следуйте инструкциям в зависимости от вашего выбора.

**Шаг 4:** Найдите имя хоста и войдите в систему. Нажмите эту команду в терминале, чтобы получить пароль:

```
/etc/gitlab/initial_root_password
```

**Необязательно:** Установите параметры связи, чтобы получать обновления продуктов и новости от GitLab.

# Ограничения Jenkins

- Jenkins может быть сложным в настройке для крупных проектов.
- Если Jenkins не настроен должным образом, он может быть уязвим для атак безопасности.
- При использовании Jenkins может быть сложно масштабировать большие проекты.
- Jenkins может потреблять много ресурсов при одновременном выполнении множества сборок.

# Ограничения Gitlab CI

- GitLab CI может быть сложным для больших проектов.
- Зависимость от GitLab.
- Масштабируемость является проблемой для больших проектов.
- Экосистема плагинов невелика по сравнению с Jenkins.

# Мнение автора

И GitLab CI, и Jenkins – фантастические инструменты в жизненном цикле разработки программного обеспечения. Я выберу Jenkins, если мне нужна более зрелая платформа и полный контроль над настройкой. С другой стороны, я выберу GitLab CI из-за его пользовательского интерфейса и необходимости воспользоваться преимуществами интеграции с GitLab. Мы уверены, что теперь вы можете увидеть различия между GitLab CI и Jenkins. Несмотря на то, что эти два инструмента предназначены для выполнения схожих функций, они различаются по возможностям и способам достижения своих функций. Выбор между этими двумя инструментами будет зависеть от характера проекта, который у вас есть, ваших навыков, вкуса и предпочтений.

**Дата Создания**
30.05.2023