




# CI/CD на основе GitLab

Различные системы CI/CD

 Проверить, идет ли запись

# Меня хорошо видно && слышно?

 Ставим "+", если все хорошо "-", если есть проблемы

Тема вебинара

# CI/CD на основе GitLab

## Различные системы CI/CD



Кирилл Сенчик

DevOps engineer в Cloud.ru

Опыт: 4 года в devops направлении, 5 лет в разработке, более 10 лет в ИТ

Telegram: [@kirill\\_senchik](https://t.me/kirill_senchik)



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в учебной группе в tg #OTUS  
CICD-GitLab-2024-12



Задаем вопросы в чат



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое на  
активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Маршрут вебинара

Цели и смысл

Обзор систем CI/CD

Сравнение систем CI/CD

Плюсы и минусы

Выбор подходящей системы CI/CD

Заключение



**GitLab**



**GitHub**



**Jenkins**



**Gitea**

# Цели вебинара

К концу занятия вы сможете

1. Понимать ключевые особенности и различия популярных CI/CD систем: GitLab, GitHub, Jenkins и Gitea

---

2. Анализировать сильные и слабые стороны каждой CI/CD системы для различных сценариев использования

---

3. Определять факторы, влияющие на выбор оптимальной CI/CD системы для конкретного проекта или организации

---

# СМЫСЛ

## Зачем вам это знать

1. Осознанный выбор инструментов:
  - Способность выбрать подходящую CI/CD систему для ваших проектов
  - Возможность аргументированно обосновать свой выбор руководству или команде

---
2. Оптимизация рабочих процессов:
  - Умение использовать сильные стороны выбранной CI/CD системы для повышения эффективности разработки
  - Способность минимизировать влияние слабых сторон системы на процесс работы

---
3. Стратегическое планирование:
  - Возможность учитывать факторы масштабируемости и адаптивности при планировании долгосрочных проектов
  - Способность предвидеть потенциальные проблемы и преимущества различных CI/CD систем в контексте развития проекта

---

# Обзор систем CI/CD

# GitLab

GitLab - это комплексная платформа DevOps, предоставляющая инструменты для всего жизненного цикла разработки ПО.

Ключевые особенности:

- Встроенные инструменты CI/CD
- Управление репозиториями Git
- Совместная разработка (merge requests, code reviews)
- Управление проектами и задачами
- Container Registry
- Поддержка как облачного, так и локального развертывания



# GitHub

GitHub - популярная платформа для хостинга репозиториях Git с интегрированными инструментами CI/CD.

Ключевые особенности:

- GitHub Actions для CI/CD
- Обширная экосистема интеграций
- Совместная разработка (pull requests, code reviews)
- Управление проектами и задачами
- GitHub Packages для хранения артефактов
- Преимущественно облачное решение, с возможностью локального развертывания (GitHub Enterprise)



# Jenkins

Jenkins - открытый сервер автоматизации, широко используемый для создания пайплайнов CI/CD.

Ключевые особенности:

- Высокая гибкость и настраиваемость
- Обширная библиотека плагинов
- Поддержка распределенной сборки
- Интеграция с различными инструментами и платформами
- Pipeline as Code с Jenkins Pipeline
- Преимущественно для локального развертывания, но есть облачные опции



# Gitea

Gitea - легковесный, самостоятельно размещаемый сервис Git с базовыми возможностями CI/CD.

Ключевые особенности:

- Простота установки и использования
- Низкие системные требования
- Базовые функции CI/CD через Gitea Actions
- Управление репозиториями и пользователями
- Интеграция с внешними CI/CD инструментами
- Подходит для небольших команд или личных проектов



# Вопросы?



Ставим "+",  
если вопросы есть

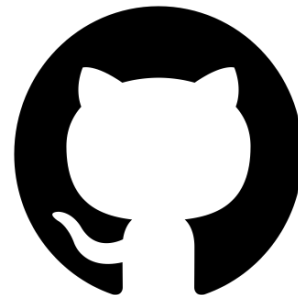


Ставим "-",  
если вопросов нет

# Сравнение систем CI/CD

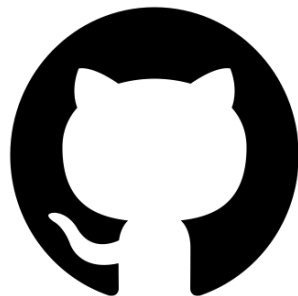
# Функциональность

1. GitLab:
  - a. Полный набор инструментов DevOps
  - b. Встроенные средства управления проектами
  - c. Расширенные возможности CI/CD
2. GitHub:
  - a. Фокус на совместной разработке
  - b. GitHub Actions для CI/CD
  - c. Интеграция с множеством сторонних сервисов
3. Jenkins:
  - a. Высокая гибкость настройки пайплайнов
  - b. Огромное количество плагинов
  - c. Поддержка сложных сценариев автоматизации
4. Gitea:
  - a. Базовые функции хостинга Git
  - b. Ограниченные возможности CI/CD
  - c. Фокус на легковесности и простоте



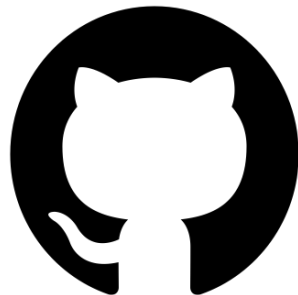
# Простота использования

1. GitLab:
  - a. Интуитивный веб-интерфейс
  - b. Встроенные инструменты упрощают настройку
  - c. Может быть сложным для новичков из-за обилия функций
2. GitHub:
  - a. Дружелюбный интерфейс
  - b. Простая настройка базовых CI/CD процессов
  - c. Хорошая документация и обучающие материалы
3. Jenkins:
  - a. Требуется значительных усилий для начальной настройки
  - b. Более сложный пользовательский интерфейс
  - c. Высокая кривая обучения для полного использования возможностей
4. Gitea:
  - a. Очень простой и интуитивно понятный интерфейс
  - b. Быстрая установка и настройка
  - c. Ограниченная функциональность упрощает освоение



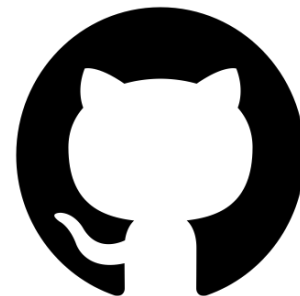
# Интеграция

1. GitLab:
  - a. Широкий спектр встроенных интеграций
  - b. Поддержка сторонних сервисов через API
  - c. Собственная экосистема расширений
2. GitHub:
  - a. Огромное количество интеграций с внешними сервисами
  - b. Обширный маркетплейс GitHub Apps
  - c. Поддержка API для создания собственных интеграций
3. Jenkins:
  - a. Тысячи доступных плагинов
  - b. Возможность интеграции с многими инструментами
  - c. Поддержка создания пользовательских плагинов
4. Gitea:
  - a. Ограниченные встроенные интеграции
  - b. Возможность интеграции с внешними CI/CD инструментами
  - c. Базовая поддержка веб-хуков



# Масштабируемость

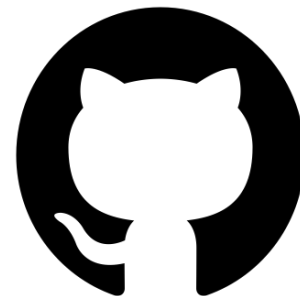
1. GitLab:
  - a. Хорошо масштабируется для крупных предприятий
  - b. Поддержка Kubernetes для автоматического масштабирования
  - c. Возможность распределенного развертывания
2. GitHub:
  - a. Отлично масштабируется для проектов любого размера
  - b. Надежная инфраструктура для крупномасштабных развертываний
  - c. Ограничения на количество действий в бесплатных планах



# Масштабируемость

## 3. Jenkins:

- a. Высокая масштабируемость с поддержкой распределенных сборок
- b. Может обрабатывать сотни параллельных задач
- c. Требуется тщательное планирование ресурсов при масштабировании



## 4. Gitea:

- a. Хорошо подходит для небольших и средних проектов
- b. Ограниченная масштабируемость для крупных предприятий
- c. Легкий вес позволяет эффективно использовать ресурсы



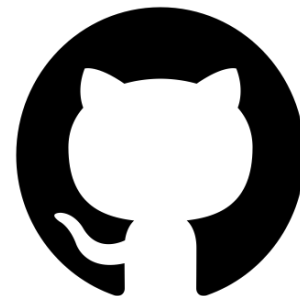
# Синтаксис пайплайнов. GitLab

```
.gitlab-ci.yml

stages:
  - build
  - test

build-job:
  stage: build
  script:
    - echo "Building the project..."

test-job:
  stage: test
  script:
    - echo "Running tests..."
```



# Синтаксис пайплайнов. GitHub

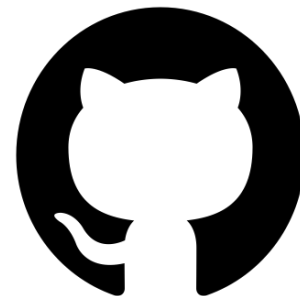
```
github/workflows/ci.yml

name: CI

on: [ push ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Build
        run: echo "Building the project..."

  test:
    needs: build
    runs-on: ubuntu-latest
    steps:
      - name: Test
        run: echo "Running tests..."
```



# Синтаксис пайплайнов. Gitea

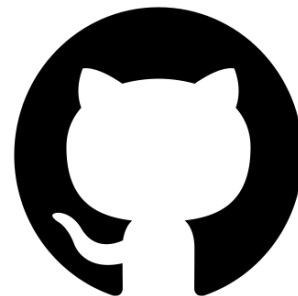
```
.gitea/workflows/ci.yml

name: CI

on: [ push ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Build
        run: echo "Building the project..."

  test:
    needs: build
    runs-on: ubuntu-latest
    steps:
      - name: Test
        run: echo "Running tests..."
```

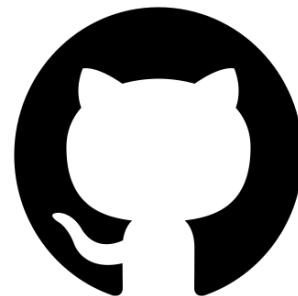


# Синтаксис пайплайнов. Jenkins

```
Jenkinsfile

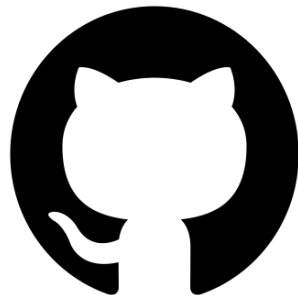
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        echo 'Building the project...'
      }
    }
    stage('Test') {
      steps {
        echo 'Running tests...'
      }
    }
  }
}
```



# Сообщество и поддержка

1. GitLab:
  - a. Активное сообщество разработчиков и пользователей
  - b. Обширная документация и обучающие ресурсы
  - c. Коммерческая поддержка для платных планов
2. GitHub:
  - a. Крупнейшее сообщество разработчиков в мире
  - b. Множество ресурсов, tutorиалов и обучающих материалов
  - c. Поддержка корпоративного уровня для бизнес-планов
3. Jenkins:
  - a. Большое сообщество с длительной историей
  - b. Обширная документация и множество плагинов от сообщества
  - c. Поддержка в основном через сообщество
4. Gitea:
  - a. Небольшое, но растущее сообщество
  - b. Базовая документация
  - c. Поддержка в основном через GitHub issues и форумы



# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# Плюсы и минусы

# GitLab

- Плюсы:
  - Комплексное решение для всего жизненного цикла DevOps
  - Встроенные инструменты CI/CD без необходимости сторонних интеграций
  - Гибкие настройки для различных рабочих процессов
  - Возможность самостоятельного хостинга или использования облачной версии
  - Хорошая документация и активное сообщество
- Минусы:
  - Может быть избыточным для небольших проектов
  - Более высокая стоимость по сравнению с некоторыми конкурентами
  - Крутая кривая обучения из-за большого количества функций
  - Производительность может страдать при больших нагрузках



# GitHub

- Плюсы:
  - Крупнейшая платформа для хостинга кода с огромным сообществом
  - GitHub Actions предоставляет мощные и гибкие возможности CI/CD
  - Отличная интеграция с множеством инструментов и сервисов
  - Удобный и интуитивно понятный интерфейс
  - Бесплатный план с базовыми функциями CI/CD
- Минусы:
  - Ограниченное количество бесплатных минут CI/CD
  - Меньший контроль над инфраструктурой по сравнению с самостоятельно хостируемыми решениями
  - Может быть дорогим для крупных команд или при интенсивном использовании CI/CD
  - Ограниченные возможности для частных репозиторий в бесплатном плане



# Jenkins

- Плюсы:
  - Высокая гибкость и настраиваемость
  - Огромное количество плагинов для расширения функциональности
  - Поддержка сложных сценариев автоматизации
  - Бесплатный и открытый исходный код
  - Сильное и активное сообщество
- Минусы:
  - Сложность в настройке и обслуживании
  - Устаревший UI
  - Требуется значительных ресурсов для хостинга и управления
  - Может быть нестабильным при использовании множества плагинов



## Jenkins



# Gitea

- Плюсы:
  - Легковесность и низкие требования к ресурсам
  - Простота установки и использования
  - Открытый исходный код и бесплатность
  - Хорошая производительность даже на слабом оборудовании и
  - Подходит для самостоятельного хостинга небольшими командами
- Минусы:
  - Ограниченные встроенные возможности CI/CD
  - Меньше функций по сравнению с более крупными платформами
  - Небольшое сообщество и ограниченная экосистема плагинов
  - Может не подходить для крупных предприятий или сложных проектов
  - Ограниченные возможности масштабирования



# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# Выбор подходящей системы CI/CD

# Факторы влияющие на выбор

1. Размер и сложность проекта:
  - a. Количество разработчиков и команд
  - b. Объем кода и частота изменений
  - c. Сложность процессов сборки и тестирования
2. Бюджет:
  - a. Стоимость лицензий и подписок
  - b. Расходы на инфраструктуру и обслуживание
  - c. Затраты на обучение персонала
3. Требования к безопасности:
  - a. Необходимость локального хостинга
  - b. Соответствие отраслевым стандартам (HIPAA, GDPR)
  - c. Управление доступом и аудит
4. Интеграция с существующими инструментами:
  - a. Совместимость с используемыми системами контроля версий
  - b. Интеграция с инструментами управления проектами и задачами
  - c. Поддержка используемых языков программирования и фреймворков



# Факторы влияющие на выбор

5. Масштабируемость:
  - a. Возможность роста вместе с проектом
  - b. Поддержка распределенных команд и удаленной работы
  - c. Способность справляться с увеличением нагрузки
6. Простота использования и обучения:
  - a. Удобство интерфейса
  - b. Качество документации
  - c. Доступность обучающих ресурсов
7. Поддержка и сообщество:
  - a. Доступность технической поддержки
  - b. Активность сообщества
  - c. Регулярность обновлений и исправлений



# Сценарии использования

- Небольшой стартап или индивидуальный проект:
  - Рекомендация: GitHub или Gitea
  - Причины: низкая стоимость, простота использования, достаточный функционал
- Средняя компания с несколькими проектами:
  - Рекомендация: GitLab или GitHub Enterprise
  - Причины: баланс между функциональностью и простотой управления, хорошая масштабируемость
- Крупное предприятие с множеством проектов и строгими требованиями безопасности:
  - Рекомендация: Self-hosted GitLab / Jenkins
  - Причины: полный контроль над инфраструктурой, гибкость настройки, соответствие корпоративным политикам
- Проект с открытым исходным кодом:
  - Рекомендация: GitHub
  - Причины: крупнейшее сообщество, отличные инструменты для совместной работы, бесплатные планы для открытых проектов



# Сценарии использования

- Компания с уникальными или сложными процессами сборки:
  - Рекомендация: Jenkins
  - Причины: высокая гибкость, возможность тонкой настройки, большое количество плагинов
- Образовательное учреждение или некоммерческая организация:
  - Рекомендация: Gitea или GitHub Education
  - Причины: низкая стоимость, простота использования, специальные программы для образовательных учреждений
- Проект с интенсивным использованием контейнеров и микросервисов:
  - Рекомендация: Self-hosted GitLab / GitHub с интеграцией Kubernetes
  - Причины: встроенная поддержка контейнеризации, инструменты для управления Kubernetes



# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

# Рефлексия

# Цели вебинара

К концу занятия вы сможете

1. Понимать ключевые особенности и различия популярных CI/CD систем: GitLab, GitHub, Jenkins и Gitea

---

2. Анализировать сильные и слабые стороны каждой CI/CD системы для различных сценариев использования

---

3. Определять факторы, влияющие на выбор оптимальной CI/CD системы для конкретного проекта или организации

---

# Рефлексия



Отправьте в чат эмодзи, который отражает ваше настроение после занятия.



Продолжите высказывание: теперь я умею...



Что планируете использовать в работе?

# Следующий вебинар



7 октября 2025

## Архитектура GitLab, GitLab Runner



Ссылка на вебинар  
будет в ЛК за 15 минут



Материалы  
к занятию в ЛК —  
можно изучать



Обязательный материал  
обозначен красной  
лентой

Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате

Спасибо за внимание!

# Приходите на следующие вебинары



Кирилл Сенчик

DevOps engineer в Cloud.ru

Опыт: 4 года в devops направлении, 5 лет в разработке, более 10 лет в ИТ

Telegram: [@kirill\\_senchik](https://t.me/kirill_senchik)



# ЭТОТ КОНТЕНТ КУПЛЕН НА САЙТЕ **SKLADCHIK.ORG**



**ТЫ ЕЩЕ  
НЕ В КЛУБЕ?**  
ПРИСОЕДИНЯЙСЯ  
И НАЧИНАЙ ЭКОНОМИТЬ!



## ЧТО ТАКОЕ КЛУБ «СКЛАДЧИК»?



### Платформа

Это платформа, где каждый день тысячи людей собираются вместе, чтобы общими усилиями находить, приобретать и изучать курсы интересные именно им.



### Сообщество

Это сообщество из 500 000 людей, открывших для себя выгодный способ быть в тренде самых актуальных и получать ценные знания по минимальной цене.



### Библиотека

Это крупнейшая библиотека инфопродуктов в которой можно найти практически любой курс или тренинг продававшийся за последние 10 лет, а также уникальные авторские инфопродукты, которые не получится найти больше нигде.

