



Общество с ограниченной ответственностью «Ребреин»

ИНН 7727409582, ОГРН 1197746106161

Адрес: 123056, город Москва, Большая Грузинская ул, д. 36а стр. 5а, офис 13

Утверждено

Приказом № ПР-1 от 17.06.2025 г.

Генеральный директор

 Фролкина Е.А.  
«17» июня 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА  
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ  
**«CI/CD-ПРОЦЕССЫ С JENKINS: НАСТРОЙКА, ПАЙПЛАЙНЫ, АВТОМАТИЗАЦИЯ»**

**Срок реализации:** 1 месяц

**Количество часов:** 72 акад. ч.

**Форма обучения:** заочная форма

**Формат обучения:** с применением  
исключительно дистанционных технологий

**Возраст обучающихся:** для лиц старше 17  
лет, имеющих или получающих среднее  
профессиональное и (или) высшее  
образование

Москва, 2025 г.

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Настоящая дополнительная профессиональная программа – программа повышения квалификации «CI/CD-процессы с Jenkins: настройка, пайплайны, автоматизация» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Профессиональным стандартом 06.026 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 26.10.2020 года № 60580;
- ФГОС высшего образования – бакалавриат по направлению 09.03.02 Информационные системы и технологии, утв. приказом Минобрнауки России от 19.09.2017 №926;
- Локальными нормативными актами ООО «Ребреин».

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

**Направленность программы:** Программа имеет техническую направленность.

### **Адресат:**

Программа ориентирована на специалистов, которые хотят освоить современные практики CI/CD и автоматизацию процессов доставки программного обеспечения с использованием Jenkins:

- Начинающие DevOps-инженеры — получат базовые знания о CI/CD и научатся строить автоматизированные пайплайны для развёртывания и сопровождения инфраструктуры.
- Смежные специалисты (системные администраторы, инженеры поддержки, специалисты по инфраструктуре) — смогут автоматизировать рутинные задачи: развёртывание серверов, резервное копирование, сканирование на уязвимости.
- Разработчики — освоят практики быстрой доставки кода, интеграции тестирования и деплоя приложений, что позволит сосредоточиться на разработке, а не на операционных задачах.
- Тестировщики — научатся автоматизировать запуск юнит-, интеграционных, нагрузочных и end-to-end тестов в Jenkins, повышая эффективность тестирования и скорость обратной связи.

### **Требования к входным знаниям обучающегося:**

Для успешного освоения программы слушателям рекомендуется:

- Основы Linux: владеть базовыми командами работы в терминале (создание файлов, каталогов, работа с процессами, настройка окружения).
- Системы контроля версий: понимать принципы работы с Git (clone, commit, push/pull, branch, merge).
- Основы разработки или администрирования: иметь общее представление о процессе сборки приложений, тестировании и развёртывании.

- Сетевые основы: знать базовые понятия (адресация, протоколы HTTP/HTTPS, DNS).
- Желательно — опыт использования виртуализации или контейнеризации (Docker), чтобы проще интегрироваться с реальными кейсами CI/CD.

#### **Актуальность реализации:**

В современных ИТ-компаниях скорость и качество поставки программного продукта напрямую зависят от автоматизации процессов сборки, тестирования и развёртывания. CI/CD-подходы позволяют снизить человеческий фактор, ускорить выпуск новых версий и повысить стабильность приложений. Jenkins остаётся одним из наиболее востребованных инструментов в области CI/CD благодаря своей гибкости, широкой экосистеме плагинов и возможности интеграции с любыми технологиями и сервисами.

#### **Отличительные особенности программы:**

- Практико-ориентированный подход: каждая тема сопровождается заданиями, которые закрепляют навыки работы с Jenkins.
- Комплексное изучение: от базовой установки и конфигурации до продвинутых тем — Pipeline, управление переменными и credentials, интеграция с Git и артефакт-менеджерами.
- Фокус на автоматизации всего цикла разработки: сборка, тестирование, развёртывание, мониторинг и отладка.
- Актуальные практики DevOps: возможность применять изученные навыки сразу в реальных проектах.

#### **Объем и срок освоения программы:** 72 академ. ч. в течение 1 мес.

Доступ к материалам Программы у обучающихся остаётся и после окончания периода обучения. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

**Выдаваемый документ о квалификации:** удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

#### **Цели и задачи программы:**

Подготовка специалистов, способных создавать и поддерживать CI/CD-процессы с использованием Jenkins, автоматизировать развёртывание приложений, управление инфраструктурой и интеграцию с другими инструментами DevOps.

#### **Программа направлена на решение следующих основных задач:**

##### **Обучающие:**

- Освоить базовые возможности Jenkins, включая установку, настройку и работу с плагинами.
- Научиться создавать и управлять Jenkins Pipeline и Jenkinsfile, работать с переменными, параметрами и credentials.
- Освоить интеграцию с системами контроля версий (Git) и артефакт-менеджерами.
- Изучить инструменты мониторинга, резервного копирования и отладки пайплайнов.

##### **Развивающие:**

- Развить навыки проектирования сложных CI/CD-процессов и оптимизации работы Jenkins.

- Научиться применять Groovy для управления логикой выполнения и переиспользования кода.
- Развить умение анализировать ошибки и повышать надёжность автоматизации.

#### **Воспитательные:**

- Формировать системное мышление при построении процессов автоматизации.
- Воспитывать ответственность за корректность и стабильность развёртывания приложений.
- Приучать к соблюдению best practices DevOps и культуре безопасной работы с инфраструктурой.

#### **Планируемые результаты:**

##### **Знания:**

- Архитектура и принципы работы Jenkins, включая распределённые агенты и плагины.
- Основы построения пайплайнов и Jenkinsfile, управление логикой выполнения.
- Интеграция Jenkins с Git, артефакт-менеджерами и внешними системами.
- Методы мониторинга, резервного копирования и отладки CI/CD-процессов.

##### **Умения:**

- Настраивать Jenkins и подключать плагины для расширения функциональности.
- Создавать и поддерживать Jenkins Pipeline с использованием Groovy, переменных и credentials.
- Подключать Git-репозитории, настраивать триггеры, webhooks и работу с тегами/release workflow.
- Управлять артефактами и интегрировать их с артефакт-менеджерами.
- Настраивать мониторинг, резервное копирование и корректное восстановление пайплайнов.

##### **Навыки:**

- Автоматизация развёртывания приложений и инфраструктуры.
- Повышение стабильности и надёжности CI/CD-процессов.
- Быстрое выявление и устранение ошибок в пайплайнах.
- Оптимизация работы Jenkins и управление нагрузкой на агенты.

#### **Перечень профессиональных компетенций, на получение которых направлено обучение:**

На основе профстандарта 06.026 «Системный администратор информационно-коммуникационных систем»:

- В/02.5 Обеспечение работы технических и программных средств информационно-коммуникационных систем;
- С/05.6 Выполнение обновления программного обеспечения сетевых устройств информационно-коммуникационных систем;
- С/08.6 Планирование и проведение работ по распределению нагрузки между имеющимися ресурсами, снятию нагрузки на сетевые устройства информационно-коммуникационных систем перед проведением регламентных работ, восстановлению штатной схемы работы в случае сбоев.

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

- ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
- ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.

## **Организационно-педагогические условия реализации программы дополнительного профессионального образования**

**Язык реализации образовательной программы:** обучение проводится на русском языке.

**Форма обучения:** заочная форма.

**Особенности реализации программы:** программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

**Условия набора:** на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

### **Формы проведения занятий:**

- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа с литературой;
- индивидуальные вопросы.

## **Материально-техническое оснащение**

### **Материальное обеспечение программы**

Занятия проводятся в системе дистанционного обучения «Rebrain». Каждый обучающийся и педагог оснащены доступом к системе дистанционного обучения: <https://rebrainme.com/>.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

### **Методическое обеспечение программы**

Программа обеспечена:

- учебно-методическими материалами (текстовые занятия, полезными материалами);
- практическими заданиями.

### **Кадровое обеспечение:**

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

- 1) отвечающее одному из требований:
  - имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;
  - имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего

образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

## 2. УЧЕБНЫЙ ПЛАН

№ п/ п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1 “Онбординг”	2	1	1	Входное тестирование
2	Модуль 2 “Основы Jenkins”	12	4	8	Практическое задание
3	Модуль 3 “Jenkins Pipeline и Jenkinsfile”	14	4	10	Практическое задание
4	Модуль 4 “Интеграция с Git и управление артефактами”	14	4	10	Практическое задание
5	Модуль 5 “Мониторинг и эксплуатация”	20	5	15	Практическое задание
6	Итоговая аттестация	10		10	Итоговое практическое задание

## 3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя	3 неделя	4 неделя
1	Модуль 1 “Онбординг”	2			
2	Модуль 2 “Основы Jenkins”	12			
3	Модуль 3 “Jenkins Pipeline и Jenkinsfile”	4	10		

4	Модуль 4 “Интеграция с Git и управление артефактами”		8	6	
5	Модуль 5 “Мониторинг и эксплуатация”			12	8
6	Итоговая аттестация				10   А

## 4. РАБОЧАЯ ПРОГРАММА

### **Модуль 1. Онбординг**

Теория 1 академ. ч. Практика 1 академ. ч.

Модуль состоит из следующих тем:

#### Тема 1: Онбординг

В модуле обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения.

Предусмотрено прохождение входного тестирования, включающего 7 вопросов, направленных на закрепление информации из онбординга. В рамках темы обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний, выбрав один из предложенных вариантов.

### **Модуль 2. Основы Jenkins**

Теория 4 академ. ч. Практика 8 академ. ч.

Модуль состоит из следующих тем:

#### Тема 1: Знакомство с Jenkins

Содержание: Что такое Jenkins. Где он используется и зачем нужен. Из чего состоит архитектура Jenkins. Как установить Jenkins на Linux и через Docker. Как выглядит интерфейс Jenkins. Практическое задание.

#### Тема 2: Плагины в Jenkins: расширение возможностей

Содержание: Как работают плагины в Jenkins. Установка и обновление плагинов. Удаление плагинов. Практическое задание.

#### Тема 3: Агенты Jenkins: распределение нагрузки

Содержание: Что такое агенты Jenkins и зачем они нужны. Настройка первого агента (SSH). Запуск задач на агенте. Практическое задание.

#### Тема 4: Первая задача с job

Содержание: Первая задача Jenkins. Freestyle и Pipeline. как Jenkins выполняет команды и работает с задачами. Практическое задание.

Модуль направлен на освоение базовой работы с Jenkins: установка и настройка сервиса, управление плагинами и агентами, создание первых задач (job). Участники научатся устанавливать Jenkins на Linux и через Docker, работать с интерфейсом, подключать и обновлять плагины, запускать задачи на распределённых агентах, а также создавать свои первые job с использованием Freestyle и Pipeline.

### **Модуль 3. Jenkins Pipeline и Jenkinsfile**

Теория 4 академ. ч. Практика 10 академ. ч.

Модуль состоит из следующих тем:

#### Тема 1: Jenkins Pipeline

Содержание: Чем отличается Freestyle от Pipeline. Какие типы Pipeline бывают: Declarative и Scripted (особенности DSL, применение). Из чего состоит структура и основные директивы Declarative Pipeline. Создание, запуск и анализ Pipeline job. Практическое задание.

#### Тема 2: Работа с переменными и параметрами и управление credentials

Содержание: Что такое переменные окружения и параметризация сборок: стандартные и динамические параметры (Active Choices). Управление секретами Jenkins Credentials. Практическое задание.

#### Тема 3: Управление логикой выполнения и пере

Содержание: Условные операторы в Groovy. Параллельное выполнение. Переиспользование кода. Практическое задание.

Модуль посвящён углублённой работе с Jenkins Pipeline и Jenkinsfile. Участники освоят типы Pipeline (Declarative и Scripted), создание и запуск Pipeline job, работу с переменными, параметрами и Jenkins Credentials. Модуль включает управление логикой выполнения с использованием условных операторов Groovy, параллельного выполнения и переиспользования кода для оптимизации пайплайнов.

### **Модуль 4. Интеграция с Git и управление артефактами**

Теория 4 академ. ч. Практика 10 академ. ч.

Модуль состоит из следующих тем:

#### Тема 1: Подключение Git-репозитория и работа с Git в разных окружениях

Содержание: Подключение Git-репозитория. Поддержка разных окружений. Практическое задание.

#### Тема 2: Работа с webhooks и триггерами. Использование Git tags и release workflow

Содержание: Настройка Jenkins на автоматический запуск при изменениях в репозитории. Работа с GitFlow и использовать теги в CI/CD. Настройка запуск пайплайна при релизе с добавлением тега. Практическое задание.

#### Тема 3: Артефакты: управление и подключение к артефакт-менеджерам

Содержание: Для чего и где хранить артефакты. Настройка архивирования и удаления старых сборок. Публикация и загрузка артефактов из внешнего хранилища. Практическое задание.

Модуль направлен на интеграцию Jenkins с системами контроля версий и артефакт-менеджерами. Участники научатся подключать Git-репозитории, настраивать работу в разных окружениях, использовать webhooks и триггеры, управлять тегами и релизами. Также изучается хранение и публикация артефактов, настройка архивирования и удаление старых сборок, подключение внешних хранилищ для артефактов.

### **Модуль 5. Мониторинг и эксплуатация**

Теория 5 академ. ч. Практика 15 академ. ч.

Модуль состоит из следующих тем:

## **Тема 1: Инструменты мониторинга**

Содержание: Как получить доступ к логам Jenkins. Настройка сбора метрик и логов. Экспорт метрик и логов в системы мониторинга. Практическое задание.

## **Тема 2: Резервное копирование**

Содержание: Уязвимость данных. Где и как хранится конфигурация Jenkins. Ручное и автоматическое резервное копирование. Тестирование восстановления Jenkins из бэкапа. Практическое задание.

## **Тема 3: Настройка retries и timeouts**

Содержание: Трудности на пайплайне. Надёжность pipeline. Конструкции retry и timeout. Практическое задание.

## **Тема 4: Анализ частых ошибок**

Содержание: Какие самые распространённые ошибки бывают в Jenkins. Как их опознать и исправить. Практическое задание.

Модуль направлен на обеспечение надёжной эксплуатации Jenkins. Участники изучат инструменты мониторинга, сбор и экспорт метрик и логов, методы резервного копирования и восстановления Jenkins. Также рассматриваются приёмы повышения надёжности пайплайнов с помощью retry и timeout, а также анализ и устранение частых ошибок при работе Jenkins.

Каждая тема модулей включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание. Практические задания рассчитаны на 2 академических часа. Выполнение заданий предполагает отправку решения на проверку через личный кабинет обучающегося. Критерии оценки прописаны в описании к каждому заданию. В случае корректного выполнения выставляется зачёт. Если работа содержит ошибки, задание возвращается на доработку. При повторной неудачной попытке (после двух доработок) обучающийся получает «незачёт».

### **Итоговая аттестация.**

Блок посвящён выполнению финального практического задания без предварительного теоретического блока.

## **5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

Программа обеспечена системой дистанционного обучения <https://rebrainme.com/>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

## **Электронно-библиотечные ресурсы и системы, информационно-справочные системы:**

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы: <https://my.rebrainme.com/course/jenkins>
3. Официальный сайт Jenkins [Электронный ресурс]: <https://www.jenkins.io/>
4. Официальный образ Jenkins для Docker [Электронный ресурс]: <https://hub.docker.com/r/jenkins/jenkins/>
5. Интерактивное руководство по работе с Jenkins [Электронный ресурс]: <https://www.jenkins.io/doc/pipeline/tour/getting-started/>
6. Официальная документация Groovy [Электронный ресурс]: <https://groovy-lang.org/documentation.html>

## **6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ**

### **Оценочные материалы:**

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

### **Текущий контроль**

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическим решиме в СДО.

### **Модуль 2 “Основы Jenkins”**

#### **Тема 2: Плагины в Jenkins: расширение возможностей**

##### **Практическое задание**

1. Используя менеджер плагинов в Jenkins, найдите и установите следующий набор плагинов, обращая внимание на их описание и возможные зависимости:
  - Основные для Pipeline: Pipeline, Git, Credentials, Credentials Binding, SSH Agent
  - Утилиты и улучшения: Timestamper, Build Timeout, Workspace Cleanup
  - Уведомления: Email Extension, Slack Notification
  - Безопасность: Matrix Authorization Strategy, Role-based Authorization Strategy (выберите один или оба для ознакомления), LDAP, Active Directory
  - Пользовательский интерфейс и визуализация: Blue Ocean, Stage View, Build Monitor View
  - Интеграции: Docker Pipeline, Active Choices, Multibranch Pipeline, GitHub Branch Source
2. Убедитесь, что все выбранные плагины успешно установлены. Может потребоваться перезапуск Jenkins для некоторых плагинов, поэтому обратите внимание на соответствующие уведомления в интерфейсе.

[...]

### **Модуль 3 “Jenkins Pipeline и Jenkinsfile”**

#### **Тема 2: Работа с переменными и параметрами и управление credentials**

##### **Практическое задание**

Ваша задача — создать параметризованный пайплайн для CI/CD приложения MyApp. Этот пайплайн будет автоматизировать процессы сборки Java-приложения, запуска тестов и деплоя в различные окружения.

### **Почему нужна параметризация?**

*Без параметров пайплайн нельзя переиспользовать: он работает одинаково для любых окружений. Без секретов — небезопасно, без переменных — неудобно. Всё должно настраиваться и защищаться.*

*Решение — использовать переменные окружения, стандартные и динамические параметры Jenkins, и credentials store для безопасного хранения секретов.*

1. Создайте новый Pipeline job с именем parametrized-job.
2. Привяжите к нему Jenkinsfile из репозитория (если используете) или вставьте код вручную в секцию Pipeline → Script.
  3. В Jenkinsfile опишите следующие параметры согласно теории:
    - APP\_VERSION (String) - строковый параметр для версии приложения
    - RUN\_TESTS (Boolean) - булев параметр для запуска тестов
    - DEPLOY\_ENV (Choice) - выбор окружения из списка ['dev', 'staging', 'prod']
    - MY\_SECRET\_TOKEN (Password) - секрет из credentials store
- [...]

## **Модуль 4 “Интеграция с Git и управление артефактами”**

### **Тема 1: Подключение Git-репозитория и работа с Git в разных окружениях**

#### **Практическое задание**

Подключите свои окружения (dev, main) и запустите каждое отдельно (в каждой ветке свой Jenkinsfile).

У вас должен быть один Git-репозиторий (вы можете создать и использовать любой репозиторий с GitLab, GitHub или BitBucket) с двумя ветками:

- main
- dev

В каждой ветке находится свой Jenkinsfile с уникальным содержимым.

Требуется:

1. Создайте две Jenkins job:
  - main\_job — подключён к ветке main
  - dev\_job — подключён к ветке dev
2. Подключите репозиторий:
  - Для main\_job — через логин/пароль (HTTPS + credentials ID)
  - Для dev\_job — через SSH (git@ + SSH credentials)
3. В Jenkinsfile каждой ветки добавьте уникальное сообщение:

Ветка main:

```
echo 'Build from MAIN branch'
```

Ветка dev:

```
echo 'Build from DEV branch'
```

У вас должно получиться:

- успешные подключения репозиториев с использованием credentials в виде логина/пароля, а также ssh-ключа;

- рабочие Jenkinsfile с вариантом использования разных окружений.

## Модуль 5 “Мониторинг и эксплуатация”

### Тема 2: Резервное копирование

#### Практическое задание

Сделайте два бэкапа: вручную через ThinBackup и автоматически скриптом.

Затем удалите директорию с плагинами из \$JENKINS\_HOME и восстановите её полученными бэкапами.

В конце убедитесь, что всё работает.

#### Итоговое оценивание

В конце программы обучающиеся сдают итоговую аттестацию.

#### Практическое задание

Представьте, вы DevOps-инженер в небольшой компании. Вам поручено настроить полноценный и безопасный CI/CD процесс для команды разработки.

**Ваша задача:** реализовать два независимых Jenkins-пайплайна. Один для ветки разработки, другой — для релизной ветки.

- Установите Jenkins (локально или в Docker), заведите учётную запись разработчика. Добавьте пользователя с именем rebrain, пароль rebrainme (нужно для проверки задания).
- Снабдите Jenkins необходимыми плагинами (найдите нужные плагины, проверьте совместимость).
- Настройте агент Jenkins и промаркируйте меткой (label). Используйте любое удобное подключение агента.
- Дайте доступ к приватному Git-репозиторию (настройте credentials для доступа).
- Подготовьте Nexus или аналогичное хранилище артефактов (можно использовать Docker Registry). Проверьте доступность хранилища с Jenkins.
- Используйте проверку образа докера на уязвимости (запуск сканера на получившийся образ).
- Настройте отправку уведомлений в Slack или Telegram (если найдены уязвимости, если сборки падают, либо успешно завершаются).

Можно использовать любой open-source сканер уязвимостей: Trivy, Grupe или другие. Образ и контейнер можно проверять на локальной машине, в Docker Desktop, Minikube и так далее. Описание пайплайна оформляется в README.md.

#### План действий

##### Пайплайн для ветки разработки (develop)

1. Сборка по событию push в репозиторий.
2. Сборка Docker-образа на основе любого Dockerfile.
3. Проверка образа на уязвимости. (При обнаружении уязвимостей пайплайн должен прерываться и отправлять уведомление разработчику).
4. Если уязвимостей нет: продолжить пайплайн. Присвоить образу любой тег (например, с хешем коммита запуска сборки). Загрузить образ в Nexus (или другое хранилище артефактов).
5. Весь пайплайн должен быть защищён таймаутом по времени сборки.

##### Пайплайн для релизной ветки (release)

1. Запуск по Git-тегу.
2. Скачивание Docker-образа, собранного на ветке разработки.

3. Смена тега образа на тот, которым помечен релизный коммит.
4. Развёртывание контейнера.
5. Через несколько минут выполнить несколько попыток проверки, что контейнер работает корректно и отправить уведомление разработчику.

Результаты текущего контроля и итогового оценивания отображаются в личном кабинете слушателя в системе дистанционного обучения <https://rebrainme.com/>.

По результатам сдачи текущего контроля, промежуточного контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.