

Маршрут вебинара

CI/CD



Jenkins jobs, views, plugins



Jenkins Job Builder



Рефлексия

Цели вебинара | После занятия вы сможете

1 Объяснить разницу между CI/CD

2 Настроить Jenkins job, pipeline, view.
Устанавливать plugins

3 Использовать Jenkins Job Builder для
управления сборками

Смысл | Зачем вам это уметь

1 Автоматизация запуска тестов и их интеграция в процесс развертывания

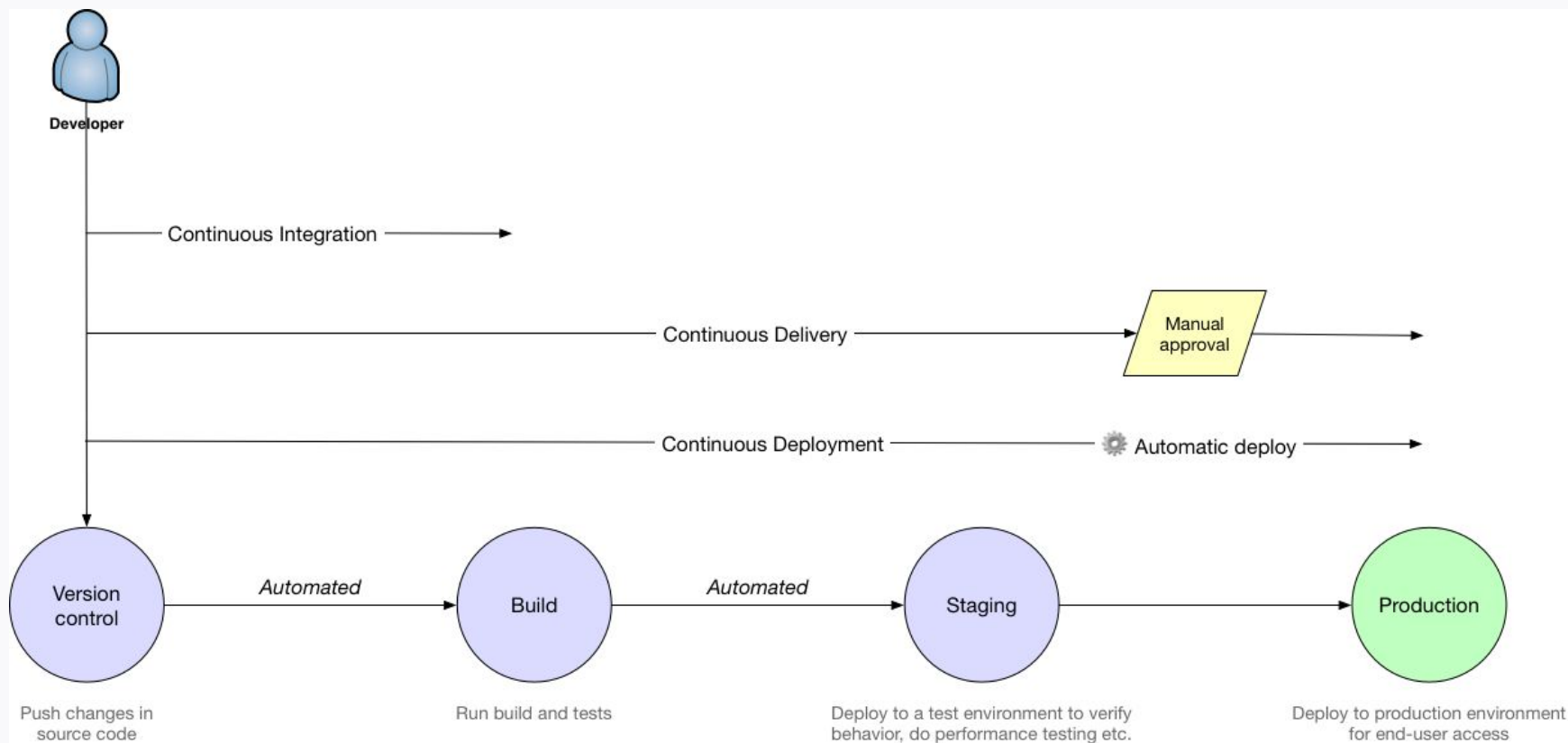
2 Возможность быстрой проверки новых изменений без участия QA

3 Хранение истории выполнения тестовых сборок

The image features a full-page background of a city skyline, likely New York City, viewed from an elevated angle. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a web-like pattern. The text "CI/CD" is centered in the middle of the image, rendered in a bold, white, sans-serif font.

CI/CD

CI/CD cxema



CI/CD уровни



Преимущества и недостатки

Преимущества

- Проблемы интеграции выявляются и исправляются быстро
- Немедленный прогон модульных тестов для свежих изменений
- Постоянное наличие текущей стабильной версии вместе с продуктами сборки — для тестирования, демонстрации, и т. п.
- Немедленный эффект от неполного или неработающего кода приучает разработчиков к работе в итеративном режиме с более коротким циклом

Недостатки

- Затраты на поддержку работы непрерывной интеграции
- Дополнительные вычислительные ресурсы под непрерывную интеграцию

The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network diagram, consisting of white dots connected by thin white lines, is superimposed on the blue layer, creating a web-like pattern across the center of the slide.

Jenkins jobs, pipelines, plugins

Jenkins

Jenkins - программная система с открытым исходным кодом на Java, предназначенная для обеспечения процесса непрерывной интеграции программного обеспечения

Позволяет автоматизировать часть процесса разработки программного обеспечения, в котором не обязательно участие человека, обеспечивая функции непрерывной интеграции

Установка Jenkins

- **Linux (Host/VM)**

<https://www.jenkins.io/doc/book/installing/linux/>

- 1) `wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key
| sudo apt-key add -`
- 2) `sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/
> \etc/apt/sources.list.d/jenkins.list'`
- 3) `sudo apt-get install jenkins`

- **Docker**

<https://hub.docker.com/r/jenkins/jenkins>

- 1) `docker pull jenkins/jenkins`
- 2) `docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts`

Job and View

Freestyle project - ключевая концепция Jenkins, с помощью которой можно настроить полный цикл сборки/тестирования/развертывания приложения

View - представление взаимосвязанных сборок

Allure integration

Как [настроить](#) генерацию Allure-отчетов в Jenkins?

1. Установить [Allure Plugin](#)
2. Выполнить настройку "Allure Commandline" в Global Tool Configuration
3. В настройках сборки добавить в Post-build Actions добавить Allure Report
4. В настройках Allure Report указать директорию, в котором хранятся данные, сгенерированные адаптером для используемого тестового фреймворка, например, [allure-pytest](#)
5. После завершения сборки - посмотреть Allure-отчет в разделе меню Allure Report

Pipeline

Pipeline - описание процессов сборки/тестирования/развертывания приложения по стадиям

Jenkinsfile - текстовый файл с описанием pipeline, который может быть сохранен в репозитории проекта

Синтаксис Jenkinsfile

- **Declarative**
- **Scripted**

Элементы Pipeline

Pipeline - код, который определяет CI/CD процессы, которые включают в себя сборку приложения, тестирование и развертывание

Node - машина, которая является частью Jenkins-окружения и на которой может выполняться pipeline

Stage - блок, определяющий набор подзадач, которые выполняются в рамках pipeline, например, "Build", "Test" и "Deploy"

Step - отдельная задача, которая сообщает Jenkins, что нужно делать в конкретный момент времени

Declarative Pipeline

Использует упрощенный синтаксис для описания pipeline

Менее гибкий по сравнению с Scripted Pipeline

Подходит для начинающих пользователей, которым нужно создавать относительно простые pipelines

`${YOUR_JENKINS_URL}/directive-generator/` - генератор сниппетов для Declarative Pipeline

Scripted Pipeline

Использует DSL основанный на Groovy

Обладает максимальной гибкостью

Подходит для продвинутых пользователей, которым нужно создавать более сложные pipelines

`${YOUR_JENKINS_URL}/pipeline-syntax` - генератор сниппетов для Scripted Pipeline

<https://www.jenkins.io/doc/book/pipeline/#pipeline-example>

Plugins

- [Multiselect parameter](#) - работа с взаимосвязанными параметрами
- [Git Parameter](#) - выбор ветки репозиторий для сборки
- [Copy Artifact](#) - копирование артефактов из другой сборки
- [Test Results Analyzer](#) - анализ результатов тестов
- [Build Pipeline](#) - удобное отображение upstream/downstream сборок
- [Git Changelog](#) - отображение информации о коммите
- [Docker](#) - Docker-контейнеры в качестве executors
- [Email Extension](#) - гибкая настройка отправки email
- [Job Configuration History](#) - изменения в настройках сборки

The background of the image is an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a network of white lines and dots, resembling a web or data structure. The text "Jenkins Job Builder" is centered in the middle of the image, in a white, bold, sans-serif font.

Jenkins Job Builder

Jenkins Job Builder

Jenkins Job Builder - утилита, которая с помощью описаний сборок в форматах YAML или JSON позволяет настраивать сборки на Jenkins-сервере

Описания сборок можно хранить в репозитории, что позволяет удобно вносить и отслеживать изменения в настройках сборок

В Jenkins Job Builder поддерживаются шаблоны, что позволяет легко создавать описания для нескольких однотипных сборок

Setup and Configuration

Установка

```
pip install --user jenkins-job-builder
```

Конфигурационный файл

`jenkins_jobs.ini` - файл для настройки взаимодействия с Jenkins-сервером

Running

- **Test Mode**

```
jenkins-jobs test ci/foo.yaml
```

```
jenkins-jobs test -r ci
```

- **Updating jobs**

```
jenkins-jobs update ci/foo.yaml
```

```
jenkins-jobs update ci/foo.yaml Foo1 Foo2
```


The background of the slide is a high-angle, aerial photograph of a dense urban skyline, likely New York City, featuring numerous skyscrapers. The image is tinted with a blue and teal color palette. A semi-transparent network of white lines and dots is overlaid on the image, creating a digital or technological feel. The word "Рефлексия" is centered in the middle of the slide in a large, white, sans-serif font.

Рефлексия

Цели вебинара | После занятия вы сможете

1 Объяснить разницу между CI/CD

2 Настроить Jenkins job, pipeline, view.
Устанавливать plugins

3 Использовать Jenkins Job Builder для
управления сборками

Материалы

Official Docs


- [Jenkins](#)
- [Jenkins Job Builder](#)

YouTube

- [Get started with Jenkins](#)

Articles

- [Using Docker Containers as Jenkins Build Slaves](#)
- [How to enable docker remote API on docker host \(Linux\)](#)
- [Enabling Docker Remote API On OSX](#)



Заполните, пожалуйста,
опрос о занятии по ссылке в чате



ОНЛАЙН ОБРАЗОВАНИЕ

Непрерывная интеграция, Jenkins



Андрей Гридяев

Ведущий инженер по автоматизации тестирования

andrey.gridyaev@gmail.com