

Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«Национальный исследовательский университет  
ИТМО»

Факультет Информационных технологий и программирования

Настройка CI/CD для проекта

Выполнил: Николаев Алексей Иванович М3111

Проверил:Повышев Владислав Вячеславович

Санкт-Петербург

2022 г.

## 1) Скачиваем TeamCity с сайта JetBrains:

The screenshot shows the TeamCity website homepage. At the top, there's a navigation bar with links: "What's New 2022.10", "Roadmap", "Features", "Learn", "Request Info", "Pricing", and a "Start building for free" button. The main header features the TeamCity logo and the tagline "Powerful continuous integration for DevOps-centric teams". Below this, there's a section titled "TeamCity 2022.10 is here! Check out what's new." and three columns of product offerings:

- TeamCity Cloud** (Hosted by JetBrains):
  - Unlimited web users
  - Unlimited concurrency
  - Fleet of Linux, Windows and macOS build agents maintained by JetBrains
  - Supports adding your own build agents
  - Try free for 14 days, then from \$45 per month
- TeamCity Professional** (Hosted by you):
  - Unlimited users
  - Unlimited build time
  - 100 build configurations
  - 3 build agents included
  - All features of TeamCity Enterprise
  - Free forever, even for commercial use
- TeamCity Enterprise**:
  - CI/CD that works at scale where others fail
  - Unlimited users
  - Unlimited build time
  - Unlimited build configurations
  - Priority support
  - From \$1,999 per year, with a 50% renewal discount

2) Запускаем через консоль TeamCity и потом берем ссылку из файла `readme` и заходим на локальный сервер TeamCity

3) Работа с локальным сервером:

Создаем проект в TeamCity:

The screenshot shows the TeamCity Administration console with the "Create Project From URL" form. The breadcrumb is "Administration / <Root project>". A green message states: "The connection to the VCS repository has been verified". The form fields are as follows:

- Project name:** LAB6 DevTools
- Build configuration name:** Build
- VCS root:** (Git) `https://github.com/Vu1cany/LAB6_DevTools`
- Default branch:** `refs/heads/main` (The main branch or tag to be monitored)
- Branch specification:** `refs/heads/*` (Edit branch specification: `refs/heads/*`)

Below the form, there's a note: "Branches to monitor besides the default one as a newline-delimited set of rules in the form of `+|-:branch name` with the optional `*` placeholder. Part matched by `*` (or part inside parentheses) will be shown as a build logical branch name." At the bottom, there are "Proceed" and "Cancel" buttons.

## Настраиваем Build:

Administration / <Root project> / LAB6 DevTools

☐ Build ⚠ 1 warning

General Settings

Version Control Settings 1

Build Steps 2

Triggers 1

Failure Conditions

Build Features 1

Dependencies

Parameters

Agent Requirements

Suggestions 1

« Hide unconfigured

Last edited 19 hours ago by vu1cany (view history)

View as code ?

Name: \*

Build

Build configuration ID: \*

Lab6DevTools\_Build

Regenerate ID

Description:

Publish artifacts: ?

Only if build status is successful

Reset

Artifact paths: ?

Console\_Calculator/cmake-build-debug => Console\_Cal

Reset

Newline- or comma-separated paths in the form of [+:]source [ => target] to include and -:source [ => target] to exclude files or directories to publish as build artifacts. Ant-style wildcards are supported, e.g. use \*\*/\* => target\_directory, -: \*\*/folder1 => target\_directory to publish all files except for folder1 into the target\_directory.

Show advanced options

Save Cancel

## Шаги сборки:

Build Step (1 of 2): test | ▾

Simple command execution Change runner

Step name:

test

Optional, specify to distinguish this build step from other steps.

Execute step: ?

If all previous steps finished successfully

Add condition ▾ ?

Working directory: ?

Optional, set if differs from the checkout directory.

Run:

Executable with parameters

Command executable: \*

Tests\_for\_Console\_Calculator/cmake-build-debug/Tests

Command parameters:

Format stderr output as:

warning

Specify how error output is processed.

Docker Settings

Run step within Docker container: ?

E.g. ruby:2.4. TeamCity will start a container from the specified image and will try to run this build step within this container. ?

Hide advanced options

Save Cancel

## Build Step (2 of 2): Build exec | ▾

Simple command execution [Change runner](#)

**Step name:**

Optional, specify to distinguish this build step from other steps.

**Execute step:**  [Add condition](#) [?](#)

**Working directory:**  [?](#)

Optional, set if differs from the checkout directory.

**Run:**

**Custom script:** \* Enter build script content:

```
1 cmake ..  
2 make
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

**Format stderr output as:**  [?](#)  
Specify how error output is processed.

### Docker Settings

**Run step within Docker container:**

E.g. ruby:2.4. TeamCity will start a container from the specified image and will try to run this build step within this container. [?](#)

[Hide advanced options](#)

Save

Cancel

4)Теперь по событию комита на gitHub прогоняются все тесты для проекта в нашей директории gitHub и создается исполняемый файл сборки и все происходит автоматически без участия разработчиков

## LAB6 DevTools ★

[Configure favorites...](#)

**Overview** [Investigations](#) [Change Log](#) [Statistics](#) [Current Problems](#) [Muted Problems](#) [Flaky Tests 0](#)

[Builds](#) [Trends](#) [Collapse All](#)

### Build

#19 ★

main

✓ Success

No changes

🍏 Default Agent

about 20 hours ago 10s

[Run](#) [...](#) [...](#)

## Вывод:

В данной лабораторной работе научились настраивать TeamCity для CI/CD разработки проекта, полностью автоматизировали процесс прогона тестов и упаковки в исполняемый файл по событию комита.