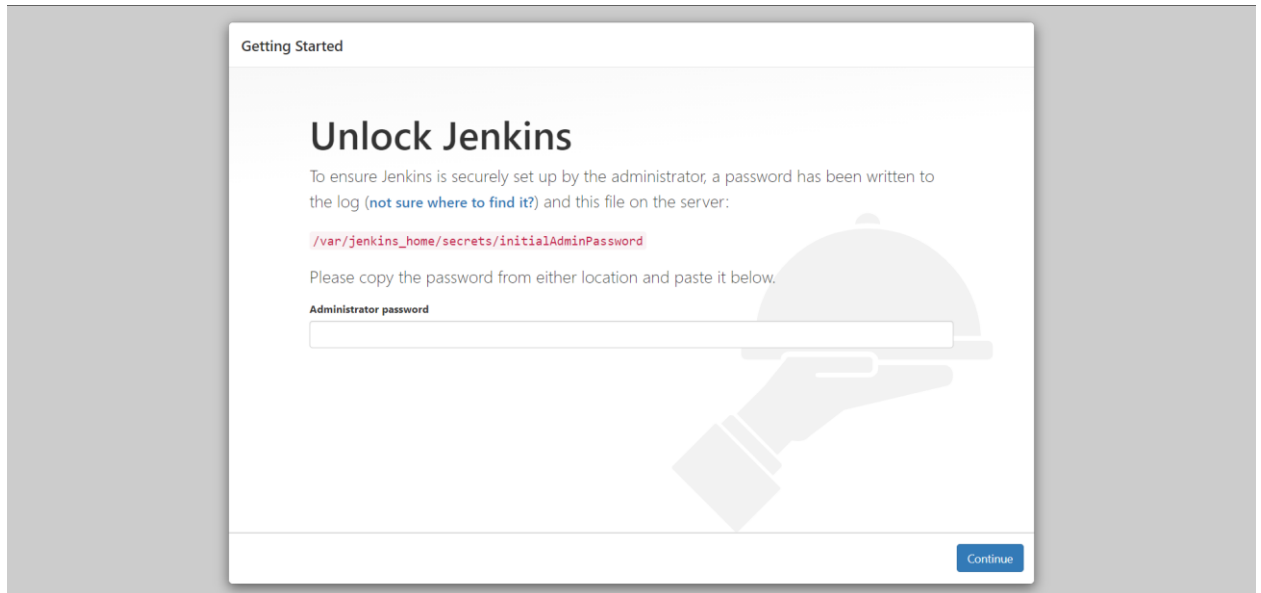


Отчёт “Настройка и интеграция сервисов *Taiga* и *Jenkins*” по курсу  
“Администрирование ОС Unix”.  
**Jenkins.**

Перейдя по <http://192.168.0.1:8080/> (где 192.168.0.1 – ip-адрес виртуальной машины) открывается стартовая страница Jenkins для первого входа:



На данном этапе необходимо разблокировать Jenkins, для этого нужно ввести секретный пароль администратора. Он хранится в /jenkins/secret/initialAdminPassword. Копируем пароль и вставляем.

Теперь необходимо установить плагины. Рекомендуется выбрать установку предлагаемых плагинов (Install suggested plugins):



Далее необходимо создать пользователя с именем admin:

Getting Started

## Create First Admin User

Имя пользователя:

admin

Пароль:

\*\*\*\*\*

Повторите пароль:

\*\*\*\*\*

Ф.И.О.:

admin

Адрес электронной почты:


admin@example.com

Jenkins 2.289.1




[Skip and continue as admin](#)

[Save and Continue](#)

Теперь нас приветствует Jenkins!

 Jenkins

ПОИСК

  admin  выход

Dashboard

Создать Item

Пользователи

История сборок

Настроить Jenkins

My Views

Lockable Resources

New View

Очередь сборки

Очередь сборки пуста

Состояние сборщиков

1 В ожидании

2 В ожидании

добавить описание

## Добро пожаловать в Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

REST API

Jenkins 2.289.1

Далее выбрать “Создать Item”, ввести название и выбрать тип “Pipeline”. Затем в открывшемся окне поставить галку у поля “GitHub project” и вставить URL-адрес проекта (<https://github.com/buchakucha/success-pipeline>):

Dashboard > test

General Build Triggers Advanced Project Options Pipeline

Описание

[Plain text] Предпросмотр

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url

https://github.com/buchakucha/success-pipeline

Расширенные...

☐ Pipeline speed/durability override

☐ Preserve stashes from completed builds

☐ Throttle builds

☐ Удалить устаревшие сборки

☐ Это - параметризованная сборка

Build Triggers

Сохранить Применить

Затем в разделе “Pipeline”->“Definition” выбрать “Pipeline script from SCM”. В роли SCM-сервиса выбрать “Git”, вновь вставить ссылку на репозиторий, не забыть поменять в разделе “Branches to build” ветку \*/master на \*/main (GitHub изменил название ветки) и нажать сохранить:

Dashboard > test

General Build Triggers Advanced Project Options Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/buchakucha/success-pipeline

Credentials

none

Расширенные...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\*/main

Сохранить Применить Add Branch

Так как код в репозитории написан на JavaScript необходимо установить соответствующий плагин, чтобы сборка не проваливалась. Для этого выбрать “Настроить Jenkins”->“Управление плагинами” и во вкладке “Доступные” в фильтр ввести “nodejs” и поставить галку на установке, затем “Install without restart”.

The screenshot shows the Jenkins Dashboard with the 'Управление плагинами' (Manage Plugins) page. The 'Доступные' (Available) tab is selected, showing a table of available plugins. The 'NodeJS' plugin is listed with version 1.4.0, released 3 months and 12 days ago. Below the table, there are buttons for 'Install without restart', 'Download now and install after restart', and 'Проверить сейчас' (Check now). The bottom right corner shows 'REST API' and 'Jenkins 2.289.1'.

Install	Name	Version	Released
<input type="checkbox"/>	NodeJS npm NodeJS Plugin executes NodeJS script as a build step.	1.4.0	3 месяца 12 дней ago

Buttons: Install without restart, Download now and install after restart, Обновлено: 1 час 0 минут назад, Проверить сейчас

Когда установка закончится выбрать “Restart Jenkins when installation is complete and no jobs are running” и ждать, пока Jenkins не закончит перезагрузку.

После перезагрузки необходимо снова войти в систему, затем “Настройки Jenkins”->“Конфигурация глобальных инструментов”, спуститься до раздела NodeJS и указать параметры, как на скриншоте:

The screenshot shows the 'NodeJS' configuration page under 'Конфигурация глобальных инструментов'. The 'NodeJS' section is expanded, showing the 'Name' field set to 'nodejs'. The 'Install automatically' checkbox is checked. Under 'Install from nodejs.org', the 'Version' is set to 'NodeJS 12.0.0'. There are checkboxes for 'Force 32bit architecture' and 'Global npm packages to install'. The 'Global npm packages refresh hours' is set to '72'. At the bottom, there are 'Save' and 'Apply' buttons.

NodeJS

NodeJS установок

Добавить NodeJS

NodeJS

Имя

nodejs

☒ Install automatically

☐ Install from nodejs.org

Версия

NodeJS 12.0.0

☐ Force 32bit architecture

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Global npm packages refresh hours

72

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

Save Apply

Вернёмся к нашей сборке и в качестве проверки запустим Jenkins (“Собрать сейчас”). Сборка прошла успешно:

**Jenkins**

Dashboard > test

Back to Dashboard

Status

Changes

Собрать сейчас

Настройки

Удалить Pipeline

Full Stage View

GitHub

Rename

Pipeline Syntax

История сборок

Тренд

find

#3 Jun 24 17:44 No Changes

Average stage times:  
(Average full run time: ~2min 32s)

**Pipeline test**

Recent Changes

добавить описание

Выключить проект

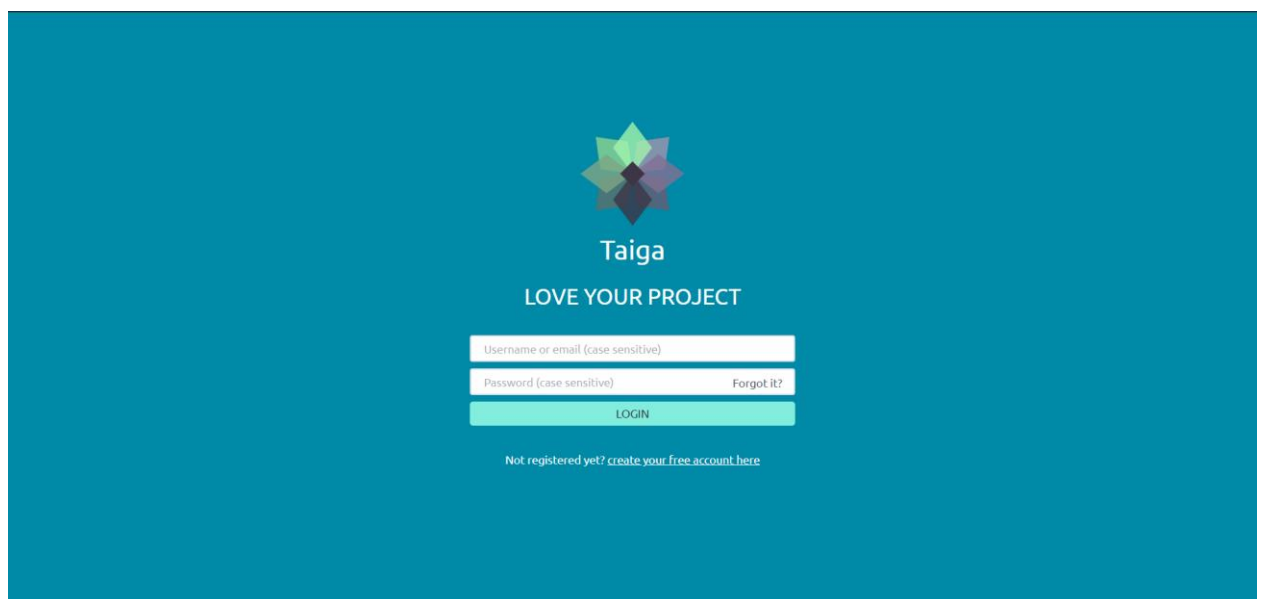
Declarative: Checkout SCM	Declarative: Tool Install	Build	Test	Deploy
4s	2min 2s	2s	2s	1s

**Постоянные ссылки**

- Последняя сборка (#3), 5 минуты 55 секунд назад
- Последняя стабильная сборка (#3), 5 минуты 55 секунд назад
- Последняя успешная сборка (#3), 5 минуты 55 секунд назад
- Last completed build (#3), 5 минуты 55 секунд назад

Taiga.

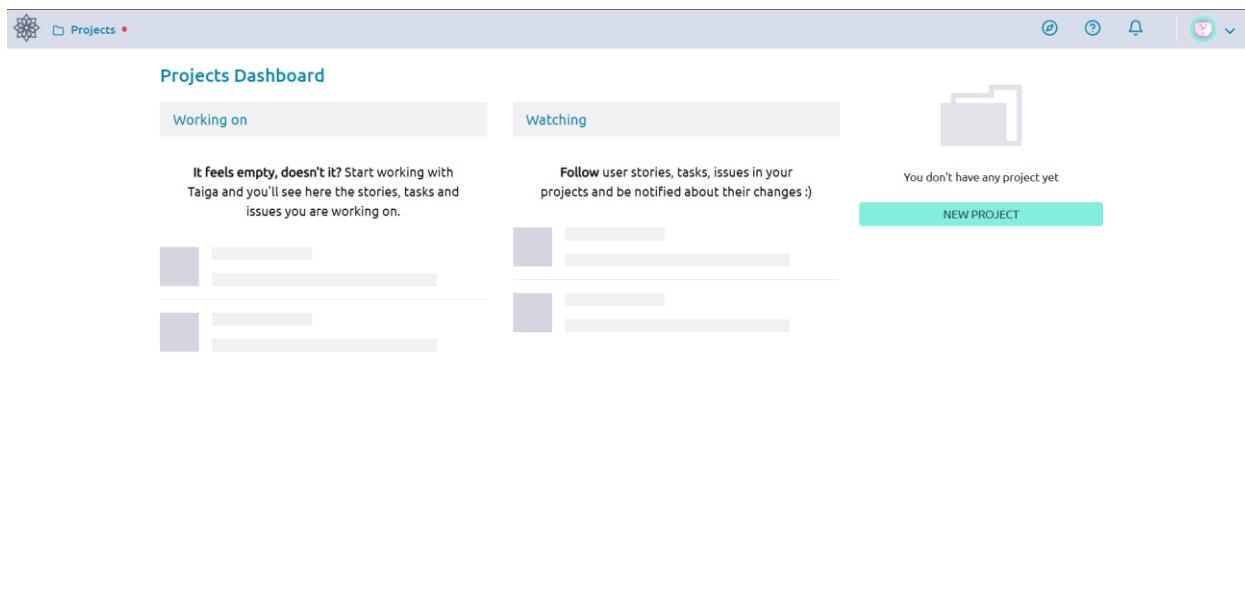
Перейдя по <http://192.168.0.1:9000/> (где 192.168.0.1 – ip-адрес виртуальной машины) открывается Taiga с приглашением ко входу:



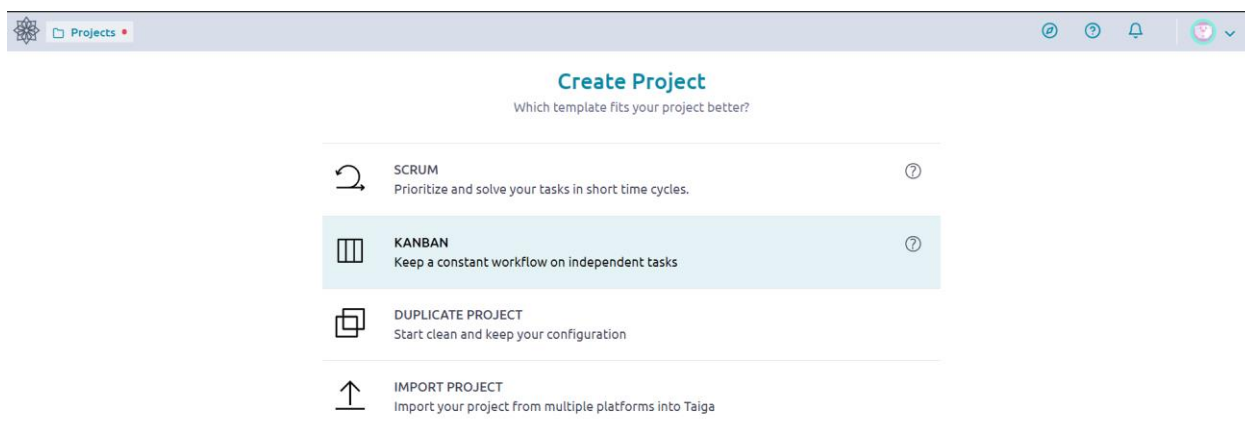
Войти в приложение пока невозможно: не существует ни одной учётной записи, поэтому необходимо создать суперпользователя. Для этого следует выполнить команду в корне проекта `docker-compose -f docker-compose.yml -f docker-compose-inits.yml run --rm taiga-manage createsuperuser` и ввести запрашиваемые данные:

```
Username: admin
Email address: admin@example.com
Password:
Password (again):
Superuser created successfully.
```

Теперь можно войти в Taiga по этим данным:



Далее нужно создать проект. В верхнем левом углу навести на “Projects”, нажать на “New Project” и выбрать шаблон “KANBAN”:



Ввести необходимые данные и создать проект:

Projects

Kanban

Keep a constant workflow on independent tasks

New project details

my\_project

My first project

PUBLIC PROJECT PRIVATE PROJECT

BACK CREATE PROJECT

## Интеграция Jenkins и Taiga.

Чтобы продемонстрировать взаимодействие двух сервисов, первым делом необходимо создать карточку (User Story) в созданном проекте в Taiga. Выбрать в колонке “New” знак плюс, добавить название карточки, описание заполнить в виде “API\_TOKEN\_JENKINS” “НАЗВАНИЕ\_PIPELINE”.

Для того, чтобы получить API Token, необходимо перейти в Jenkins, нажать на “admin” в правом верхнем углу, затем “Настроить”. В модуле “API Token” выбрать “Add new token”, ввести название, затем нажать “Generate”:

Jenkins

Dashboard admin

Пользователи

Статус

Сборки

Настроить

My Views

Credentials

Full name

admin

Description

API Token

Current token(s)

my\_token 1199df2fb9cd32eef75280d92a83b76cd3

⚠ Copy this token now, because it cannot be recovered in the future.

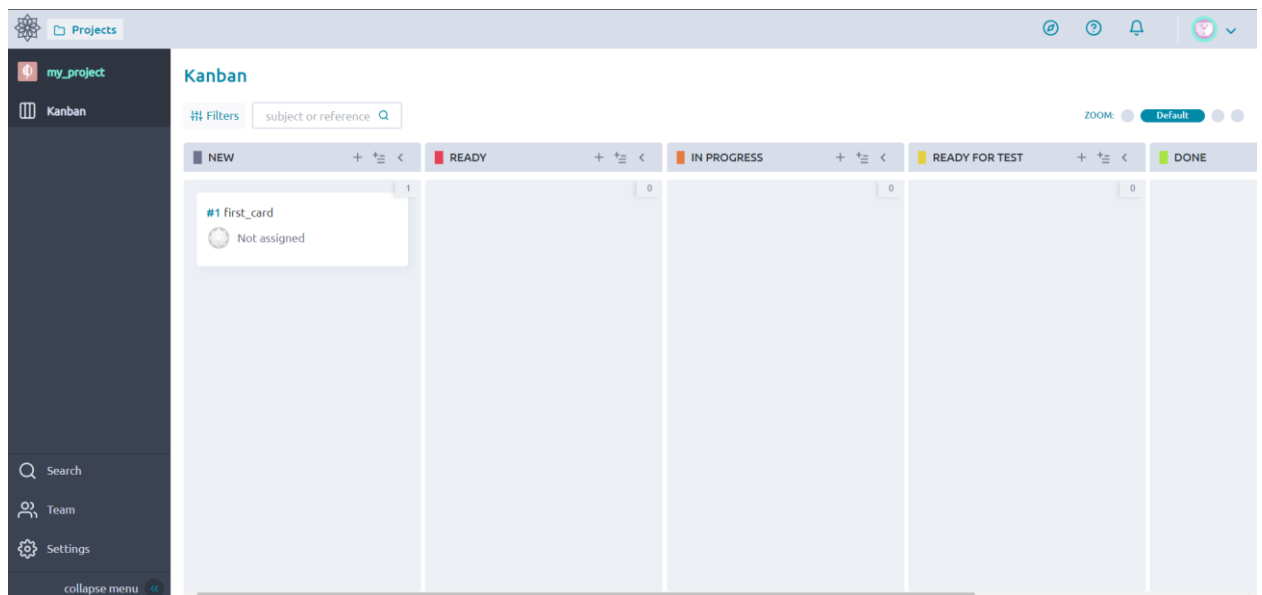
Add new Token

Credentials

Credentials are only available to the user they belong to

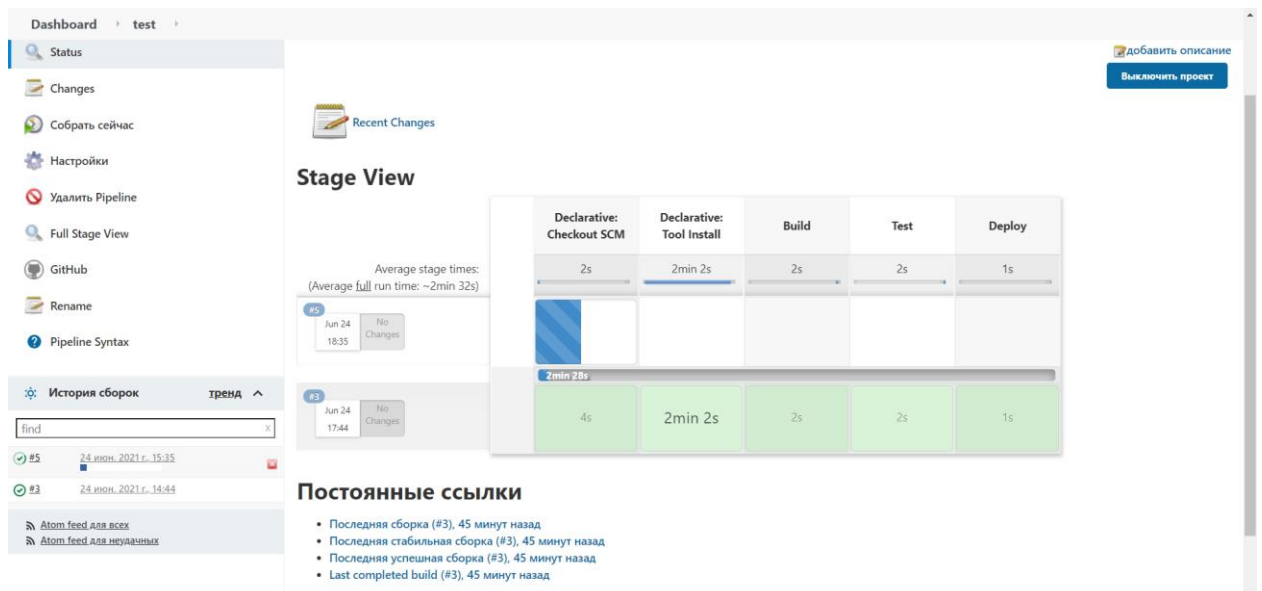
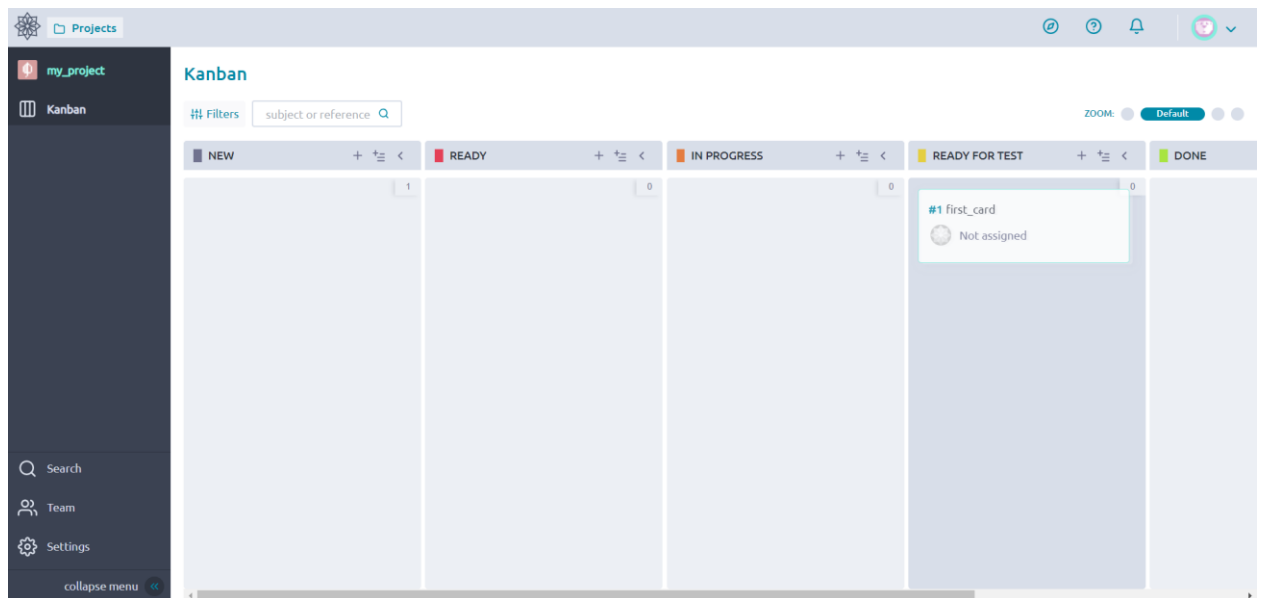
Сохранить Применить

Скопировать полученный API Token и добавить в описание карточки в формате, указанном выше, сохранить:



Теперь, если перетащить карточку в колонку “READY FOR TEST”, автоматически запустится сборка Jenkins:





Скрипт проверяет изменение в колонке каждую минуту, сравнивая массив карточек с прошлой минутой: если добавились новые - запускает сборки, удалились/не добавились - продолжает следить за изменениями.