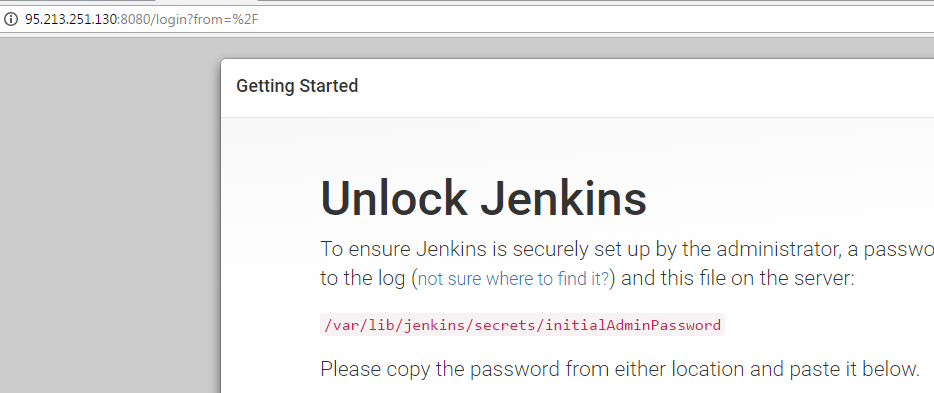
# Установка Jenkins

Скачиваем установочный пакет под нужную ОС с официального сайта: <https://jenkins.io/index.html>

Запускаем, устанавливаем всё по умолчанию.

После установки Jenkins будет доступен из браузера по адресу: localhost:8080. Должен появиться приветственный экран Jenkins:



*Рисунок 1*

Если экран появился, можно переходить к [первоначальным настройкам](#_Первоначальная_настройка_и) системы. В противном случает придётся еще [побороться с закрытым портом 8080](#_Решение_проблемы_занятого).

Здесь описан порядок подключения на локальной машине. В общем случае, в адресной строке браузера следует писать IP сервера, на котором установлен Jenkins: http://[IP-адрес сервера]:8080

Про более сложную распределенную установку Jenkins на сервере/клиентах написано здесь: <https://www.ibm.com/developerworks/ru/library/d-continuous-delivery-framework-jenkins/>

## Другие способы установки Jenkins

В описаниях на разных сайтах рассказывается про другие способы установки. Самый распространённый из них - скачать установочный пакет java по ссылке:

<http://mirrors.jenkins-ci.org/war/latest/jenkins.war>

и выполнить команду:

java –jar jenkins.war

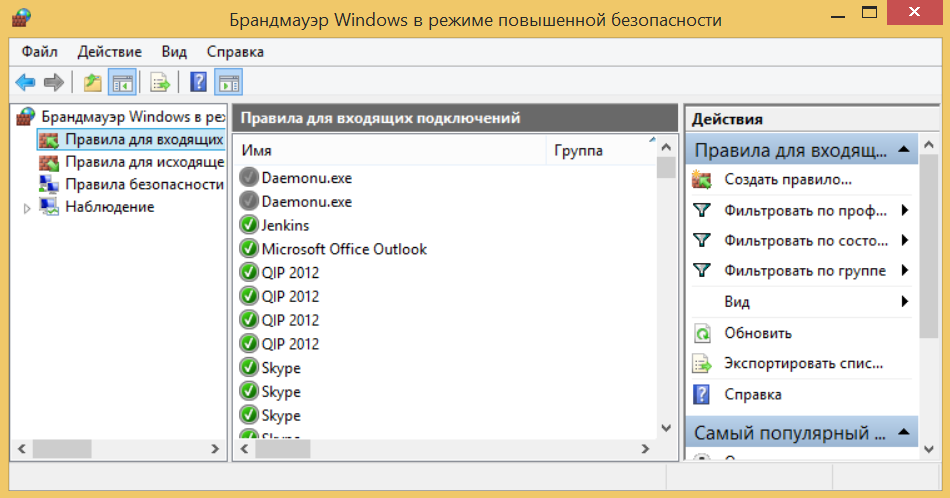
Подробнее о способах установки, их плюсах и минусах можно почитать здесь: <http://automation-remarks.com/tri-sposoba-podniat-jenkins-ci-dlia-vashikh-avtotiestov/>

## Решение проблемы занятого порта 8080

Если после установки Jenkins в браузере появляется что угодно, но только не то, что изображено на рисунке 1, значит порт 8080 на вашем компьютере закрыт или занят другим приложением (PS а заодно проверьте статус работы сервиса Jenkins и наличие оперативной памяти, Jenkins потребляем от 256Мб до 1Гб оперативы). Решения проблемы два: либо открыть порт средствами Windows, либо переопределить номер порта для Jenkins.

### Как открыть порт в брандмауэре Windows

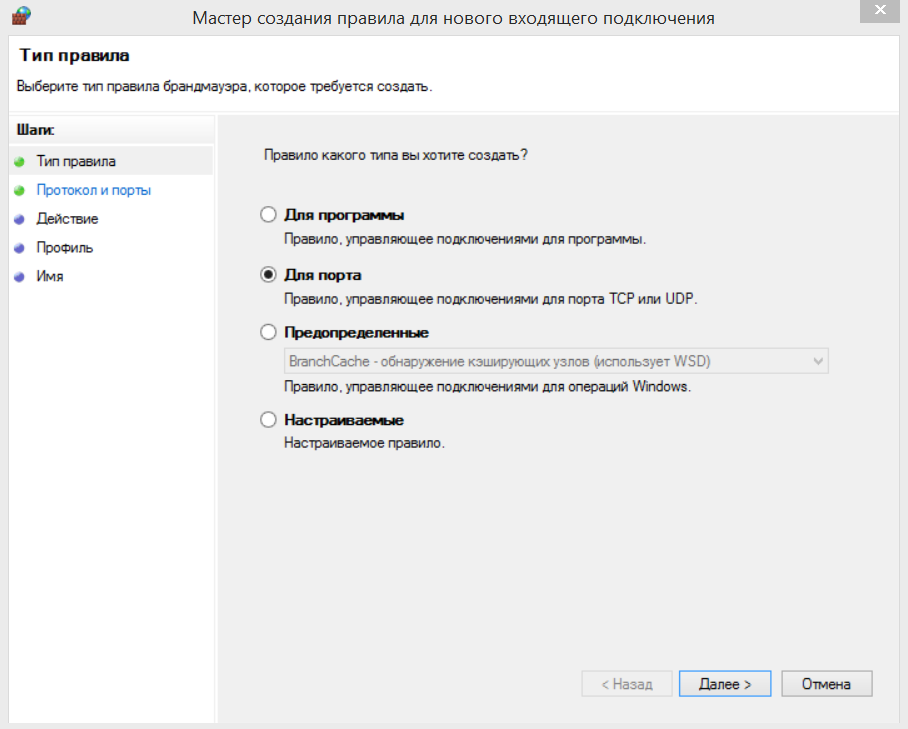
Открываем брандмауэр: нажимаем [Win+R], даём команду firewall.cpl. В открывшемся окне переходим в «Дополнительные параметры», затем открываем раздел «Правила для входящих подключений». Должно появиться окно:



*Рисунок 2*

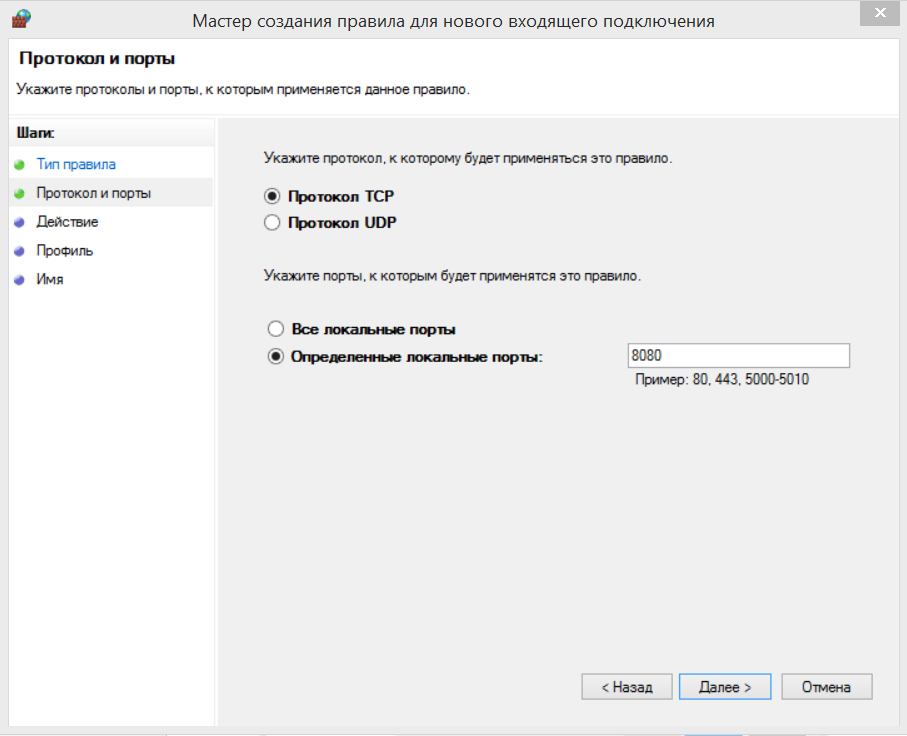
В правой колонке нажимаем «Создать правило…».

Шаг 1. Выбираем тип правила «Для порта» и нажимаем «Далее».



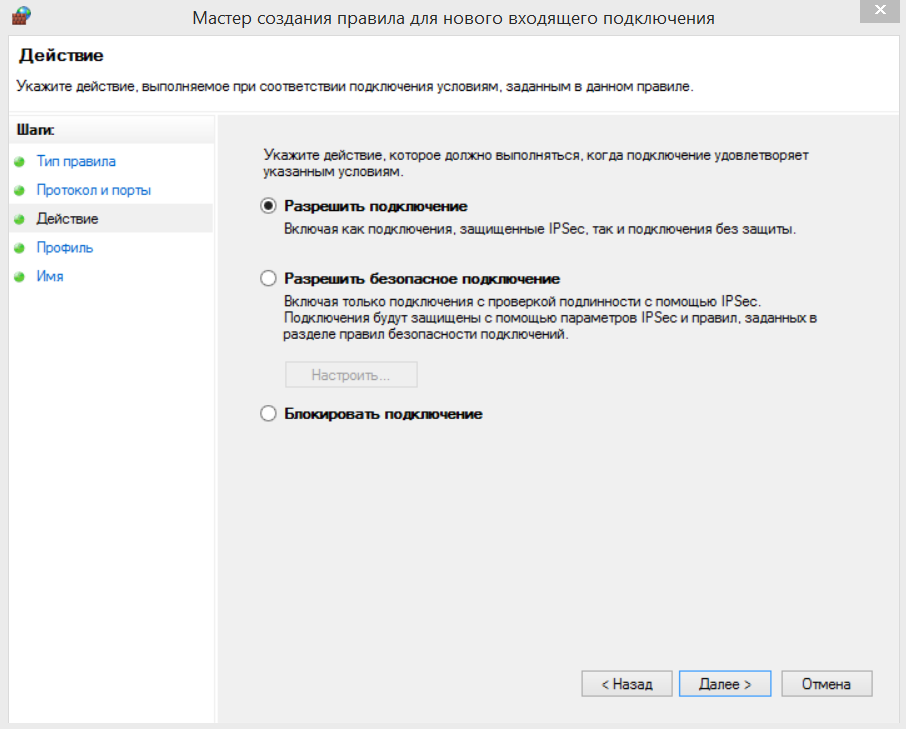
*Рисунок 3*

Шаг 2. В поле «Определенные локальные порты» вводим номер нашего порта 8080, и нажимаем «Далее»



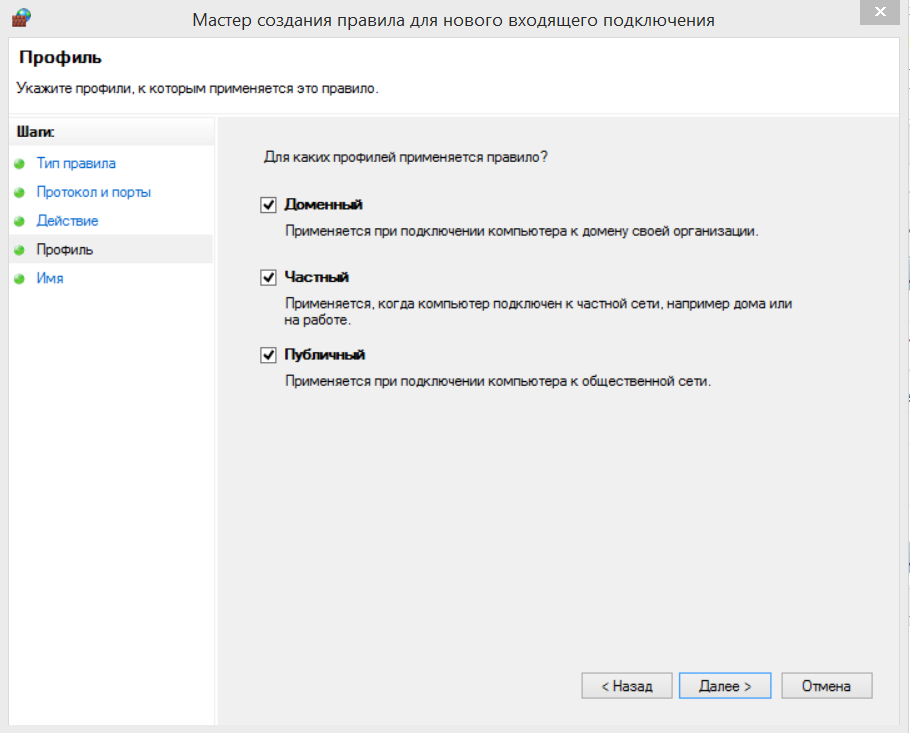
*Рисунок 4*

Шаг 3. Выбираем вариант «Разрешить подключение» и жмём «Далее»



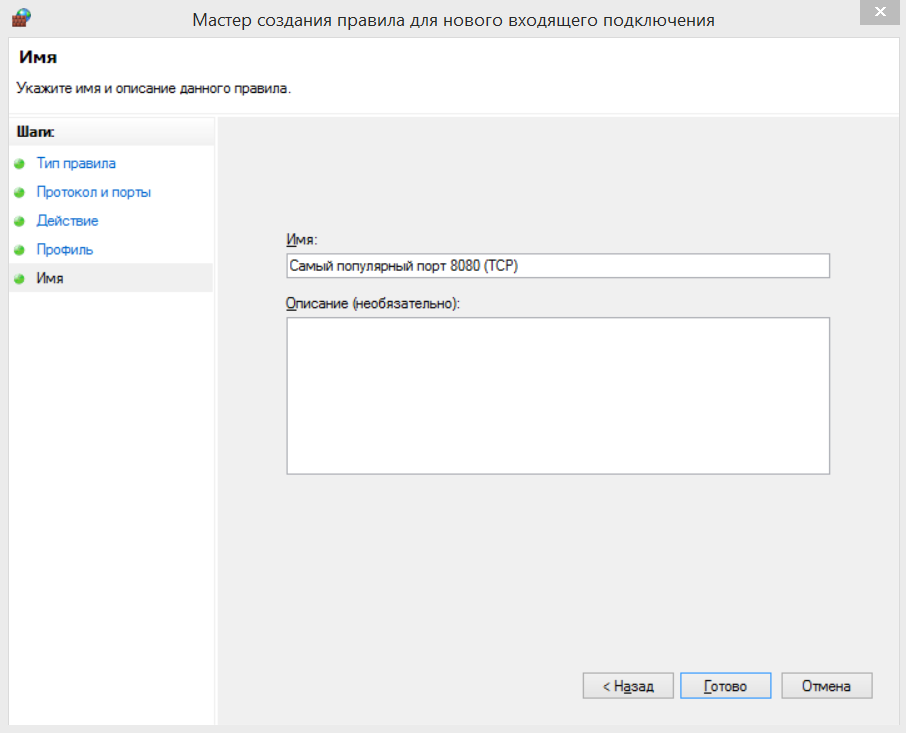
*Рисунок 5*

Шаг 4. Оставляем всё по умолчанию и нажимаем «Далее»



*Рисунок 6*

Шаг 5. На заключительном шаге указываем имя для нового правила файервола и нажимаем «Готово».



*Рисунок 7*

Если и после этого Jenkins не запустится, добавьте еще одно такое же правило, только на шаге 2 укажите правило UDP.

### Как переопределить рабочий порт Jenkins

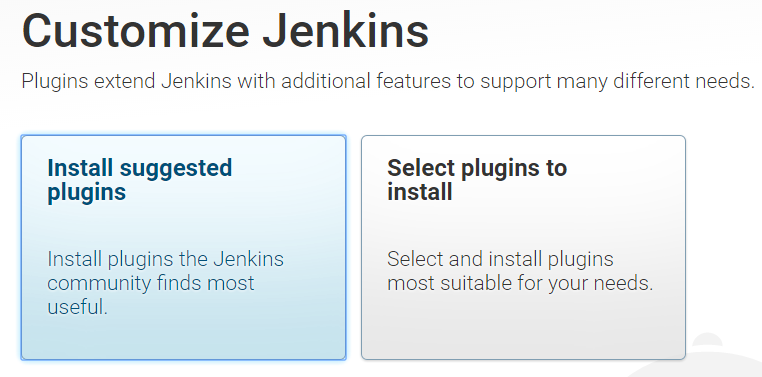
Чтобы Jenkins подключался к другому порту, в файле \Program Files\Jenkins\jenkins.xml нужно найти ключ:

--httpPort=8080

и поменять номер на желаемый, после чего перезапустить сервис Jenkins.

## Первоначальная настройка и установка плагинов

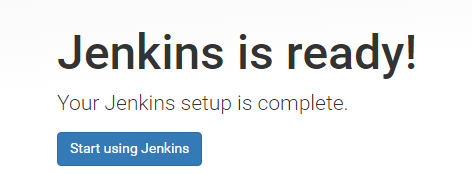
После того, как установка выполнена, в браузере мы попадаем в окно приветствия (см. Рисунок 1). Здесь нужно вставить пароль из файла \Program Files\Jenkins\secrets\initialAdminPassword и нажат «Продолжить». На следующем шаге выбираем режим установки плагинов: Install suggested plugins.



*Рисунок 8*

На первых порах нам достаточно будет одного плагина Git plugin, его и устанавливаем.

После установки создаём профиль администратора, имя пользователя и пароль запоминаем для использования в дальнейшей работе. Нажимаем «Продолжить», на экране появится сообщение, свидетельствующее о готовности Jenkins к работе.



*Рисунок 9*

Нажимаем «Start using Jenkins» и переходим в рабочую панель системы.

**Дополнительно**

В дальнейшем, при необходимости установки новых плагином или удалении ранее установленных следует зайти в «Настроить Jenkins – Управление плагинами – Доступные» («Manage Jenkins – Manage Plugins - Available»).

Панель управления пользователями находится в «Настроить Jenkins – Manage Users» («Manage Jenkins – Manage Users»). Если при установке не определить нового пользователя, то по умолчанию работа будет продолжена от имени **jenkins**, либо **admin** – какой именно заведётся у вас, предположить трудно, не понятно, отчего это зависит, но полагаю, что от версии Windows или самого Jenkins.

# Настройка проекта в Jenkins

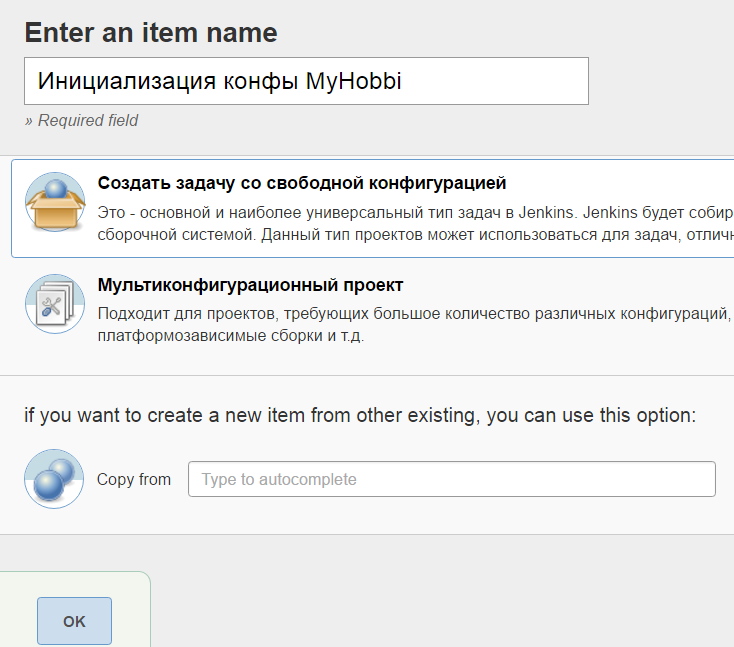
## Создание простой задачи

Теперь, когда Jenkins настроен и готов к работе, создадим наш первый сценарий. Это еще не боевая задача, но на ее примере мы увидим базовые возможности Jenkins.

Первый сценарий будет выполнять первоначальную настройку репозитория Git под проект 1С. К этому времени предварительно должно быть выполнено:

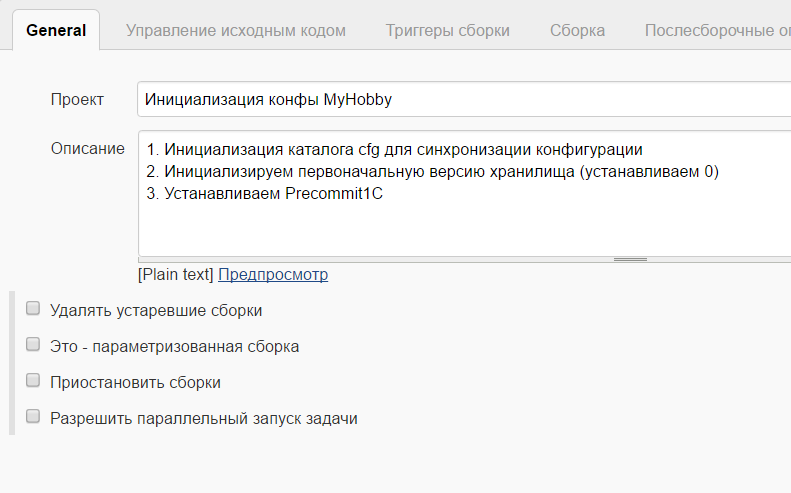
1. Создан пустой локальный репозиторий Git
2. В каталоге репозитория создан пустой подкаталог cfg
3. Настроено хранилище 1С

Итак, для создания нового сценария выбираем на главной странице Jenkins команду «Создать Item» («New Job»). Задаём название нашему проекту, выбираем тип «Создать задачу со свободной конфигурацией» («Free-style project») и нажимаем OK.



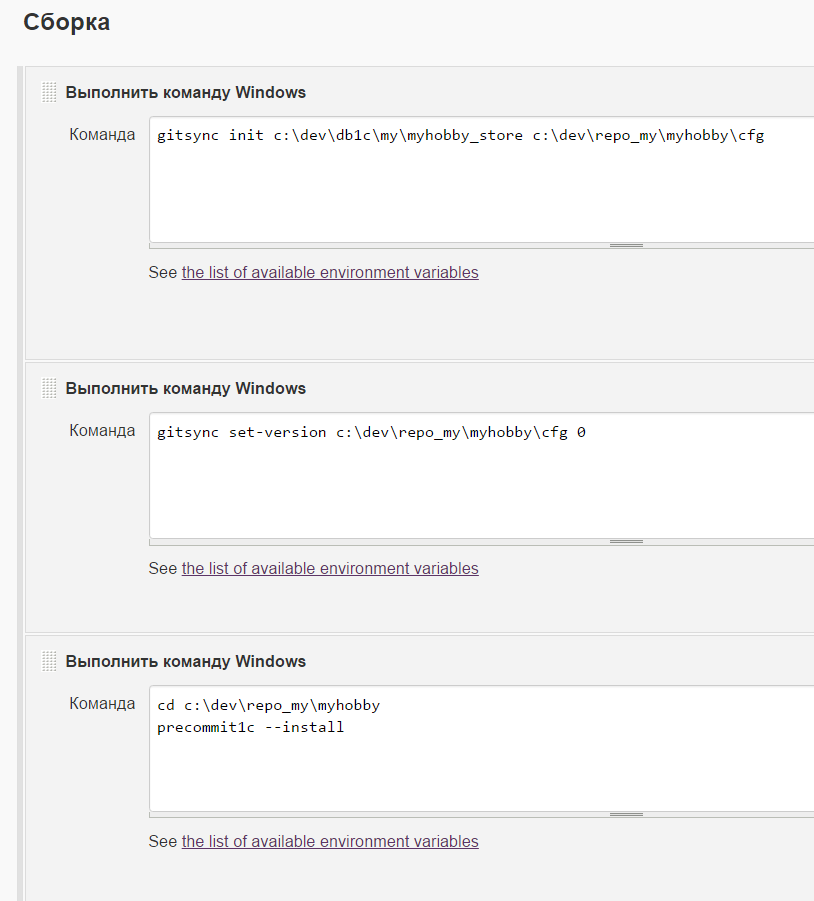
*Рисунок 10*

Далее открывается окно настройки проекта. В основном разделе при желании можно описать, что реализуется в данном сценарии.



*Рисунок 11*

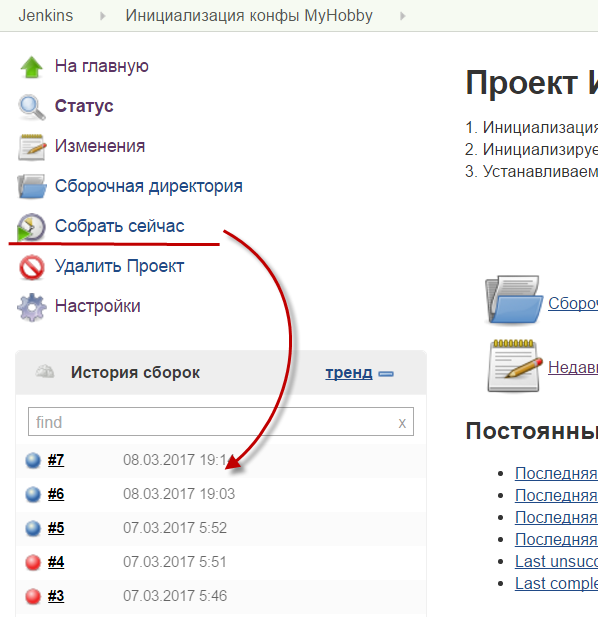
В разделе «Сборка» выбираем команду «Добавить шаг сборки – Выполнить команду Windows», и в появившемся поле ввода прописываем команду соответствующего шага. Затем повторяем то же самое для остальных шагов. В результате должно получиться то, что показано на рисунке 12.



*Рисунок 12*

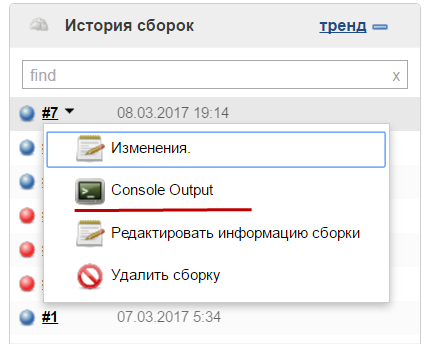
Нажимаем «Сохранить», и наш первый мини-проект готов. Теперь запустим его на выполнение.

После сохранения сценария открывается страница управления этим сценарием. Если выбрать команду «Собрать сейчас», то сценарий будет запущен, а результат появится в разделе «История сборок».



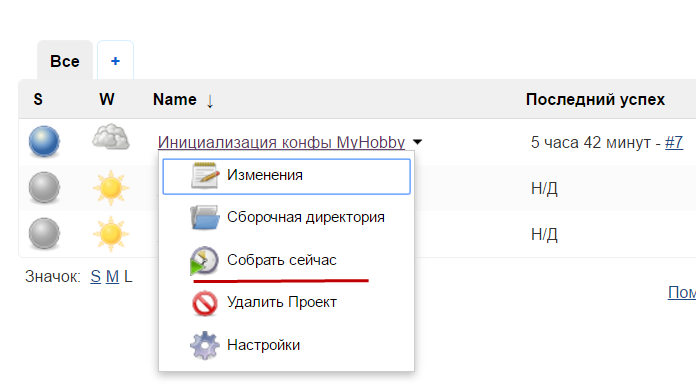
*Рисунок 13*

Из контекстного меню конкретного запуска сценария можно через команду «Console Output» посмотреть детали выполнения.



*Рисунок 14*

Также запустить сценарий можно из контекстного меню конкретного проекта в списке главного окна системы.



*Рисунок 15*

**Важно!**

Для реализации дальнейших команд будет необходимо выполнять команды Git-а из-под Jenkins. Чтобы задачи не валились с ошибкой, необходимо единовременно в самом начале выполнить заполнение глобальных настроек Git из-под юзера jenkins:

git config -–global user.name jenkins

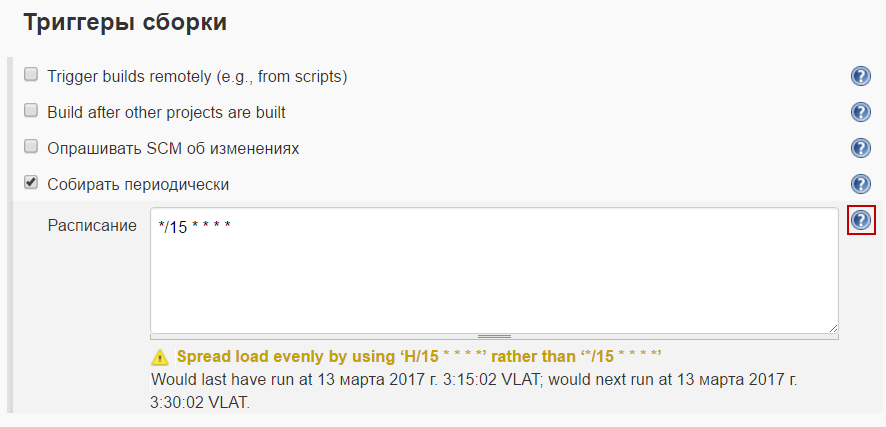
git config -–global user.email jenkins@example.ru

Я, для простоты, просто выполнил отдельную задаю Jenkins с этими командами Windows. В качестве имени и электронного адреса указываются те, от имени которых будет выполнять коммиты в Git автосборщик Jenkins-а.

## Периодическая синхронизация конфигурации с Git

Теперь создадим первый из двух рабочих сценариев для нашего проекта. Задача будет выполнять синхронизацию конфигурации из хранилища 1С в Git посредством инструмента gitsync (работа самого gitsync должна быть настроена предварительно, см. статью «Развёртывание проекта разработки 1С с использованием системы контроля версий Git»). При наличии изменений в хранилище 1С, gitsync автоматически запускает процесс распаковки конфигурации из хранилища 1С на исходные фалы, выкладывает изменения в локальный репозиторий Git и выполняет коммит.

Создаём новую задачу со свободной конфигурацией. В разделе «Триггеры сборки» («Build Triggers») ставим галку «Собирать периодически» («Build periodically»), и в появившемся поле «Расписание» («Schedule») прописываем строку: **\*/15 \* \* \* \***. Это означает, что сценарий будет запускаться каждые 15 минут. Подробнее о формате написания этой строки можно почитать в справке, которая открывается по «знаку вопроса» справа от поля ввода.



*Рисунок 16*

После этого добавляем командe Windows на синхронизацию gitsync:

gitsync export c:\dev\db1c\my\myhobby\_store c:\dev\repo\_my\myhobby\cfg

## Сборка конфигурации из исходников при появлении коммита в Git

(Вторая основная задача проекта. Периодически опрашиваем репо Гита, и, если есть новые коммиты, выполняем Пул, собираем конфу из исходников и деплоим в целевые контуры. Возможно, деплой вынесем в отдельные Джобы)

Вопросы, с которыми еще не разобрался:

* Параметризация сценариев
* Пуш