# Experiment No. 1.7

**Student Name:** Parvinder Singh **UID:** 22MCC20043
**Branch:** MCA - CCD **Section/Group:** 22MCD-1/ Grp B
**Semester:** III
**Subject Name:** DevOps Process Automation Lab **Subject Code:** 22CAP-745

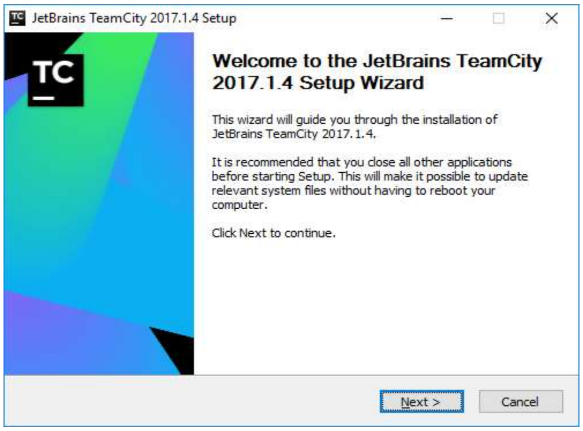
## 1. Aim/Overview of the practical:

- Install and configure TeamCity.
- Make sure you fork the SpringPetClinic repository in your Github and connect that repository with TeamCity.
- Create a new project via TeamCity dashboard and build all the 9 goals of Maven.

## 2. Code for experiment/practical:

### A. Install and configure TeamCity.

- Go to https://www.jetbrains.com/teamcity/download/#section=windows and click on the DOWNLOAD button to download the TeamCity installation package from the TeamCity website.
- Once downloaded, open the installation package and click Next:
- On the next screen, agree to the license and go to the next step:
- Select the path to install TeamCity in and click Next:



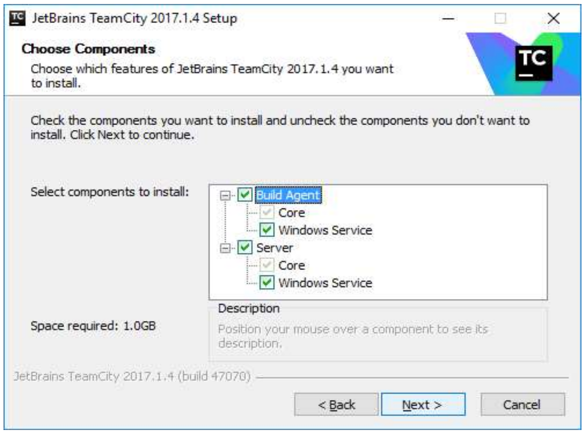
JetBrains TeamCity 2017.1.4 Setup

Welcome to the JetBrains TeamCity 2017.1.4 Setup Wizard

This wizard will guide you through the installation of JetBrains TeamCity 2017.1.4.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



JetBrains TeamCity 2017.1.4 Setup

Choose Components

Choose which features of JetBrains TeamCity 2017.1.4 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Build Agent
  - ☑ Core
  - ☑ Windows Service
- ☑ Server
  - ☑ Core
  - ☑ Windows Service

Space required: 1.0GB

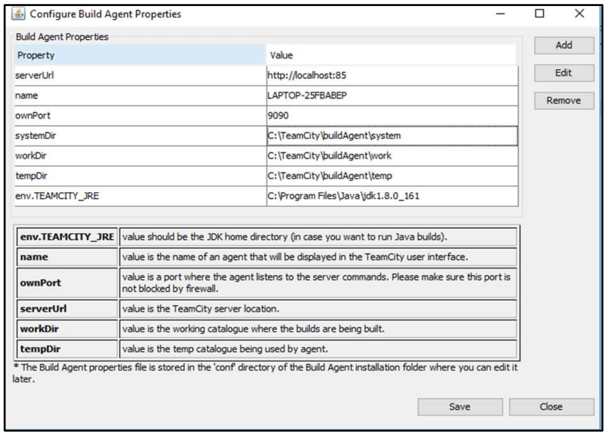
Description

Position your mouse over a component to see its description.

JetBrains TeamCity 2017.1.4 (build 47070)

- Select the packages to be installed; for example, if you are installing Build Agent and Server on different servers, then select accordingly. For learning purposes, you can select to install both on the same machine.

- Once the installation is done, select the port you would like the TeamCity server to run on. Make sure this port is not used by other services on the machine, and choose a unique port number.
- Once installation is done configure TeamCity server by creating Administrator Account and start using TeamCity.

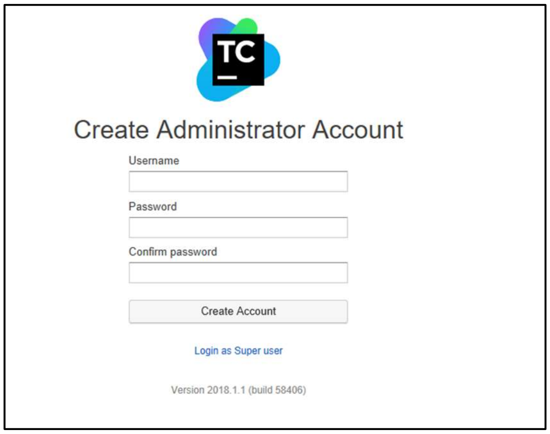


Build Agent Properties

| Property | Value |
|---|---|
| serverUrl | http://localhost:85 |
| name | LAPTOP-25FBABEP |
| ownPort | 9090 |
| systemDir | C:\TeamCity\buildAgent\system |
| workDir | C:\TeamCity\buildAgent\work |
| tempDir | C:\TeamCity\buildAgent\temp |
| env.TEAMCITY_JRE | C:\Program Files\Java\jdk1.8.0_161 |

| | |
|---|---|
| **env.TEAMCITY_JRE** | value should be the JDK home directory (in case you want to run Java builds). |
| **name** | value is the name of an agent that will be displayed in the TeamCity user interface. |
| **ownPort** | value is a port where the agent listens to the server commands. Please make sure this port is not blocked by firewall. |
| **serverUrl** | value is the TeamCity server location. |
| **workDir** | value is the working catalogue where the builds are being built. |
| **tempDir** | value is the temp catalogue being used by agent. |

* The Build Agent properties file is stored in the 'conf' directory of the Build Agent installation folder where you can edit it later.



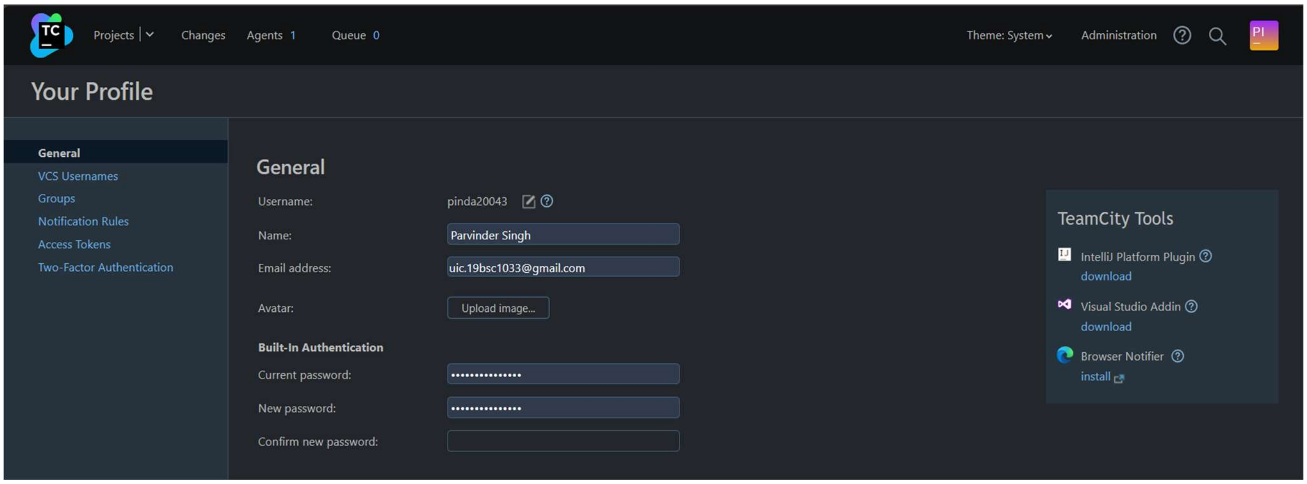
Create Administrator Account

Username:

Password:

Confirm password:

Version 2018.1.1 (build 58406)



Your Profile

General

Username: pinda20043

Name: Parvinder Singh

Email address: uic.19bsc1033@gmail.com

Avatar:

Built-In Authentication

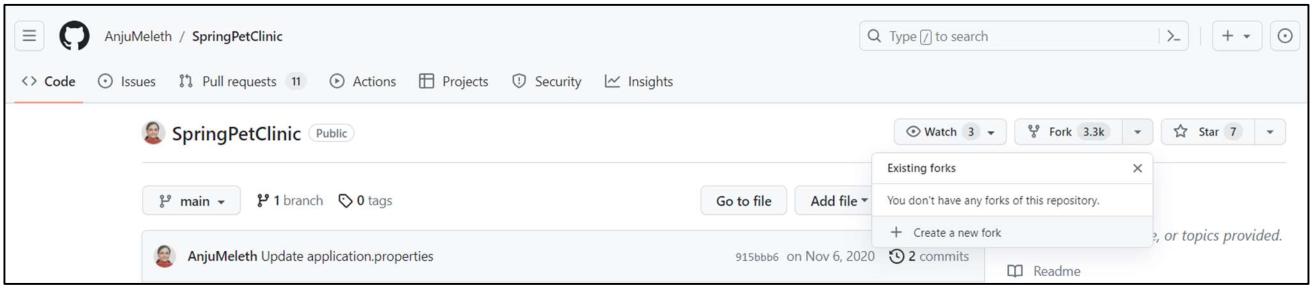
Current password: ••••••••••••••••

New password: ••••••••••••••••

Confirm new password:

## B. Make sure you fork the SpringPetClinic repository in your Github and connect that repository with TeamCity.

- Go to https://github.com/AnjuMeleth/SpringPetClinic
- Click on the Fork and Create a new fork



You don't have any forks of this repository.

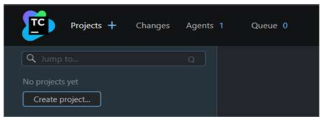
AnjuMeleth Update application.properties 915bbb6 on Nov 6, 2020 2 commits

, or topics provided.

## C. Create a new project via TeamCity dashboard and build all the 9 goals of Maven.

- Go to http://localhost:8111/favorite/projects?mode=builds

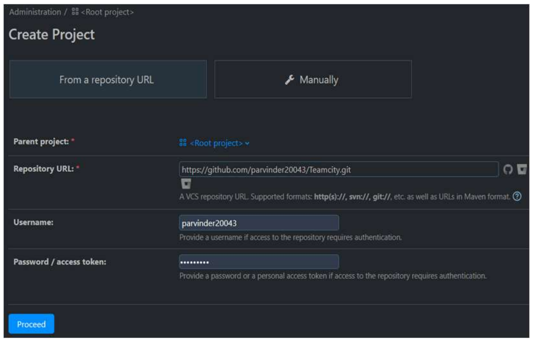
- Click **+ Create Project**.



No projects yet

Create project...

- Select the **From a repository URL tab**. In the Repository URL field, enter your repository, for example: https://github.com/parvinder20043/Teamcity.git .

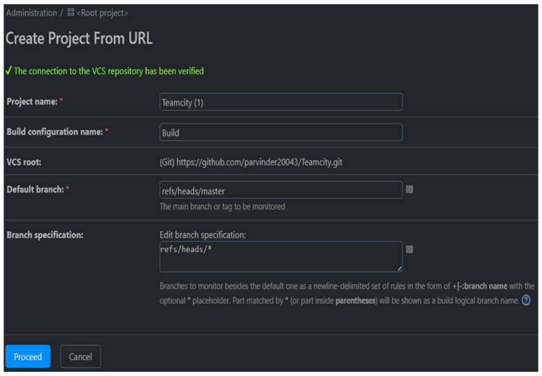


Administration / <Root project>

Create Project

| | |
|---|---|
| Parent project: * | <Root project> |
| Repository URL: * | https://github.com/parvinder20043/Teamcity.git<br>A VCS repository URL. Supported formats: **http(s)://, svn://, git://**, etc. as well as URLs in Maven format. |
| Username: | parvinder20043<br>Provide a username if access to the repository requires authentication. |
| Password / access token: | •••••••••<br>Provide a password or a personal access token if access to the repository requires authentication. |

- If your repository requires authentication, enter your username and password / access token.
- Click **Proceed**.
- If TeamCity successfully connects to your repository, you will see the following dialog.
- In the **Create Project From URL** dialog you have the option to change the project name and initial build configuration name.



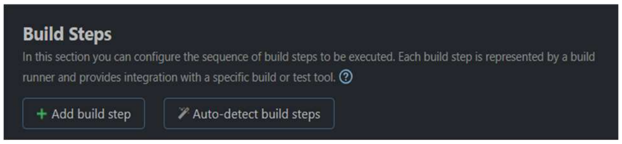
Administration / <Root project>

Create Project From URL

✓ The connection to the VCS repository has been verified

| | |
|---|---|
| Project name: * | Teamcity (1) |
| Build configuration name: * | Build |
| VCS root: | (Git) https://github.com/parvinder20043/Teamcity.git |
| Default branch: * | refs/heads/master<br>The main branch or tag to be monitored |
| Branch specification: | Edit branch specification:<br>refs/heads/*<br>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. |

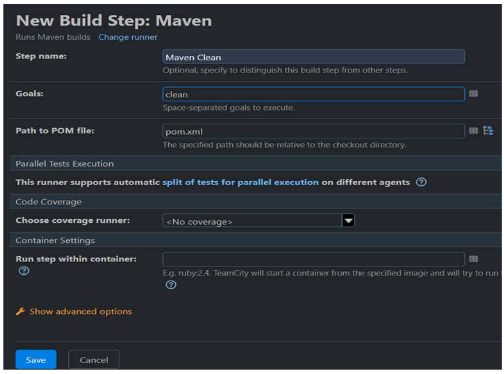
- Click **Proceed**. Project created successfully.
- Now **Select Build Steps > +Add Build Step > Maven.**



Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool.

- Start Adding Build Steps for **Maven Clean, Validate, Compile, Test, Package, Integration-test, Verify, Install, Deploy.**



**New Build Step: Maven**

Runs Maven builds Change runner

Step name: Maven Clean
Optional, specify to distinguish this build step from other steps.

Goals: clean
Space-separated goals to execute.

Path to POM file: pom.xml
The specified path should be relative to the checkout directory.

Parallel Tests Execution

This runner supports automatic split of tests for parallel execution on different agents

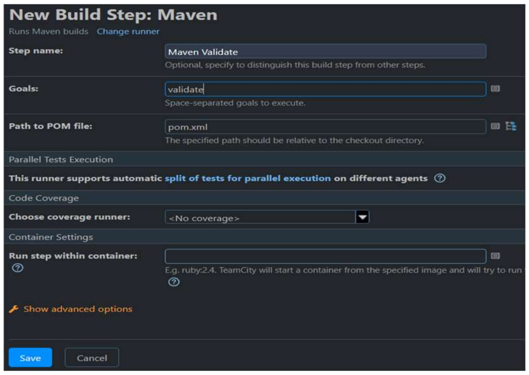
Code Coverage

Choose coverage runner: <No coverage>

Container Settings

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

→



**New Build Step: Maven**

Runs Maven builds Change runner

Step name: Maven Validate
Optional, specify to distinguish this build step from other steps.

Goals: validate
Space-separated goals to execute.

Path to POM file: pom.xml
The specified path should be relative to the checkout directory.

Parallel Tests Execution

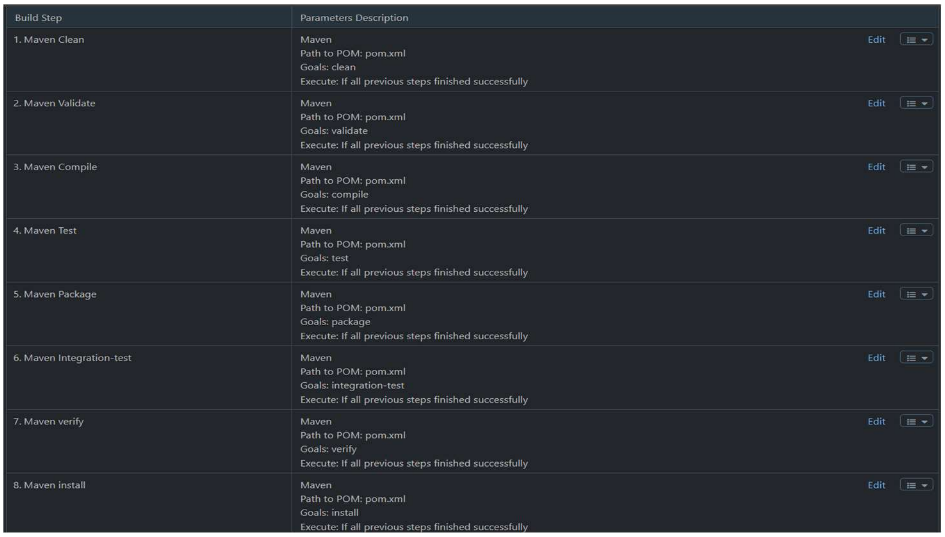
This runner supports automatic split of tests for parallel execution on different agents

Code Coverage

Choose coverage runner: <No coverage>

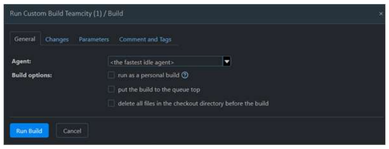
Container Settings

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



| Build Step | Parameters Description | |
|---|---|---|
| 1. Maven Clean | Maven<br>Path to POM: pom.xml<br>Goals: clean<br>Execute: If all previous steps finished successfully | Edit |
| 2. Maven Validate | Maven<br>Path to POM: pom.xml<br>Goals: validate<br>Execute: If all previous steps finished successfully | Edit |
| 3. Maven Compile | Maven<br>Path to POM: pom.xml<br>Goals: compile<br>Execute: If all previous steps finished successfully | Edit |
| 4. Maven Test | Maven<br>Path to POM: pom.xml<br>Goals: test<br>Execute: If all previous steps finished successfully | Edit |
| 5. Maven Package | Maven<br>Path to POM: pom.xml<br>Goals: package<br>Execute: If all previous steps finished successfully | Edit |
| 6. Maven Integration-test | Maven<br>Path to POM: pom.xml<br>Goals: integration-test<br>Execute: If all previous steps finished successfully | Edit |
| 7. Maven verify | Maven<br>Path to POM: pom.xml<br>Goals: verify<br>Execute: If all previous steps finished successfully | Edit |
| 8. Maven install | Maven<br>Path to POM: pom.xml<br>Goals: install<br>Execute: If all previous steps finished successfully | Edit |

- After Adding Build Steps Click on **Run**.

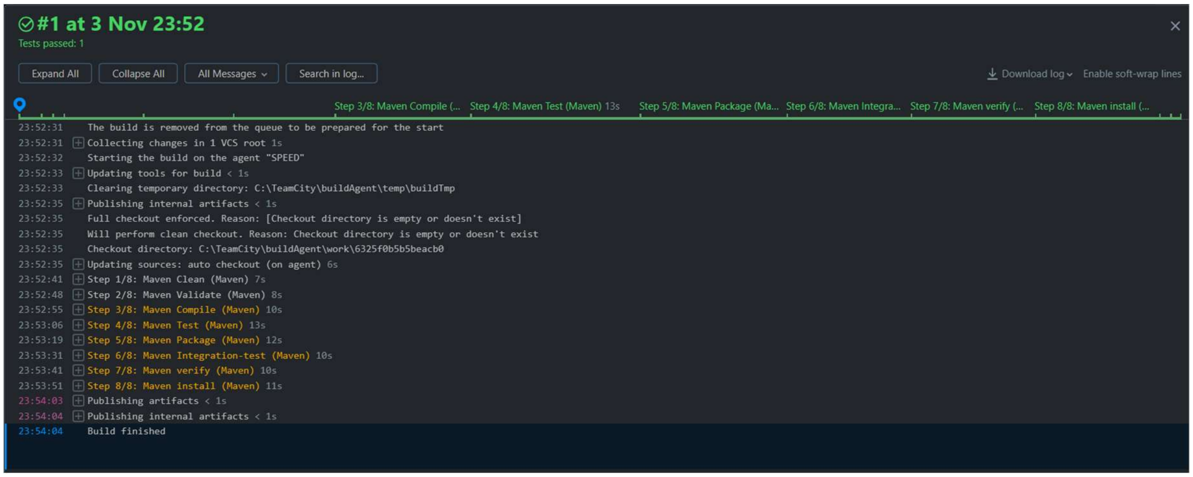


Run Custom Build Teamcity (1) / Build

Agent: <the fastest idle agent>

Build options:
□ run as a personal build
□ put the build to the queue top
□ delete all files in the checkout directory before the build

## D. Output



#1 at 3 Nov 23:52

Tests passed: 1

Expand All
Collapse All
All Messages
Search in log...
Download log
Enable soft-wrap lines
Step 3/8: Maven Compile (...
Step 4/8: Maven Test (Maven) 13s
Step 5/8: Maven Package (Ma...
Step 6/8: Maven Integra...
Step 7/8: Maven verify (...
Step 8/8: Maven install (...

```
23:52:31    The build is removed from the queue to be prepared for the start
23:52:31  ⊞ Collecting changes in 1 VCS root 1s
23:52:32    Starting the build on the agent "SPEED"
23:52:33  ⊞ Updating tools for build < 1s
23:52:33    Clearing temporary directory: C:\TeamCity\buildAgent\temp\buildTmp
23:52:35  ⊞ Publishing internal artifacts < 1s
23:52:35    Full checkout enforced. Reason: [Checkout directory is empty or doesn't exist]
23:52:35    Will perform clean checkout. Reason: Checkout directory is empty or doesn't exist
23:52:35    Checkout directory: C:\TeamCity\buildAgent\work\6325f0b5b5beacb0
23:52:35  ⊞ Updating sources: auto checkout (on agent) 6s
23:52:41  ⊞ Step 1/8: Maven Clean (Maven) 7s
23:52:48  ⊞ Step 2/8: Maven Validate (Maven) 8s
23:52:55  ⊞ Step 3/8: Maven Compile (Maven) 10s
23:53:06  ⊞ Step 4/8: Maven Test (Maven) 13s
23:53:19  ⊞ Step 5/8: Maven Package (Maven) 12s
23:53:31  ⊞ Step 6/8: Maven Integration-test (Maven) 10s
23:53:41  ⊞ Step 7/8: Maven verify (Maven) 10s
23:53:51  ⊞ Step 8/8: Maven install (Maven) 11s
23:54:03  ⊞ Publishing artifacts < 1s
23:54:04  ⊞ Publishing internal artifacts < 1s
23:54:04    Build finished
```