

TeamCity

This section guides you through deploying TeamCity as a continuous integration (CI) tool on a Hyper-V virtual machine in your environment so that you can observe the test runs from the CI Dashboard in the **Test Analysis** view.

The URLs and values shown in the images that are included in these steps serve as examples only.

Prerequisites

Make sure you have the following installed:

- [TeamCity](#)
- [JDK 1.8](#)

On this page:

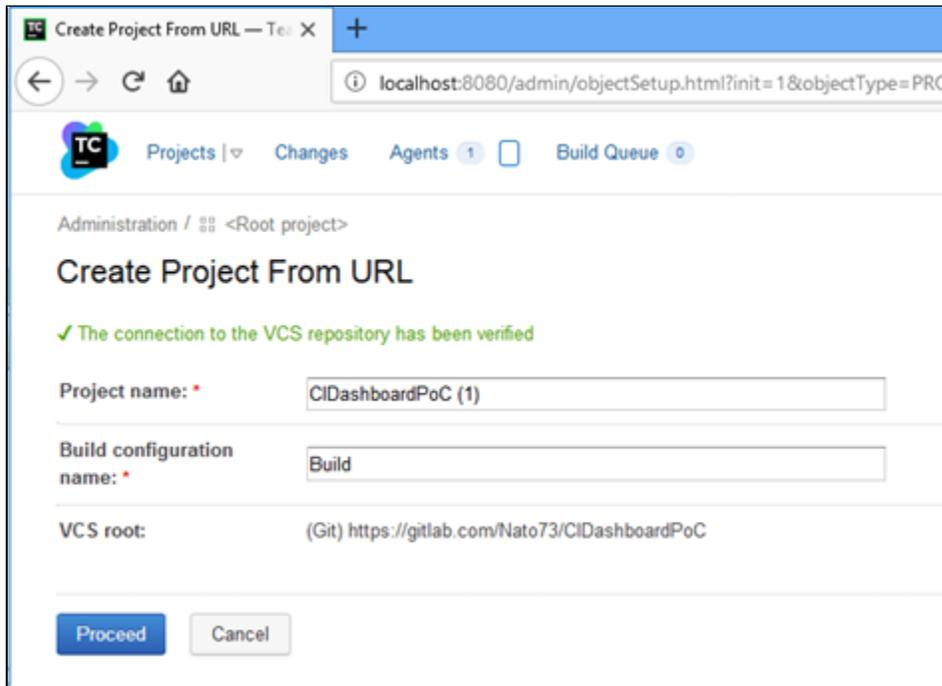
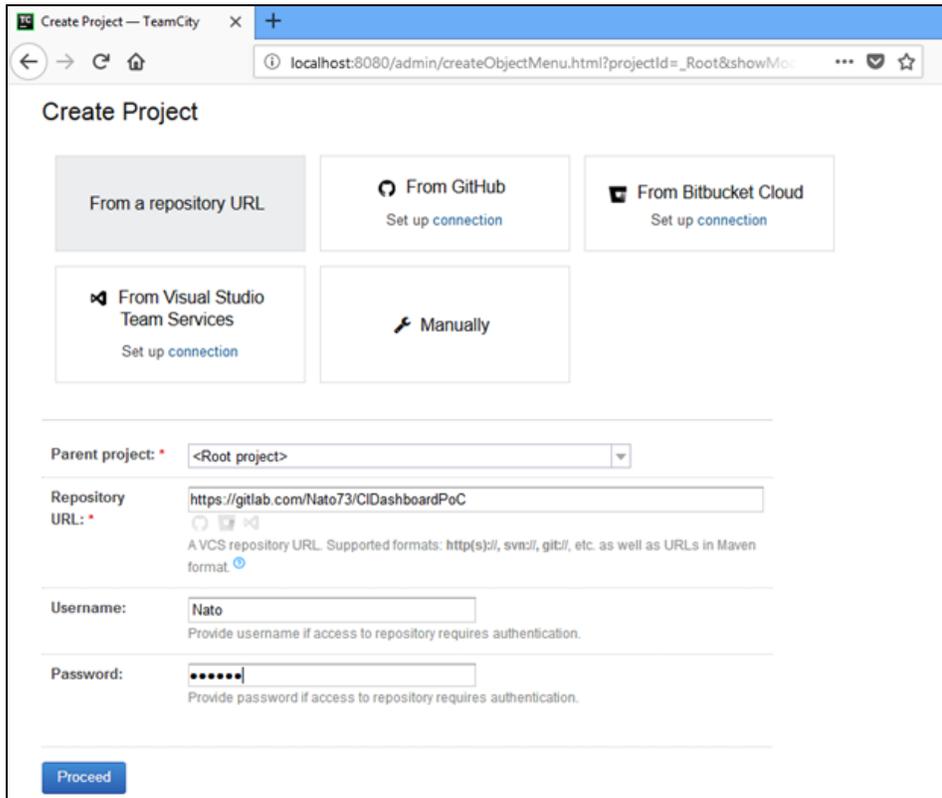
- [Prerequisites](#)
- [1 | Store source code in a source code repository](#)
- [2 | Create a TeamCity project](#)
- [3 | Add a Maven building agent to the TeamCity project](#)
- [4 | Run the TeamCity project manually at any time](#)

1 | Store source code in a source code repository

To allow the TeamCity building agent to pull the source code of the automation package, you need to store the package in a version control system such as Bitbucket, Git, CVS, Microsoft Team Foundation Server (TFS), Perforce Helix Core, and so on. This section uses Bitbucket as an example.

2 | Create a TeamCity project

1. Restart the TeamCity VM.
2. Access the console. (Depending on the settings, you can have a different URL.)
3. Create a new project.



3 | Add a Maven building agent to the TeamCity project

1. When prompted for Maven settings, install the Maven support plugin by selecting the checkbox for the **Maven** build step and clicking **Use selected**.

Administration / <<Root project> / CIDashboardPoC (1) Run ... Actions Build Configuration Home

Build

General Settings
Version Control Settings 1
Build Steps
Triggers 1
Failure Conditions
Build Features
Dependencies
Parameters
Agent Requirements
Suggestions 1
Hide unconfigured

New project "CIDashboardPoC (1)", build configuration "Build" and VCS root "https://gitlab.com/Nato73/CIDashboardPoC.git#refs/heads/master" have been successfully created.

Auto-detected Build Steps
Build steps and their settings are detected automatically by scanning VCS repository. You can [configure build steps manually](#) if auto-detect did not find relevant build steps.

<input type="checkbox"/>	Build Step	Parameters Description
<input type="checkbox"/>	Maven	Path to POM: pom.xml Goals: clean test

Use selected Refresh

Maven support	58245	JetBrains, s.r.o.	<WEB-INF>/plugins/Maven2	Disable
---------------	-------	-------------------	--------------------------	---------

2. Select the **Maven** build step and click **Use selected**.
3. In the **Build Step: Maven** panel, provide the following information:

- **Runner type:** Maven
- **Execute step:** If all previous steps finished successfully
- **Goals:** clean test
- **Path to POM file:** pom.xml
- **Additional Maven command line parameters:**

```
-Dmaven.test.failure.ignore=true -Dreortium-job-name="%env.TEAMCITY_PROJECT_NAME%" -
Dreortium-job-number=%env.BUILD_NUMBER% -Dreortium-job-branch="%myBranch%" -Dreortium-
tags="%myTag%"
```

Administration / <<Root project> / CIDashboardPoC Run ... Actions Build Configuration Home

Build 1 info item

General Settings
Version Control Settings 1
Build Step: Maven
Triggers 1
Failure Conditions
Build Features
Dependencies
Parameters 2
Agent Requirements
Suggestions
Hide unconfigured

Last edited 6 hours ago by nato (view history)

Build Step + Add build step >

Runner type: Maven
Runs Maven builds

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

Execute step: If all previous steps finished successfully
Specify the step execution policy.

Goals: clean test
Space-separated goals to execute.

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

Additional Maven command line parameters: -Dmaven.test.failure.ignore=true -Dreortium-job-name="%env.TEAMCITY_PROJECT_NAME%" -Dreortium-job-number=%env.BUILD_NUMBER% -Dreortium-job-branch="%myBranch%" -Dreortium-tags="%myTag%"

Working directory:
Optional, set if differs from the checkout directory.

4 | Run the TeamCity project manually at any time

When setup is done, run the build and check the CI Dashboard in the **Test Analysis** view of your Perfecto cloud. Sample URL: <https://<your-cloud>.reporting.perfectomobile.com/dashboard-ci>