Apache Ignite Teamcity Bot

- Introduction
- How to check a PR with the TC Bot
- How to use
  - Notification Types
    - New test failure notification
    - Flaky test failure notification
    - Newly contributed tests failure
    - Critical suite failure
- Contributing to TC Bot

Introduction

This tool monitors TeamCity [https://ci.ignite.apache.org/](https://ci.ignite.apache.org/) and send notifications to dev. list if it detects new failures. Source code available at

- ASF tools repository: [https://gitbox.apache.org/repos/asf/ignite-teamcity-bot.git](https://gitbox.apache.org/repos/asf/ignite-teamcity-bot.git)
- Mirror [https://github.com/apache/ignite-teamcity-bot](https://github.com/apache/ignite-teamcity-bot)

Ignite Developers are welcome to give feedback and post issues.

This tool can be also used for checking PR for introduction of new failures.

How to check a PR with the TC Bot

![Diagram](https://via.placeholder.com/150)

How to use

Analyze latest TeamCity tests results against the latest master:

1. Go to the PR/Branch check page
2. Fill the form "Check branch/PR":
   a. Leave base branch empty.
   b. Enter branch from TeamCity ("pull/<pull-request-number>/head").
   c. Press "Latest" button to compare with the latest master.
3. On the appeared page you can see a table with failed tests and suites. Also, you can see test's failure rate for last 100 runs of master branch. Most important failures are placed under "Possible Blockers" header in the table. You need to fix them or ensure that they aren't yours before moving ticket to Patch Available.

Analyze latest TeamCity tests results against the other branch:

1. Go to the PR/Branch check page
2. Fill the form "Check branch/PR":
   a. Enter base branch from TeamCity ("pull/<pull-request-number>/head").
   b. Enter branch from TeamCity ("pull/<pull-request-number>/head").
   c. Press "Latest" button to compare PR branch against the base branch.
3. On the appeared page you can see a table with failed tests and suites. Also, you can see test's failure rate for last 100 runs of base branch. Most important failures are placed under "Possible Blockers" header in the table.

Analyze specific TeamCity build against the latest master:

1. Go to the PR/Branch check page
2. Fill the form "Check build" with build ID from TeamCity.
3. On the appeared page you can see a table with failed tests and suites. Also, you can see test's failure rate for last 100 runs of master branch.

Print possible blockers to the JIRA ticket:

1. Go to the PR/Branch check page
2. Fill the form "Notify JIRA":
   a. Leave branch empty or "<pull-request-number>":

---

*Fill in the blanks*
Trigger "Run All" and comment JIRA by one button:

1. Go to the PR/Branch check page
2. Find PR, press "Inspect" and then "Trigger build and comment JIRA after finish". Fill the form "TeamCity Run All";
   a. Enter branch from TeamCity (or "<pull-request-number>" only).
   b. Enter JIRA ticket number. If PR is named according to contributing guide (name starts with "IGNITE-"), you can leave this field empty.
3. Press "Start tests and comment JIRA ticket on ready" button.

View and compare the statistics of "Run All" for master branch for a period of time.

1. Go to the Master Trends page
2. Select the interval and click "Apply". The minimum interval is 1 day, the maximum interval is 7 days.

The selection includes builds, the start date of which is between 00:00:01 of the first day and 23:59:59 of the last day of the interval.

For each field of "TESTS" and "PROBLEMS" sections show general results: min - median - max. By clicking on the icon near field name will show line chart for all data this field. Cells in the "PROBLEMS" section are additionally dyed green if the current median is less than the median in the other column. If more, then red.

The TeamCity server stores information about the builds for ~ two weeks. At the end of this period, the data is deleted from the TC server. Apache Ignite Teamcity Helper saves all received data in the cache so that they are available after this time. But still some of the information may be missing. If for the whole period there is no information on any build, you will receive a message «No data for the selected period». If data are available for not all days, you will receive a message «Data for the period from DD-MM-YYYY to DD-MM-YYY» or «Data for DD-MM-YYY».

Notification Types

This section covers notification types utility can send to dev@ list. User can specify his or her email and select desired tracked branches. All failures in this branches will be also forwarded to user.

New test failure notification

New test failure notification is generated if test was stable passing and then became stable failing. Stable passing test requires at least 5 sucessfull runs in a row. Stable failure requires 4 failures in a row.

Let's define 0 as success, 1 as failure.

So history of runs ..000001111... will cause notification. First transition 0->1 can be bug introduced test failure, and notification is linked to this particular build. Notification will not be resend in case tests continues to fail. This is done because all contributors are considered as interested in sucessfull tests passing and will do required steps to fix issue.

Duplicate notification can occur for same test if there is history ...000001111...000001111.. And second transition 0 ->1 can be potentially new problem, and test failure would be re-notified.

Flaky test failure notification

Some tests are flaky and sometimes change its state. This means test can be unlucky failed 4 times in a row. To protect from spam the Bot checks if fail row is happened on the one commit. If yes - test fail is considered as fail for 8 failures in a row. So history should be at least ..0000011111111...

Newly contributed tests failure

If test has no previous history and failed 4 times in a row, then it is considered as newly introduced failure.

Critical suite failure

The Bot is able to handle timeouts & JVM crashes in suites in a special way. These type of failures are named Critical. If 4 or more (timeouts/JMV crashes) occcurred several times in a row, this will generate suite-related notification. Also it is required that last run was completed with timeout.

Lets name result 2 as critical failure. So it is required to have 5 non critical failures and 4 critical to generate notification: ....(0/1)(0/1)(0/1)(0/1)2222....

Contributing to TC Bot

Please see Github integrated documentation on how to contribute: https://github.com/apache/ignite-teamcity-bot#development