Selenium Starter Kit
Running tests in CI/CD Pipeline

Contents

What's this guide?

This document gives you step-by-step instructions on running the CRM Automation Selenium-based UI tests in your CI/CD pipeline. It shows how you can run your UI tests in a Selenium Grid and how to trigger tests from Jenkins.

Using a Selenium Grid framework enables you to run your tests on different browsers and devices without having to maintain and configure test environments and hardware. Popular Selenium Grid providers are **CrossBrowserTesting**, **BrowserStack**, and **SauceLabs**.

To demonstrate running the UI tests in a CI/CD pipeline, we will use the test provider CrossBrowserTesting (CBT) as the test service to run the UI tests on. Therefore, you must have a CrossBrowserTesting account to proceed with this exercise. See https://crossbrowsertesting.com/ for information on CBT.

Prerequisites

Before you proceed, make sure your test projects are already setup. If not, go to Running OOTB CRM Automation UI Tests and follow the instructions for setting up the test projects and running the UI tests locally.

Required CrossBrowserTesting (CBT) Account

You need a CBT account to proceed with this exercise because the test project is configured to run on CBT. See https://crossbrowsertesting.com/ for information on CBT.

Required Third-Party Software

- Java version 1.8
- Apache Maven
- Install Git (if the project is source-controlled via Git)
- Jenkins War File
- Jenkins plug-ins
  - Cucumber reports plugin
  - Maven Plugin

Downloading the CBT tunnel binary

Download the CBT tunnel binaries from here and put them under **pega-crm-ui-testframework/binaries** folder.

Configure Project to run against CBT account

First, make sure the you configure the CRM Automation UI test project **pega-crm-ui-testframework** to run on your CrossBrowserTesting (CBT).

To run the UI tests on your CBT account:

1. In the **pega-crm-ui-testframework/data/global-settings.properties** file, set **hub.url** property to point to the CBT hub URL.
2. In the **pega-crm-ui-testframework/data/cross-browser-tunnel.properties** file, enter your credentials for your CBT provider.
3. Make sure you installed Java and Maven on your machine and have set them to your system paths.
4. To build and run the tests on CBT, navigate to **pega-crm-ui-testframework** project and input the command "mvn clean test".

The UI test project **pega-crm-ui-testframework** is pre-configured to run on Chrome. This is specified in global-settings.properties

Running the UI tests from a CI/CD pipeline

Now that you have configured your UI tests to run on CBT, you can now integrate your UI test automation suite as part of your continuous integration pipeline.

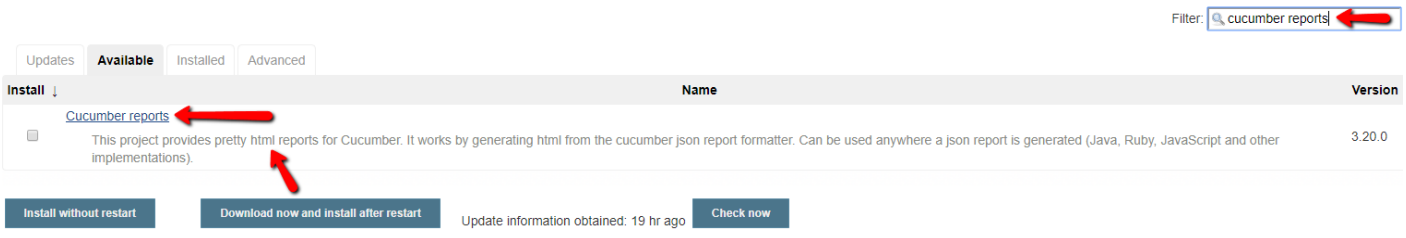
Requirement:

You need to store your UI test project and binaries in a repository.

Installing Required Plugins

Install **Cucumber reports** plugin

- Open Jenkins
- Go to **Manage Jenkins** and then **Manage Plugins**
- Switch to **Available** tab
- Search for **Cucumber reports** via the filter box

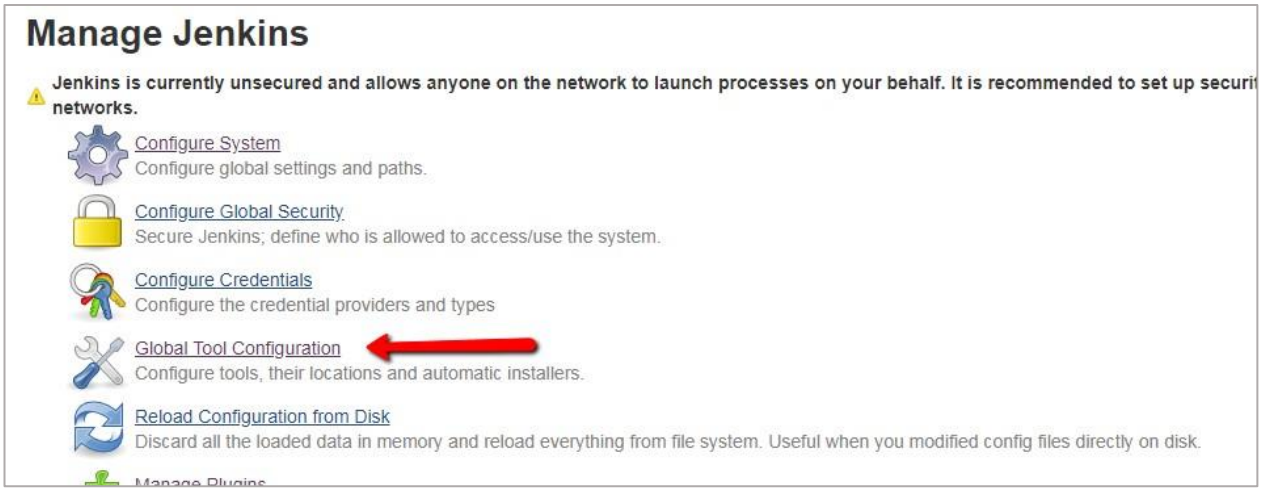


Check the **Cucumber reports** checkbox and click **Install Without Restart.**

Setting Up Jenkins

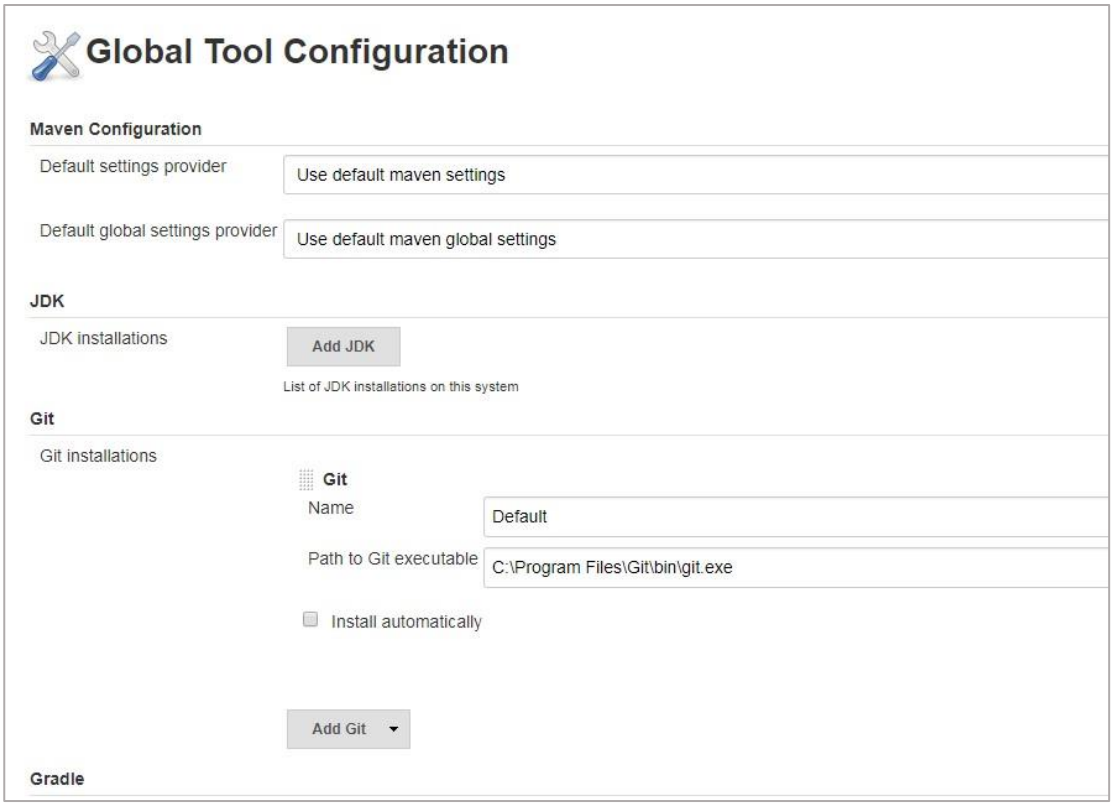
- Click **Manage Jenkins**

- Go to **Global Tool Configuration**



Set up Git

- In **Global Tool Configuration**, go to **Git** section
- Set the **Path to Git executable** to the location of the git.exe file installed on your system

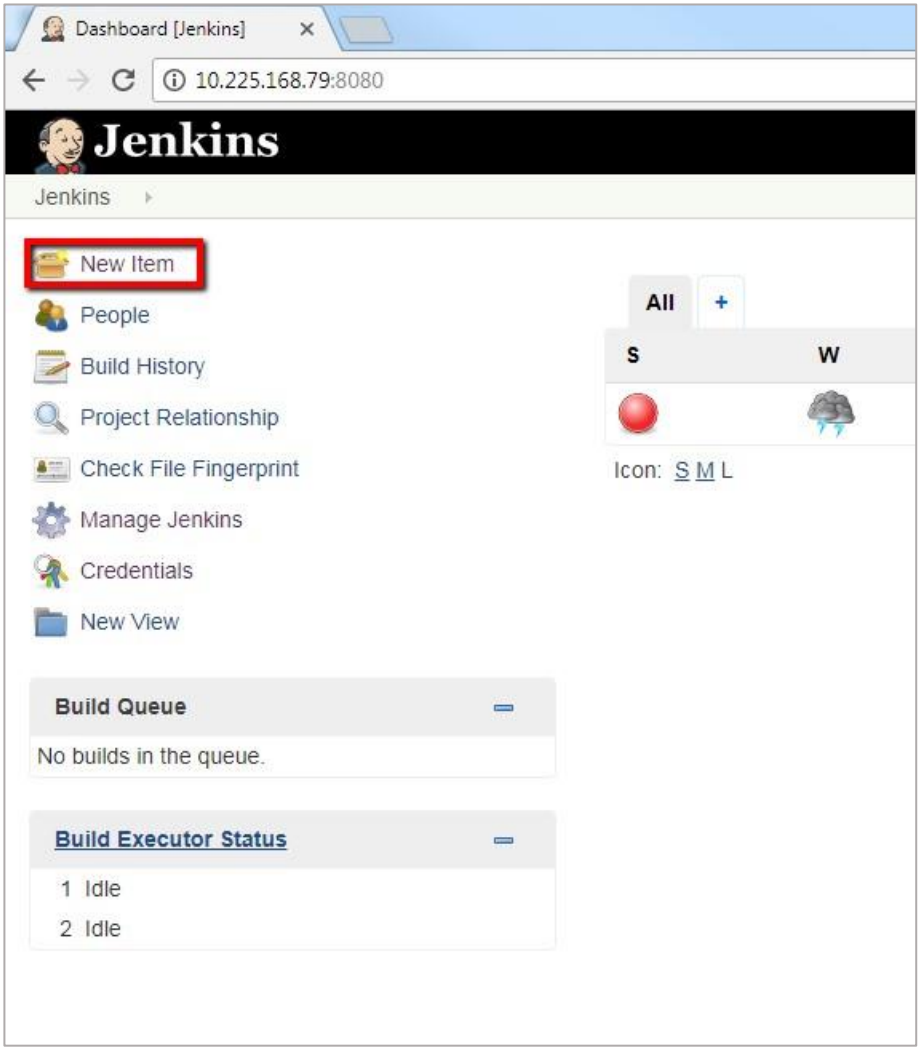


Set up Maven

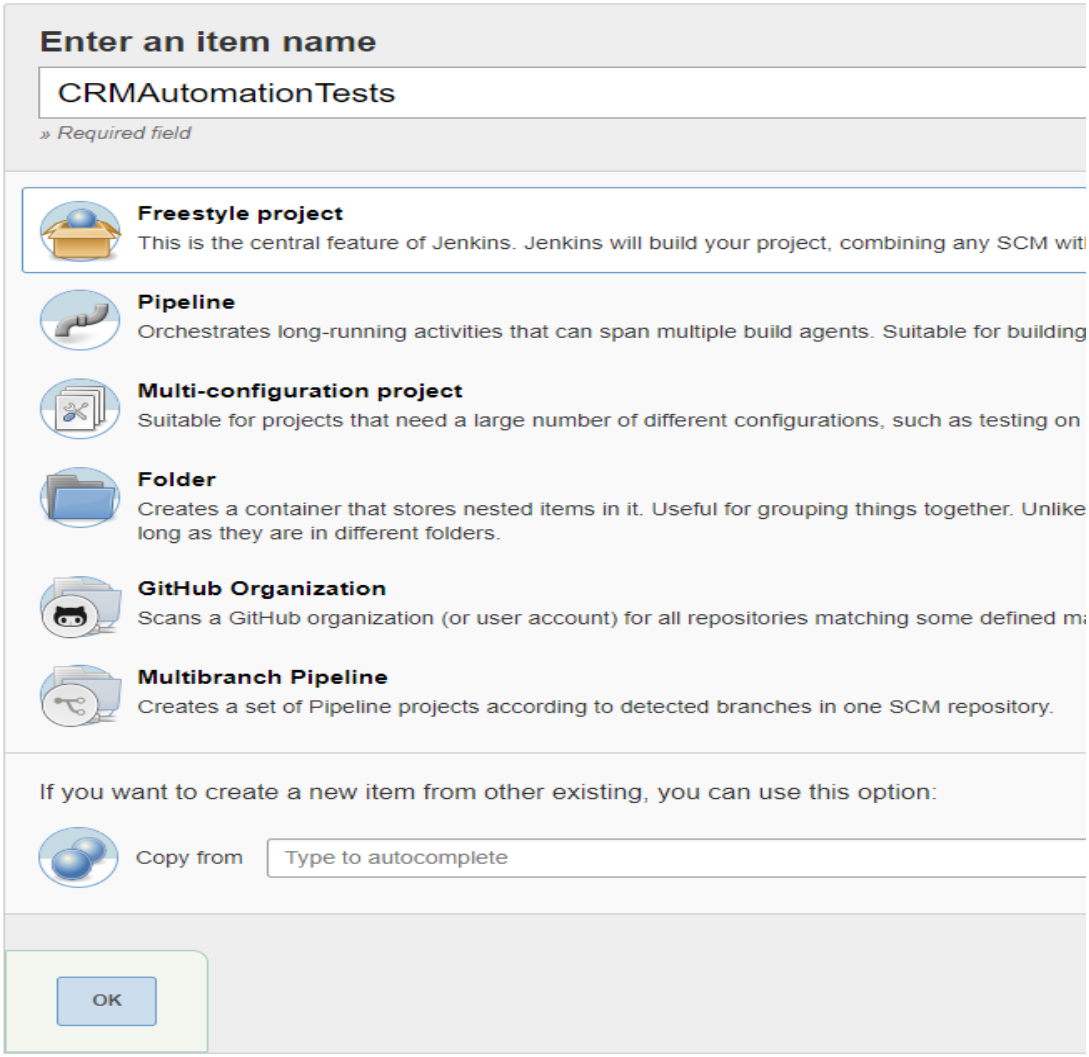
- Under **Maven** section, click **Add Maven**
- Set **Name** as `Maven`
- Check **Install Automatically**
- In **Install from Apache**, select the latest version
- Click on **Save** Button

Creating a Jenkins Job

1. Create a new job in Jenkins
    - Click **New Item**
    - Set the job name to `CRMAutomationTests`



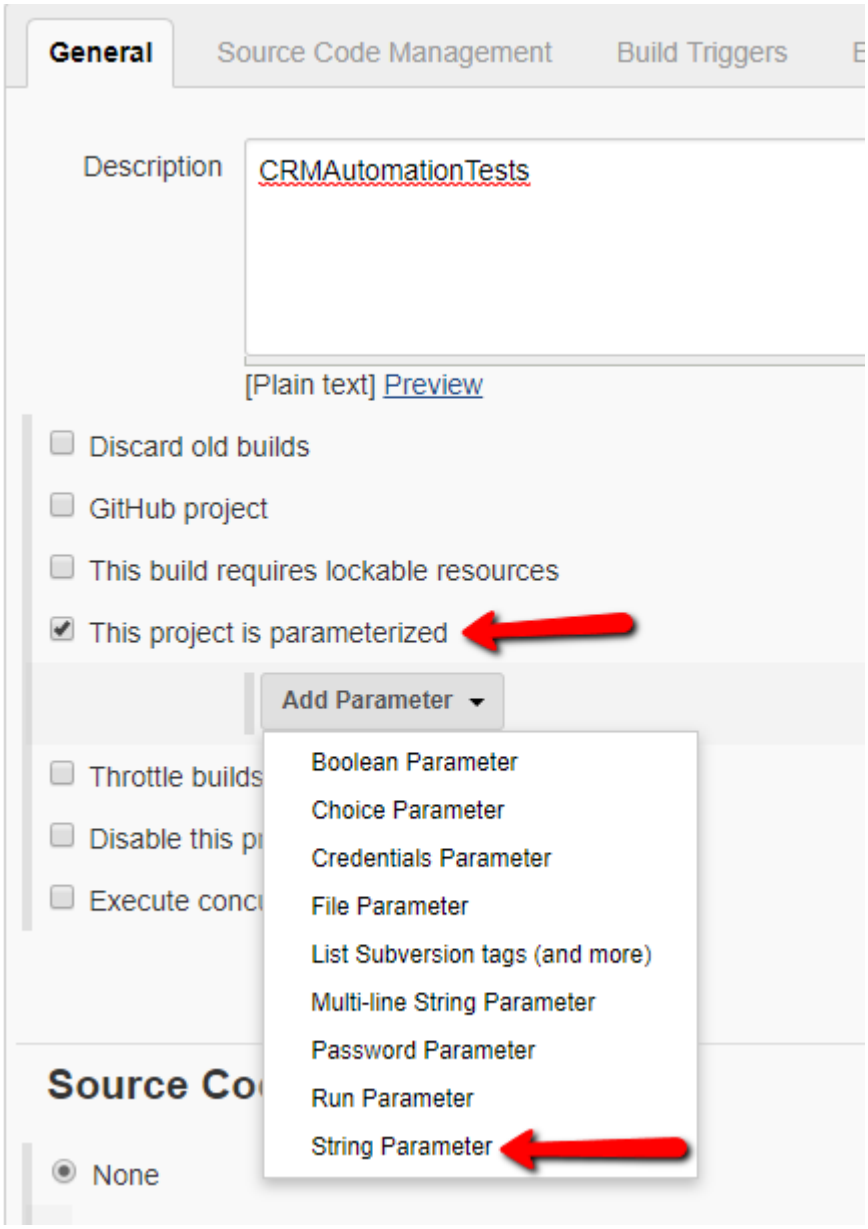
- Select **Freestyle project** and click **OK**

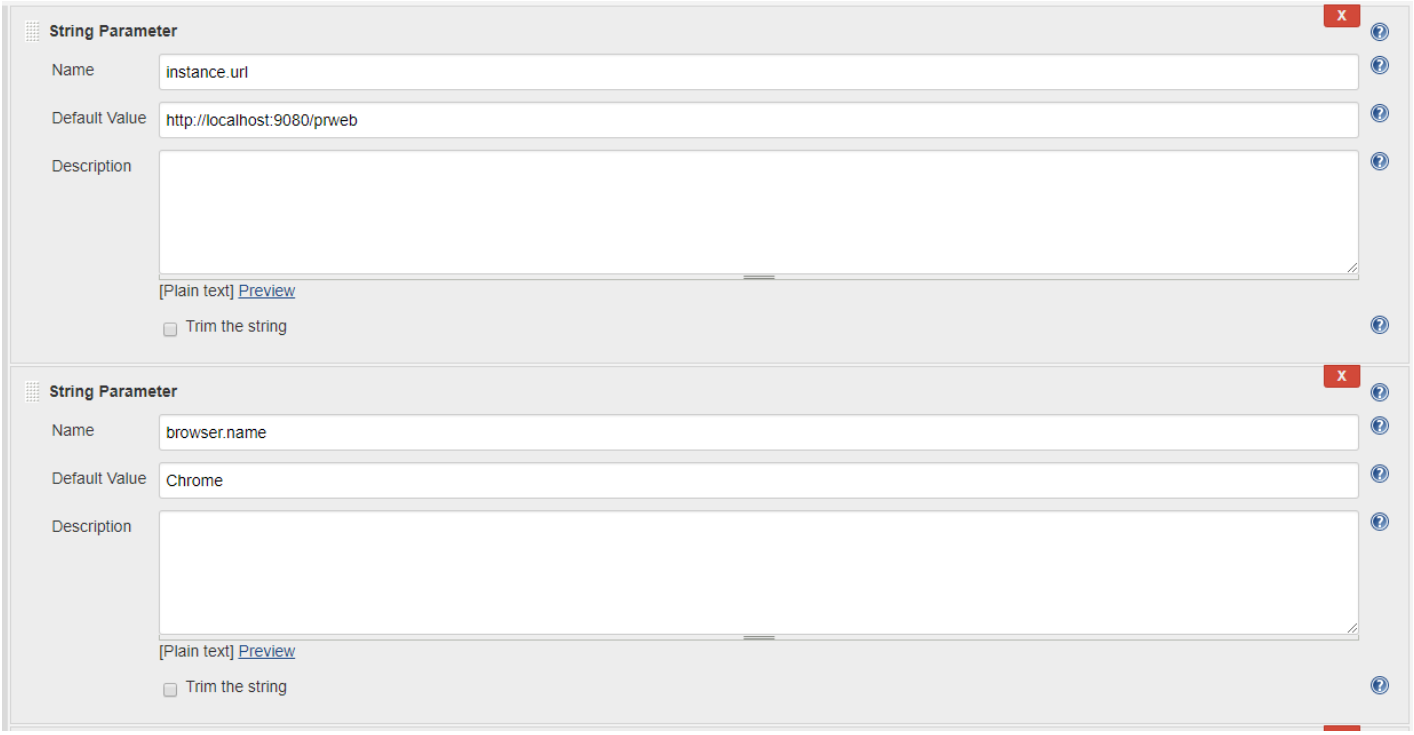


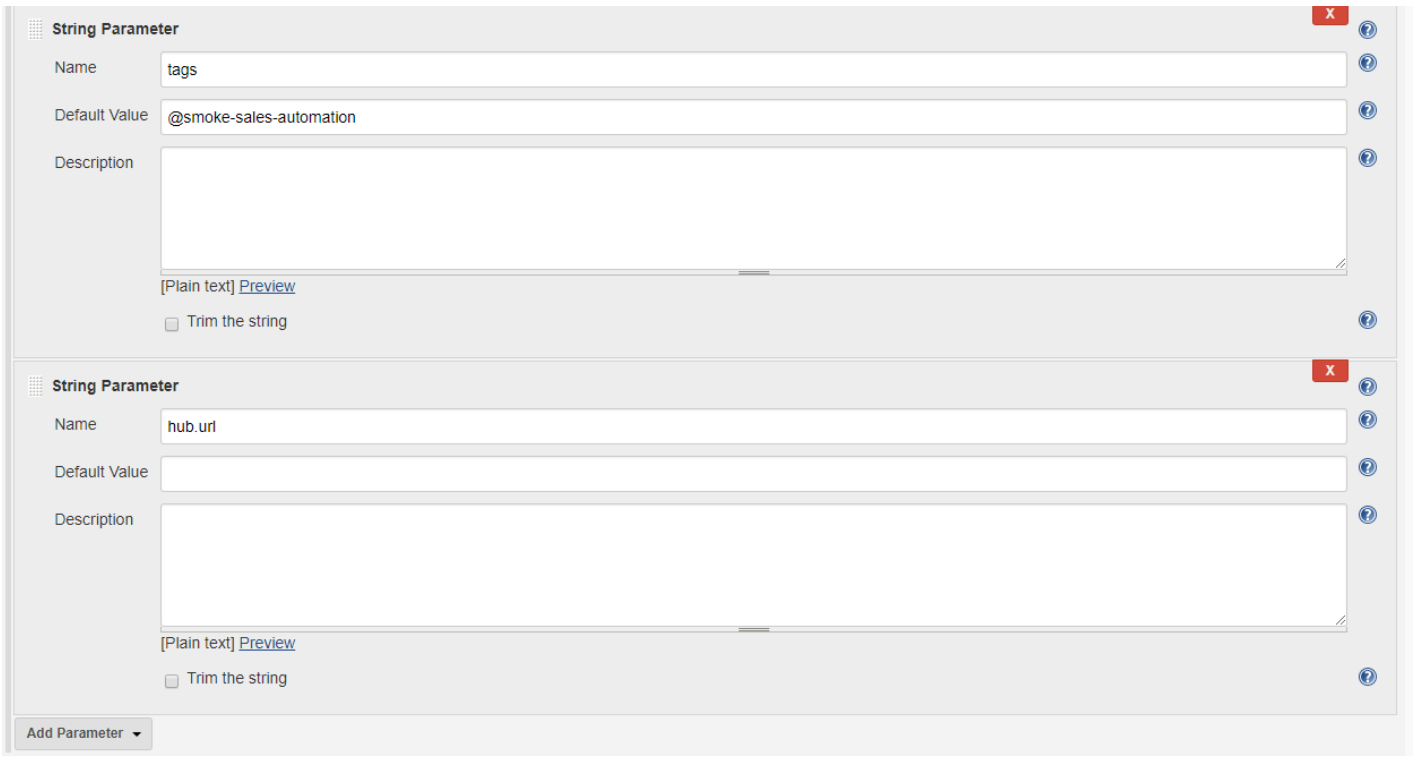
2. Configure the parameters required to run the tests in CBT
   - Select the checkbox: `This build / project is parameterized`
   - Add String Parameter:
     - Name: `instance.url`
     - Description: This is the application URL under test
   - Add String Parameter:
     - Name: `browser.name`
     - Description: This is the browser the tests will be run on, e.g. "chrome"
   - Add String Parameter:
     - Name: `tags`

- Description: This is the tag name corresponding to the testsuite or testcase you want to run. Tag name can be found on top of each feature or scenario in cucumber gherkin files

- Add String Parameter:
  - Name: `hub.url`
  - Description: This is the CrossBrowserTesting URL
- For each parameter, provide any default value and the description of each parameter.



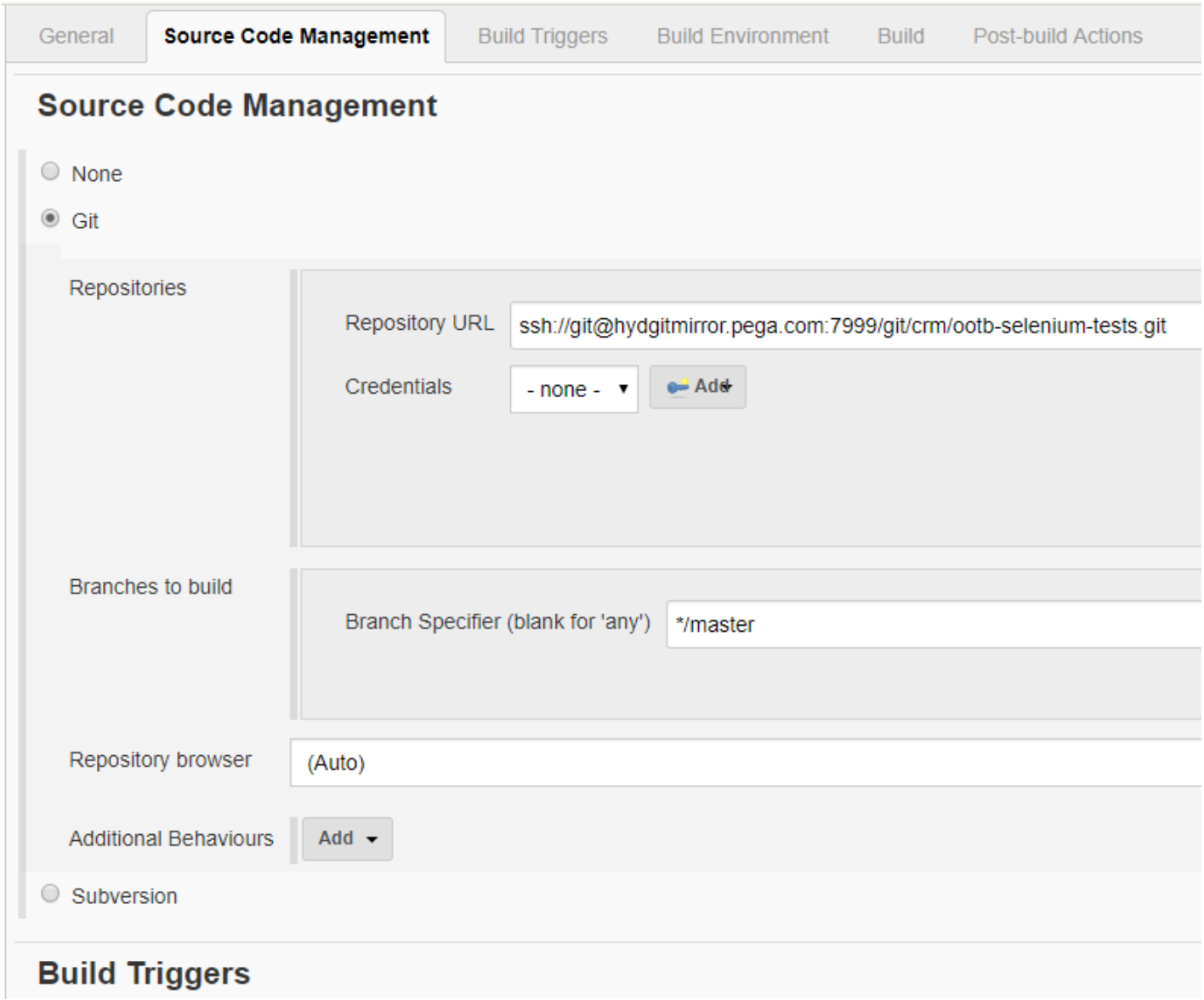

String Parameter
Name
instance.url
Default Value
http://localhost:9080/prweb
Description
[Plain text] Preview
Trim the string
String Parameter
Name
browser.name
Default Value
Chrome
Description
[Plain text] Preview
Trim the string


String Parameter
Name
tags
Default Value
@smoke-sales-automation
Description
[Plain text] Preview
Trim the string
String Parameter
Name
hub.url
Default Value
Description
[Plain text] Preview
Trim the string
Add Parameter

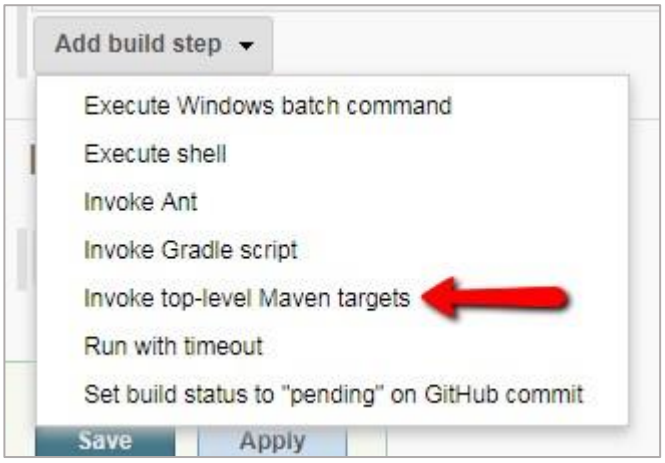
Leave `hub.url` empty if you are just running the tests on a local system.

In case `instance.url` is not publicly available, CrossBrowserTesting provides tunneling to enable local connection. In this case, you need to provide the `username` and `authkey` for tunneling in the `crossbrowsertunneling.properties` file in your test project.

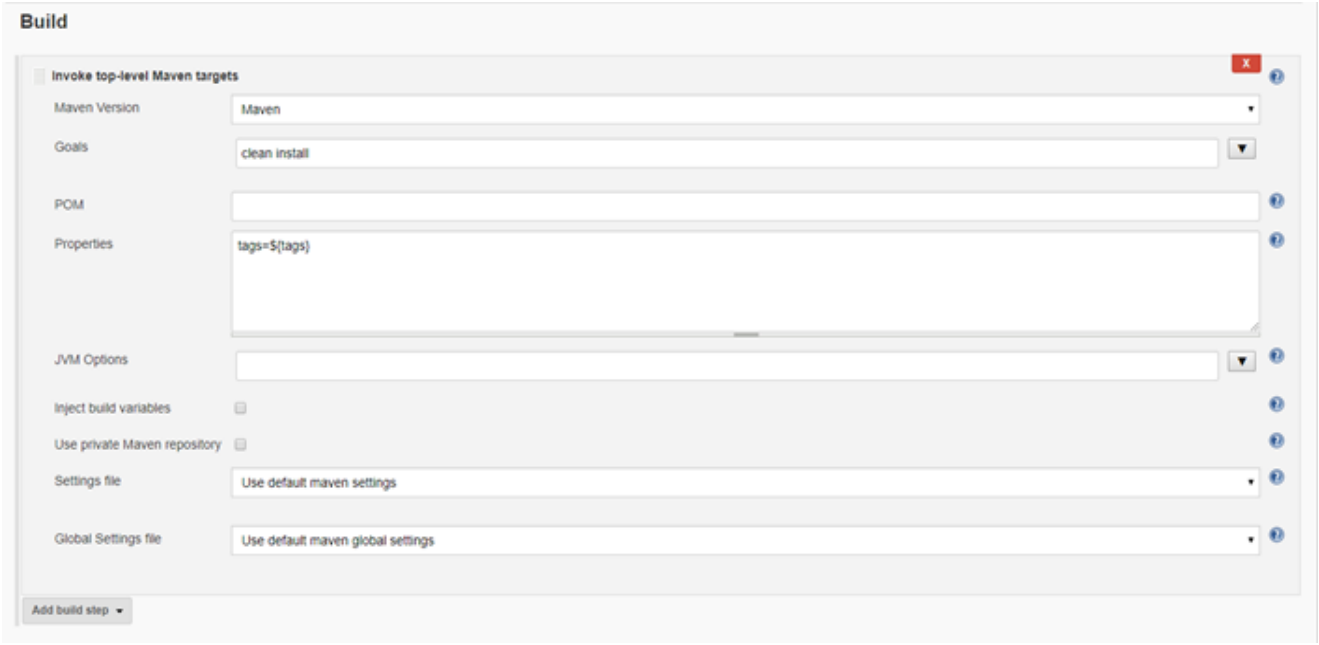
3. Configure the Source Code Management (SCM) via GIT or Subversion (where ever the tests are placed).



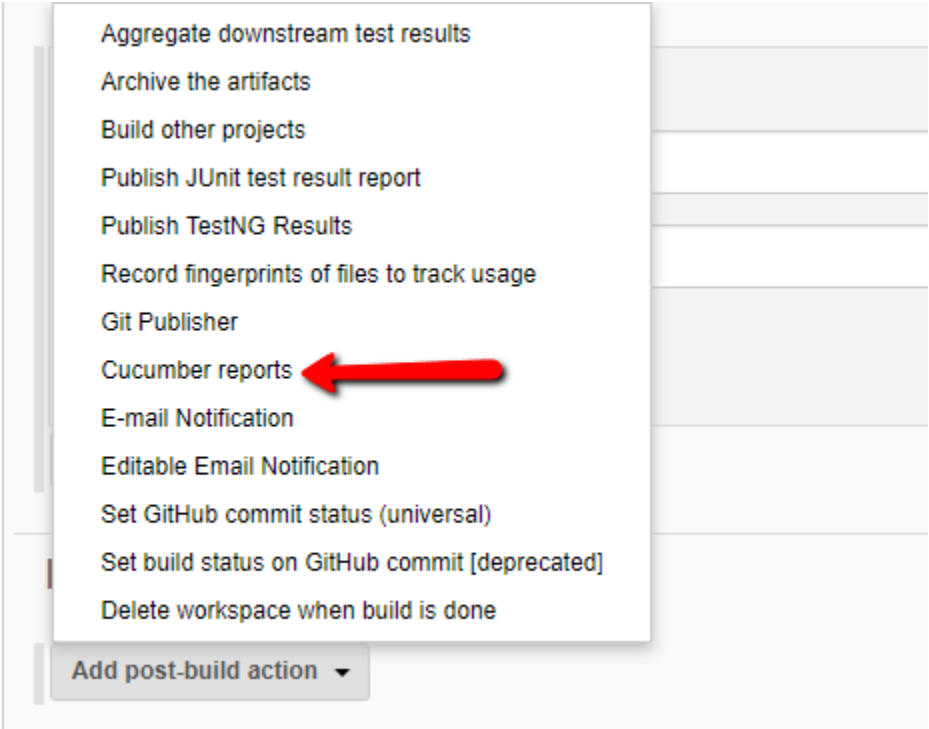
4. Configure the Maven build steps

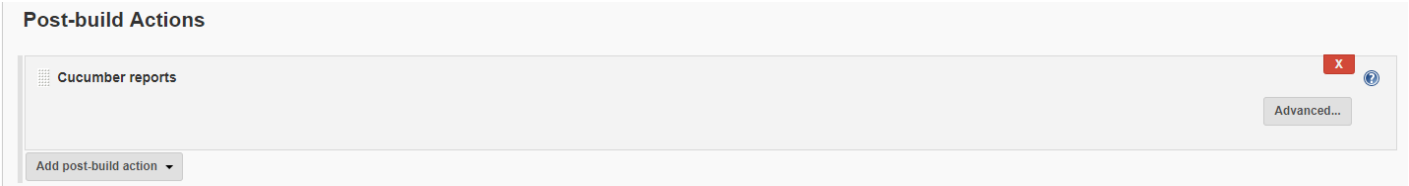


- Select Maven Version as ‘Maven’



5. Add Cucumber Reports from the `PostBuild` Actions

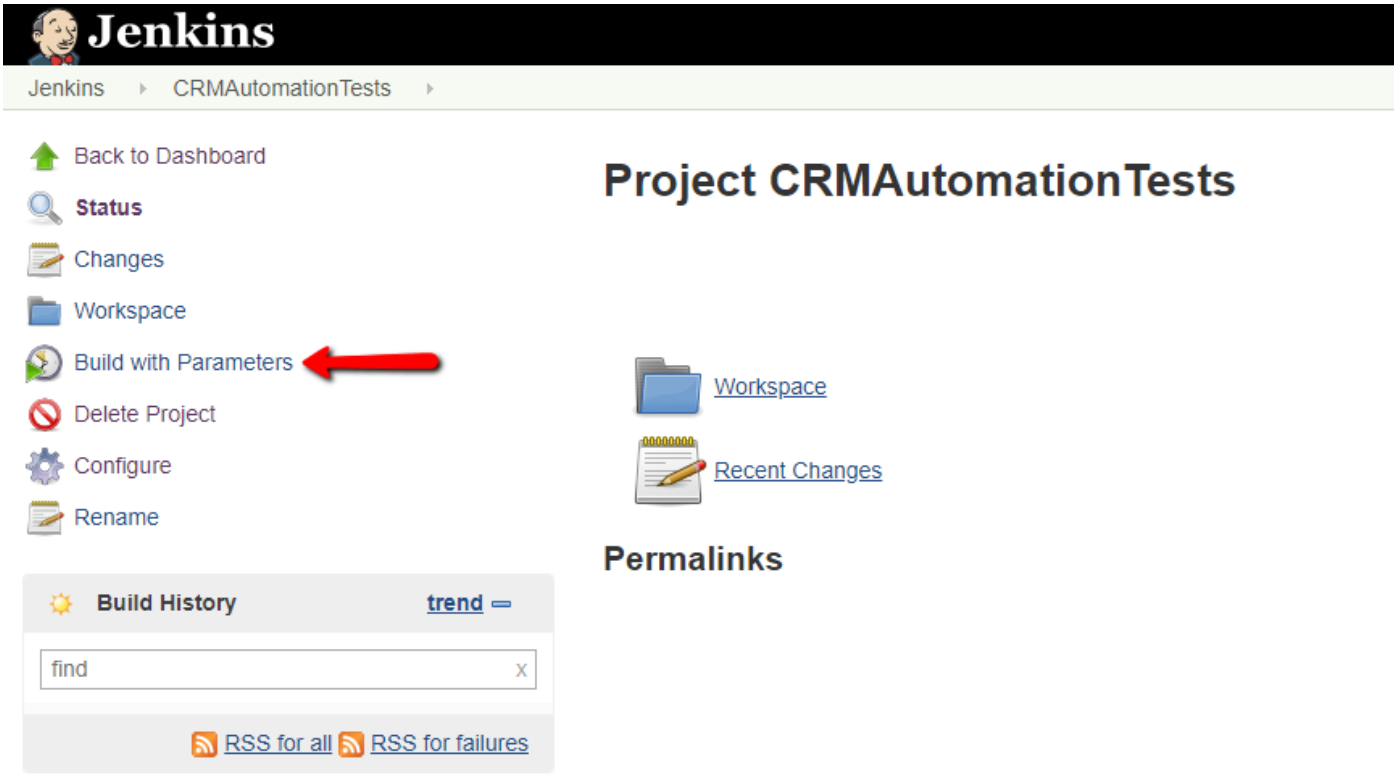




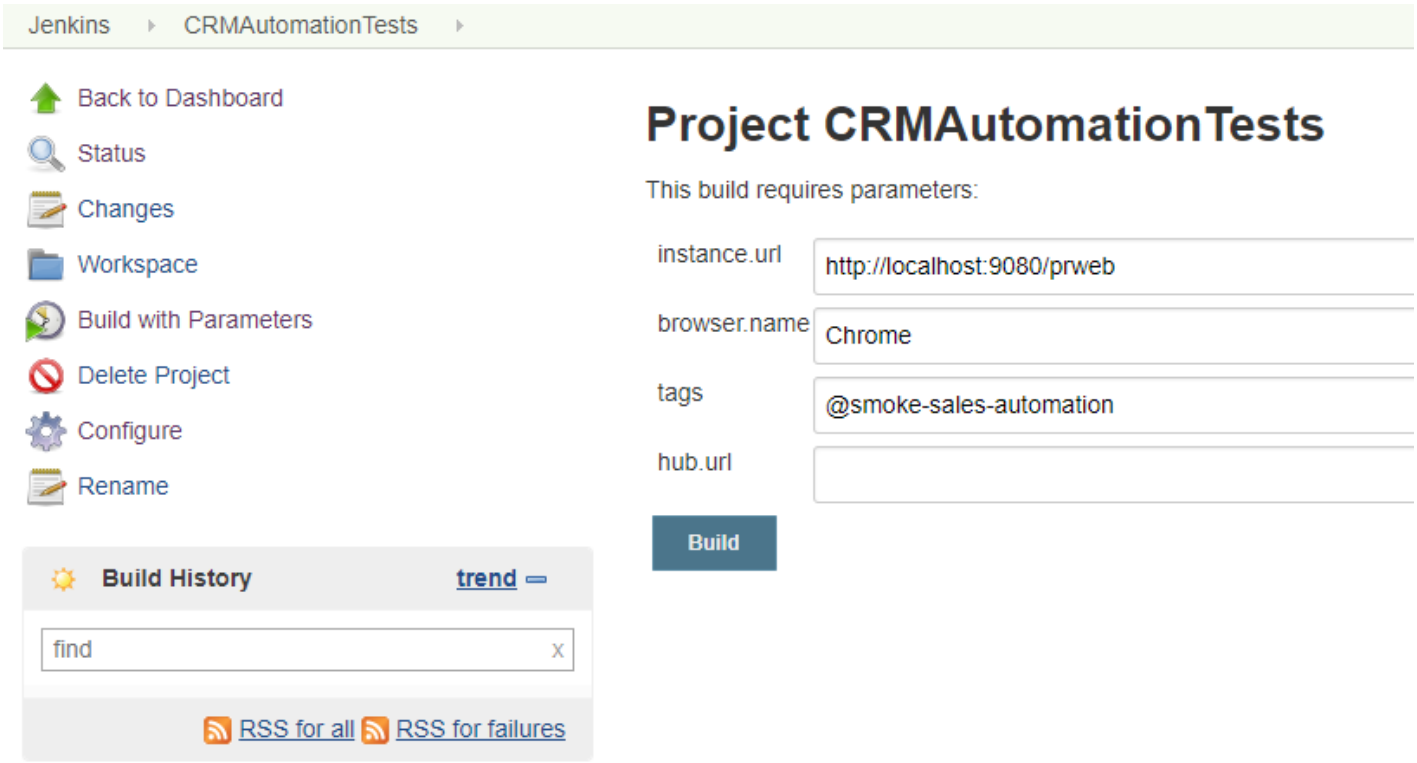
6. Save the Jenkins job

Running the Tests

1. Open Jenkins
2. Go to the `CRMAutomationTests` job
3. Click **Build with Parameters** and supply the required parameters
4. Click **Build** to start the job which will run the UI tests in CBT (If the hub.url is provided, else tests run locally)



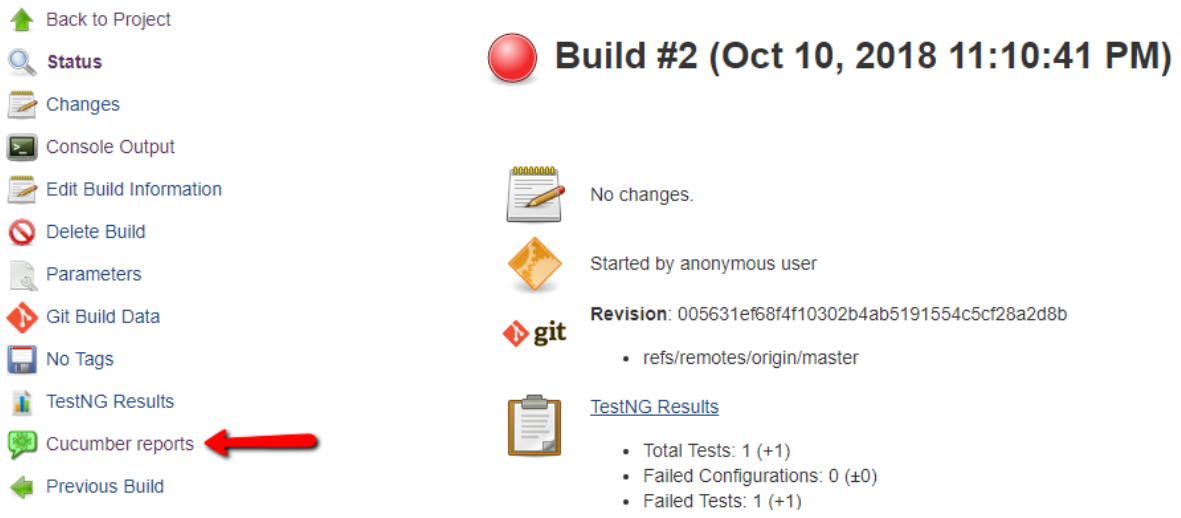
5. Provide the needed parameters: application URL, browser name and hub URL, then click **Build**.



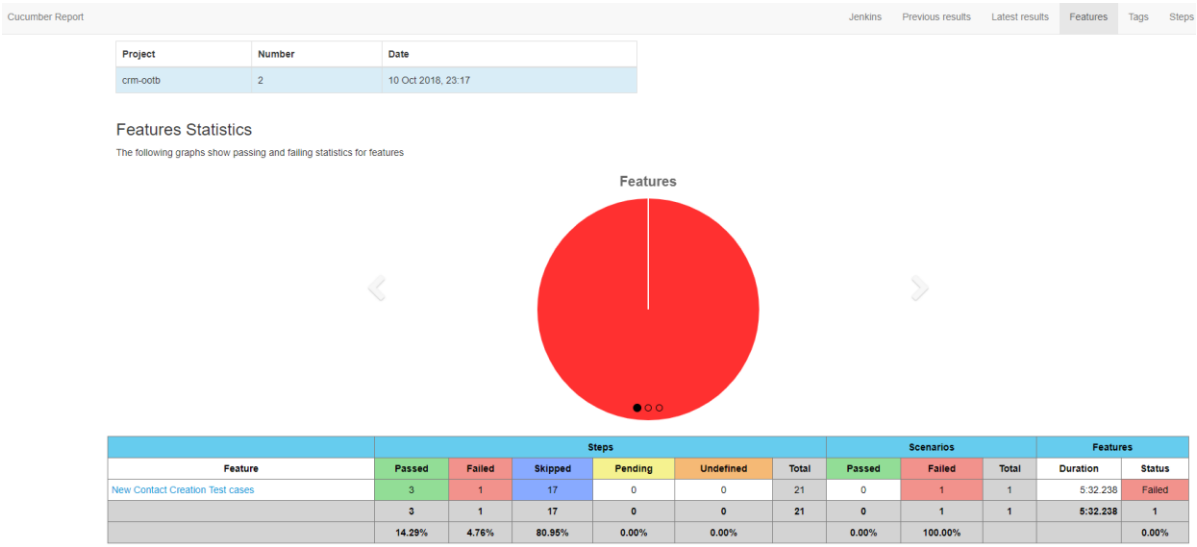
This will run your CRM Automations either locally or on CBT (depending on whether hub.url is provided or not).
To visually inspect the tests running on CBT, open your CBT account and check that the tests are running.

Viewing the Cucumber Results

Once the test runs are completed on Jenkins, you can navigate to the current build that just got completed and search for the Cucumber reports link on the left hand side menu. Click on the link to view the cucumber reports



Below is a sample report



Cucumber Report

Jenkins | Previous results | Latest results | Features | Tags | Steps

Project	Number	Date
CRMAutomationTests	1	12 Oct 2018, 14:30

Feature Report

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Demo Interaction Test cases	2	1	20	0	0	23	0	1	1	5:31.571	Failed

Tags: @demointeraction @smoke @smoke-customer-service

Feature Demo Interaction Test cases

Tags: @TC-accountswitching

Scenario Validation when Account is switched from Individual to Commercial 5:31.571

Hooks

Steps

Given a user is on login page of CS Portal	0.601
When CS operator logs in to the portal "cacsr" and "install12345!"	30.842
When CSR launches Demo Interaction for "Demo Pop - CONNOR"	5:0.127

org.openqa.selenium.TimeoutException:

Then verify the toaster pop values for connor	0.000
When CSR accepts the demo call	0.000
When Click on Add Task to launch Service Process	0.000
Then Verify all the service process items and other fields	0.000
When Launch "Address Change" service process	0.000
When select the verification questions for service cases and click on verified	0.000
Then verify "Address Change" flow is launched	0.000
Then verify that "Address Change" flow is launched with dialog	0.000
When change the address and other fields and submit	0.000
When check additional account for address change and submit	0.000
Then check the dialog in in confirm screen and in progress task in left nav	0.000
When confirm the changes made	0.000
Then check for completed or cancelled task "Address Change"	0.000
When launch warpup to complete the interaction	0.000

Then Verify all the service process items and other fields	0.000
When Launch "Address Change" service process	0.000
When select the verification questions for service cases and click on verified	0.000
Then verify "Address Change" flow is launched	0.000
Then verify that "Address Change" flow is launched with dialog	0.000
When change the address and other fields and submit	0.000
When check additional account for address change and submit	0.000
Then check the dialog in in confirm screen and in progress task in left nav	0.000
When confirm the changes made	0.000
Then check for completed or cancelled task "Address Change"	0.000
When launch warpup to complete the interaction	0.000

Then verify the wrapup dialog	0.000
When complete the wrap up	0.000
Then User will be navigated to the portal	0.000
When search for I- and S- items	0.000
Then verify the status	0.000
When Operator logs of the portal	0.000

Output 1

Attachment 1 (png)