InfluxDB Plugin

Plugin Information
View InfluxDB on the plugin site for more information.

Sends Jenkins build metrics into InfluxDB

Description

InfluxDB Plugin allows you to send various metrics to InfluxDB. The plugin gets its data from the other results published by Jenkins. For example, unit test results are received from the JUnit Plugin and Robot Framework results from the Robot Framework Plugin etc.

From version 1.13 onwards different plugins are listed as optional dependencies. In order to get rid of mandatory dependency errors, InfluxDB plugin must be re-installed.

Supported Metrics

Measurement	Data	Related plugin
All	• Build number • Project name • Project path	
jenkins_data	• Build health • Build status message • Build time • Build Agent name • Job duration • Build result • Build result ordinal (0=Stable, 1=Unstable, 2=Failure, 3=Not built, 4=Aborted) • Successful build boolean • Last stable build number (or 0 if never) • Last successful build number (or 0 if never) • Tests failed (unit test results from JUnit Plugin) • Tests skipped (unit test results from JUnit Plugin) • Tests total (unit test results from JUnit Plugin) • Time in queue (from Metrics plugin) • Build scheduled time • Build start time • Build measured time	JUnit Plugin Metrics Plugin
cobertura_data	• Package coverage % • Class coverage % • Line coverage % • Branch coverage % • Amount of packages • Amount of source files • Amount of classes	Cobertura Plugin

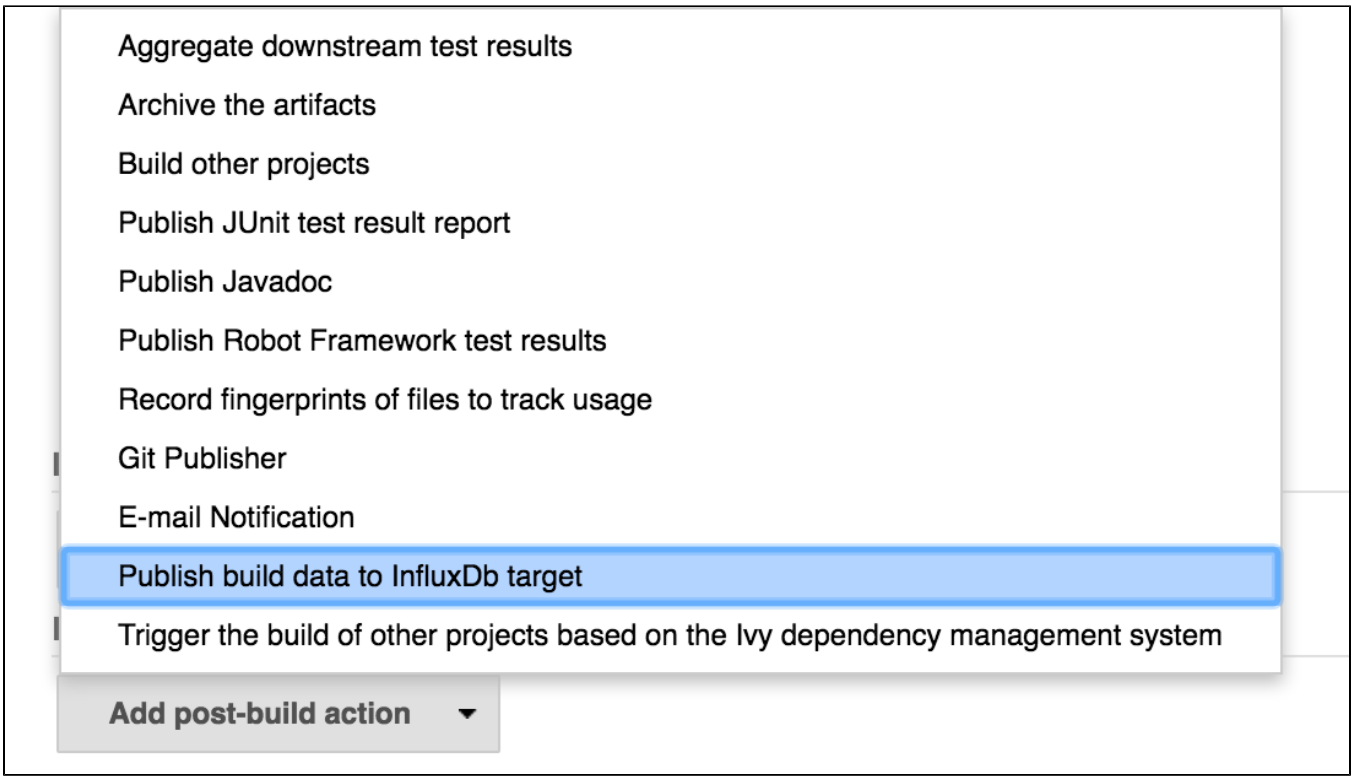
rf_results	• Test case name • Suite name • Duration • Amount of suites • Amount of passed tests • Amount of failed tests • Amount of total tests • Pass percentage of all tests • Amount of passed critical tests • Amount of failed critical tests • Total amount of critical tests • Pass percentage of critical tests	Robot Framework Plugin
suite_result	Same as rf_results except: • Pass percentage of all tests • Pass percentage of critical tests • Test case name	Robot Framework Plugin
tag_point	Same as rf_results except: • Pass percentage of all tests • Pass percentage of critical tests • Test case name • Suite name • Amount of suites Extra data sent for this measurement: • Tag name	Robot Framework Plugin
testcase_point	Same as rf_results except: • Amount of suites • Amount of total tests • Pass percentage of critical tests • Total amount of critical tests • Pass percentage of all tests	Robot Framework Plugin
jacoco_data	• Package coverage % • Class coverage % • Line coverage % • Branch coverage % • Method coverage % • Instruction coverage %	Jacoco Plugin
performance_data	• Total amount of errors • Percentage of errors • Average sample duration • Max/min sample duration • Size of samples • Total count of samples • 90 percentile	Performance Plugin
sonarqube_data	• Amount of major/minor/critical/blocker/info issues • Lines of code • Build display name	-
changelog_data	• Affected files • Commit authors • Commit messages • Commit count	-

perfpublisher_summary	• Number of total/executed/not-executed/passed/failed/success/true-false tests • Best compile time test value/name • Worst compile time test value/name • Average compile time • Best performance test value/name • Worst performance test value/name • Average performance • Best execution time test value/name • Worst execution time test value/name	Perfpublisher Plugin
perfpublisher_metric	• Metric name • Average value for metric • worst value for metric • best value for metric	Perfpublisher Plugin
perfpublisher_test	• Test name • Was test successful/executed • Test message • Compile time • Execution time • Performance	Perfpublisher Plugin
perfpublisher_test_metric	• Test name • Metric name/value/relevancy	Perfpublisher Plugin

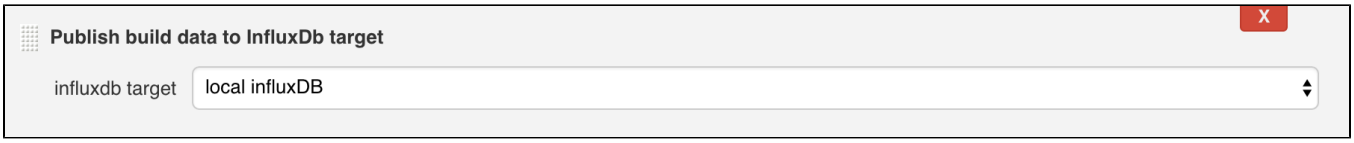
Configuration

Create a database in InfluxDB and a user with rights to your database. In Jenkins, go to Manage Jenkins > Configure System and click new influxdb target. Provide the database information to Jenkins. The url parameter needs the whole url of the influxDB database, including the http:// and the database port. Also, provide the retention policy you want the data to be stored in Influxdb (e.g. 15m or 2d for 15 minutes or 2 days respectively). By default, the retention policy is infinite. Exceptions generated by InfluxDB plugin can also be ignored by unselecting the `exposeExceptions` checkbox.

In your job, select `Publish build data to InfluxDb target` from the post-build actions.



Select the InfluxDB target you wish to publish the data.



Pipelines

The plugin can be used in pipelines by calling the step function, e.g.

```
step([$class: 'InfluxDbPublisher',
        customData: null,
        customDataMap: null,
        customPrefix: null,
        target: 'local influxDB',
                selectedTarget: 'local influxDB', // OPTIONAL, recommended if you have multiple InfluxDB
targets configured to ensure you write to correct target
                jenkinsEnvParameterTag: 'KEY=' + env.PARAM,                // OPTIONAL, custom tags
                jenkinsEnvParameterField: 'KEY=' + env.PARAM, // OPTIONAL, custom fields
                measurementName: 'myMeasurementName', // OPTIONAL, custom measurement name
                replaceDashWithUnderscore: true, // OPTIONAL, replace "-" with "_" for tag names. Default=false
        ])
```

`customData` and `customDataMap` are custom data generated during the build and not by the plugin. `customPrefix` is added to the beginning of the measurements. `target` takes the value of your `description` in your Global configuration.

NOTE! Up to release 1.10.3, pipeline was configured with the use the url and database.

```
step([$class: 'InfluxDbPublisher',
        customData: null,
        customDataMap: null,
        customPrefix: null,
        target: 'http://127.0.0.1:8086,jenkins_db'])
```

This form of configuration is not supported from version 1.11 onwards.

Pipelines don't have post-build actions, so the build result, build ordinal, and the build success boolean must be set manually. They will default to `"?"`, `5`, and `false` respectively unless set before calling the `step`. The boolean value is set according to the ordinal value and the ordinal value is fetched according to build result, so only the build result have to be set manually. Also, the build status will appear as `"?"` and the build duration might be a little off, because the build is not actually finished. If you want to get those pieces of information you need to configure the plugin separately on each job as a post-build action. The jobs can be run with, for example, the Build Pipeline Plugin to get data from all jobs to InfluxDB. Alternatively, you can insert the information in your build manually inside your Groovy script.

```
try {
    // Build things here
    if(currentBuild.result == null) {
        currentBuild.result = "SUCCESS"        // sets the ordinal as 0 and boolean to true
    }
} catch (err) {
    if(currentBuild.result == null) {
        currentBuild.result = "FAILURE"        // sets the ordinal as 4 and boolean to false
    }
    throw err
} finally {
    step([$class: 'InfluxDbPublisher', ...
}
```

From version 1.19 onwards, it's also possible to create and remove targets in pipelines.

Create a target in pipeline

```
// Get Influxdb plugin descriptor
def influxdb = Jenkins.instance.getDescriptorByType(jenkinsci.plugins.influxdb.DescriptorImpl)

// Create target
def target = new jenkinsci.plugins.influxdb.models.Target()

// Set target details
// Mandatory fields
target.description = 'myNewTarget'
target.url = 'http://192.168.99.100:8086'
target.username = 'myUsername'
target.password = 'myPassword'
target.database = 'myDatabase'
// Optional fields
target.exposeExceptions = true  // default = true
target.jobScheduledTimeAsPointsTimestamp = true  // default = false
target.usingJenkinsProxy = true // default = false
target.retentionPolicy = '1d' // default = 'autogen'
// NEW in version 1.20.1
target.replaceDashWithUnderscore = true // default = false

// Add a target by using the created target object
influxdb.addTarget(target)
influxdb.save()

// Write stuff to InfluxDB
// NOTE! If you have more targets configured in Jenkins, it's safer to add "selectedTarget" parameter to the
InfluxDB step to ensure
// data is sent to the correct target.
// step([$class: 'InfluxDbPublisher', target: 'myNewTarget', selectedTarget: 'myNewTarget', customPrefix:
'myPrefix', customData: myDataMap])

// Remove a target by using the target description field value
influxdb.removeTarget("myNewTarget")
influxdb.save()
```

Custom Data

You can create custom data in Jenkins pipelines. Custom data can be written to Influx like this:

```
def myDataMap = [:]
myDataMap['myKey'] = 'myValue'
step([$class: 'InfluxDbPublisher', target: myTarget, customPrefix: 'myPrefix', customData: myDataMap])
```

This adds the key `myKey` with a value `myValue` to a measurement called `jenkins_custom_data`.

You can also create your own measurements with customDataMaps:

```
def myDataMap1 = [:]
def myDataMap2 = [:]
def myCustomDataMap = [:]
myDataMap1["myMap1Key1"] = 11 //first value of first map
myDataMap1["myMap1Key2"] = 12 //second value of first map
myDataMap2["myMap2Key1"] = 21 //first value of second map
myDataMap2["myMap2Key2"] = 22 //second value of second map
myCustomDataMap["series1"] = myDataMap1
myCustomDataMap["series2"] = myDataMap2
step([$class: 'InfluxDbPublisher', target: myTarget, customPrefix: 'myPrefix', customDataMap: myCustomDataMap])
```

This creates measurements `series1` and `series2` and adds the keys `myMap1Key1` and `myMap1Key2` with values `11` and `12` respectively to `series1`; and `myMap2Key1` and `myMap2Key2` with values `21` and `22` respectively to `series2`.

You can also set tags for your custom measurements with either `customDataTags` or `customDataMapTags`. Tags added with `customDataTags` are added to the measurement `jenkins_custom_data`. If you use `customDataMapTags` you **must** use the same map keys as the measurement names in `customDataMap`.

You can also set a custom measurement name for your `jenkins_data` and `jenkins_custom_data` measurements. To do this you must add

```
measurementName: "measurementName_data"
```

inside your InfluxDbPublisher build step. This will change the name of your `jenkins_data` measurement into `measurementName_data` and `jenkins_custom_data` into `custom_measurmentName_data`.

Release catalogue

1.20.4 (20.2.2019)

- Re-continue support for older SonarQube versions (JENKINS-56038)

1.20.3 (5.2.2019)

- InfluxDbPublisher: avoid NullPointerException on empty targets (Pull Request 53 / JENKINS-55594)
- Fix writing usernames and passwords to be logged by InfluxDB when data is written (JENKINS-55823)

1.20.2 (14.12.2018)

- Fix support for SonarQube 7.4 and discontinue support for lower SonarQube versions (JENKINS-55009)
- Fix calculating build time in custom data points (Pull Request 52)
- Improved logging if plugin fails to connect to SonarQube (JENKINS-55032)

1.20.1 (29.11.2018)

- Don't change dashes to underscores in tag value (JENKINS-50575)
- Added tags for suite_result, testcase_point, and tag_point (JENKINS-51699)
- Fix adding multiple targets (JENKINS-54595)
- Check null values in Sonarqube URL (JENKINS-54560)

1.20 (4.10.2018)

- Fix for NullpointerException in CustomPointDataGenerator (Pull Request 47)
- Allow Targets to be set as global listeners (Pull Request 49)
- Reduce verbosity in Jenkins system logs (Pull Request 50)
- Fixed losing targets after Jenkins restart (JENKINS-53861)

1.19 (5.9.2018)

- Set influxdb servers detail from groovy scripts (Pull Request 35, Pull Request 46, JENKINS-50962)
- Add possibility to configure the scheduled job time as InfluxDB points timestamp (Pull Request 48)
- Use SONAR_HOST_URL environment if possible for SonarQube server (JENKINS-49799)

1.18 (15.8.2018)

- Adding test_name tag so that the 'group by' clause can be used for perfpublisher results (Pull Request 39)
- Add a possibility to change measurement "jenkins_data" to a custom name in pipelines (Pull request 40)
- Add additional log output for skipped point generators (Pull Request 41)
- Set a consistent timestamp for all point generation (Pull Request 42)
- Fix bug preventing setting target in pipeline step (Pull Request 43)

1.17 (29.6.2018)

- Fixed an issue with Jenkins log being cluttered with warning messages (JENKINS-49105)
- Changed minimum required Cobertura Plugin to get rid of warnings when creating a maven package
- Enable the use of environment variables as fields or tags for jenkins_data (Pull Request 32)
- Add build_scheduled_time, build_exec_time, and build_measured_time as fields in JenkinsPointGenerator (Pull Request 38)

1.16 (4.6.2018)

- Fixed issue with null custom data map tags (JENKINS-51389)
- Added "Time in queue" metric if plugin is available (Pull request 29)
- Fixed SonarQube data collection if project name has slashes (JENKINS-50763 / Pull request 30)

1.15 (11.5.2018)

- Allow to use Jenkins global proxy when connecting to Influx server (Pull request 23)
- Added project path and and agent name to jenkins_data (Pull request 26)
- Added support for custom tags (Pull request 28)

1.14 (13.2.2018)

- Added SonarQube authentication support (JENKINS-47776)
- Support for custom project name (Pull request 24)
- Fixed pipeline support if user doesn't have Robot Framework plugin (JENKINS-49308)

1.13.2 (23.1.2018)

- Fixed Sonarqube data fetching (JENKINS-48858)

1.13.1 (20.12.2017)

- Fix for running pipelines without all optional plugins installed (Pull request 25)

1.13 (20.12.2017)

- Fix for empty or null usernames (Pull request 19)
- Ignore resolved issues from Sonarqube scan (Pull request 20)
- Perfpublisher plugin support (Pull request 21)
- Fixed insertion with empty customPrefix (Pull request 22)
- Added help texts for global configuration page (Issue 47307)
- Changed other plugins to be optional dependencies

1.12.3 (30.6.2017)

- Fixed SonarQube integration (Pull request 17)
- Travis CI integration (Pull request 15)

1.12.2 (21.6.2017)

- Changed consistency level from ALL to ANY (Issue 44840)
- Changed default retention policy to "autogen" (Pull request 16)
- Temporarily disabled SonarQube integration due to multiple errors

1.12.1 (26.5.2017)

- Fixed Performance Plugin support for latest version (Issue 43539)

1.12 (15.5.2017)

- Added 90-percentile for performance_data measurement (Pull Request 12)
- Fixed Influx-Java version bump (Pull request 13)
- Added measurement changelog_data (Pull request 14)
- Fixed plugin crash if data generation fails (Jenkins-44011)

1.11 (22.3.2017)

- Updated pipeline usage
- Added SonarQube integration (Pull request 11)

1.10.3 (10.2.2017)

- `rf_tag_name` added to `tag_point`

1.10.2 (17.1.2017)

- Fixed an issue with pipelines causing a NullPointerException for several metrics (JENKINS-41067)

1.10.1 (30.12.2016)

- Changed performance measurement name from `<unit test file>` to `performance_data`

1.10 (23.12.2016)

- Exceptions are now ignorable (Pull request 3)
- New fields for `jenkins_data`: build_result, build_result_ordinal, build_successful, last_stable_build, last_successful_build(Pull request 10)

1.9 (8.11.2016)

- Added support for custom data maps (Pull request 8)
- Added support for custom data(Pull request 7)
- Removed `build_<job_name_with_dashes_changed_to_underscores>` measurement as obsolete
- Added support for Performance plugin (JENKINS-38298\)

1.8.1 (28.9.2016)

- Added custom prefixes (Pull request 6)

1.8 (13.9.2016)

- Changed Cobertura integration to get data from Cobertura Plugin
- Metrics can now be grouped by project name (Pull request 4)

1.7 (1.9.2016)

- Added JaCoCo support
- Retention Policy is now configurable in global settings
- Metrics are now sent as a bulk, instead of sent separately (Pull request 2)

1.6 (10.8.2016)

- Fixed issue with Cobertura report files on slave nodes were not found.

1.5 (5.8.2016)

- Fixed issue with selected target not stored correctly in configuration

1.4 (4.8.2016)

- Support for pipelines

- Fixed issue with multiple targets with same url, but different database
- Fixed issue with Cobertura reports in other than default location

1.3 (26.7.2016)

- First release available from the update center.