New Relic Jenkins Plugin

The New Relic Jenkins Plugin is a Jenkins plugin that provides various mechanisms for reporting information from your Jenkins server to New Relic.

The following integrations are supported.

1. Application build events (initialized, started, finalized, etc) with various build metrics reported as custom attributes are automatically pushed into Insights using the Insights Insert API.
2. APM deployment markers can be pushed to New Relic using the "Record New Relic Deployment Marker" notifier by adding a post-build step.
3. Application deployment events that mirror the APM deployment markers can be pushed into Insights whenever a deployment marker is sent using the "Create Insights Deployment Event" check box on the "Record New Relic Deployment Marker" notifier.

Installation

To install the New Relic Jenkins plugin, perform the following steps.

1. Build the New Relic Jenkins Plugin `.hpi` file or download the latest plugin release
2. Login to your Jenkins server as an adminstrator
3. Navigate to `/jenkins/pluginManager/advanced`
4. Locate the section labeled "Upload Plugin"
5. Click on the button labeled "Choose File"
6. Navigate to the New Relic Jenkins Plugin `.hpi` file and select it
7. Click on the button labeled "Upload"
8. Restart Jenkins

Usage

1. Register an Insights Event API insert key New Relic
2. Install the plugin Jenkins
3. Setup the required credentials Jenkins
4. Setup the proxy configuration (optional) Jenkins
5. Create custom dashboards! New Relic

Setup credentials

The Jenkins Credentials plugin is required to store the New Relic API keys used by the New Relic Jenkins plugin. To setup the required credentials, perform the following steps.

1. Login to your Jenkins server as an adminstrator
2. Navigate to `/jenkins/credentials/store/system/domain/_/`
3. Locate the section labeled "Upload Plugin"
4. Click on the button labeled "Add Credentials"
5. Ensure that "Username with password" is selected from the "Kind" dropdown menu
6. Ensure "Global (…)" is selected from the "Scope" dropdown menu
7. Enter your RPM account ID in the "Username" field
8. Enter your Insights Insert API Key in the "Password" field

9. Enter a value for the "ID" field, e.g. "insights-insert-key"
10. Optionally enter a value for the "Description" field.
11. Click on the button labeled "OK"
12. Navigate to `/jenkins/configure`
13. Scroll down to the section labeled "New Relic"
14. Select the key you just created in the dropdown labeled "Insights Insert Credentials"
15. Click on the button labeled "Save"

Setup proxy

The New Relic Jenkins Plugin supports the use of
the global Jenkins Proxy Configuration
for allowing Jenkins installations behind proxies to send data to New Relic. To setup the proxy configuration, perform the following steps.

1. Login to your Jenkins server as an adminstrator
2. Navigate to `/jenkins/pluginManager/advanced`
3. Locate the section labeled "HTTP Proxy Configuration"
4. Enter the host name of your proxy server in the field labeled "Server"
5. Enter the port of your proxy server in the field labeled "Port"
6. If your proxy server requires HTTP Proxy Authentication:
    i. Enter the proxy server username in the "User name" field
    ii. Enter the proxy server password in the "Password" field
7. Optionally enter host name patterns for connections that should *not* go through the proxy in the field labeled "No Proxy Host", one per line.
8. Click on the button labeled "Submit"

Dashboards

Once installed and configured, the New Relic Jenkins Plugin will immediately start sending build events for all builds in Jenkins to Insights. Build events are collected and reported to Insights in 1 minute harvest cycles. Insights dashboards can use the custom event type `AppBuildEvent` in NRQL queries to display Jenkins build data.

AppBuildEvent

Each application build event is represented by a custom Insights event of type `AppBuildEvent`. `AppBuildEvent`s have the following attributes.

Attribute name	Attribute description	Example value(s)
provider	Name of the CI/CD provider	Jenkins
providerVersion	Version of the CI/CD provider	2.140
buildId	The numeric build ID of the build	65
buildUrl	The relative URL of the build	job/my%20job/65
buildName	The short name of the build	#65
buildFullName	The full "display name of the build	my job #65
buildEventType	One of initialized, started, completed, finalized	started

Attribute name	Attribute description	Example value(s)
buildQueueId	The numeric queue ID on which the build runs	10
buildMessage	A human readable string for the build event	Completed build "my job #65" for job "my job"
buildResult	One of SUCCESS, FAILURE	SUCCESS
buildScheduled	Time the build was scheduled to start	April 19, 2019 16:48
buildStarted	Time the build was actually started	April 19, 2019 16:49
buildStartDelay	The delay between the build scheduled and start time, if any, in milliseconds	0
buildDuration	The total duration, in milliseconds, of the build	2,748
buildStatusSummary	The Jenkins build status	"broken since this build", "back to normal"
buildAgentName	Name of the node where the build ran	master
buildAgentDesc	Description of the node where the build ran	master
buildAgentLabels	The labels of the node where the build ran, separated by "|"	master|docker|macos
buildAgentHost	The host name or IP of the node where the build ran	master|docker|macos
jobUrl	The relative URL of the Jenkins job for the build	job/my%20job/
jobName	The short name of the Jenkins job for the build	my job
jobFullName	The full "display" name of the Jenkins job for the build	My Job
jenkinsMasterLabels	The node labels of the Jenkins master node, separated by "|"	master|docker|macos
jenkinsMasterHost	The host name or IP of the Jenkins master node	master-jenkins.myco.com

In addition to the attributes above, each job supports sending custom attributes with the `AppBuildEvent` as well as a switch to disable events for the job from being reported via the "Customize New Relic build event settings".

AppDeploymentEvent

`AppDeploymentEvent`s can be created simultaneously with APM deployment markers by selecting the "Create Insights Deployment Event" check box on the "Record New Relic Deployment Marker" notifier when creating a post-build step. `AppDeploymentEvent`s have the following attributes.

Attribute name	Attribute description	Example value(s)
appId	The APM application ID	2536781
revision	The revision string for the deployment marker	prod-master-13.3
changelog	The change log string for the deployment marker	Added bug fix for #14

Attribute name	Attribute description	Example value(s)
description	The description string for the deployment marker	The build for prod-master-13.3
user	The user string for the deployment marker	beeker@muppetmaster.com

Example NRQL queries

Below are some sample NRQL queries that can be used to visualize build event information.

Build count billboard

```
SELECT uniqueCount(buildId) AS Builds, filter(uniqueCount(buildId), WHERE buildEventType='finalized' AND buil
```

Build results timeseries

```
SELECT filter(uniqueCount(buildId), WHERE buildEventType='finalized' AND buildResult ='SUCCESS') AS 'Passed',
```

Build count by job bar chart

```
SELECT uniqueCount(buildId) AS Builds FROM AppBuildEvent FACET jobName
```

Average build duration timeseries

```
SELECT average(buildDuration), max(buildDuration), min(buildDuration) FROM AppBuildEvent TIMESERIES
```

Average build duration by job bar chart

```
SELECT average(buildDuration) FROM AppBuildEvent FACET jobName
```

Average build duration by job timeseries

```
SELECT average(buildDuration) FROM AppBuildEvent FACET jobName TIMESERIES
```

Today's builds vs yesterday's builds timeseries

```
SELECT uniqueCount(buildId) AS Builds FROM AppBuildEvent SINCE TODAY COMPARE WITH 1 DAY AGO TIMESERIES
```

Average build delay timeseries

```
SELECT average(buildDelay) FROM AppBuildEvent
```

Build message event stream

```
SELECT buildMessage FROM AppBuildEvent LIMIT 50
```

Troubleshooting

To troubleshoot issues, the first steps is creating a new Jenkins logger to view the New Relic Jenkins logs. This is accomplished as follows.

1. Login to your Jenkins server as an adminstrator
2. Navigate to `/jenkins/log`
3. Click on the button labeled "Add new log recorder"
4. In the field labeled "Name", enter "New Relic Logger"
5. Click on the button labeled "OK"
6. Click on the button labeled "Add" next to the label "Loggers"
7. In the field labeled "Logger", enter "com.newrelic.experts"
8. Select the value "ALL" from the menu labeled "Log level"
9. Click on the button labeled "Save"

Once this logger has been created, logs can be viewed as follows.

1. Login to your Jenkins server as an adminstrator
2. Navigate to `/jenkins/log`
3. Click on the link labeled "New Relic Logger"
4. This page will refresh every 30s to display detailed logging output for the New Relic Jenkins plugin

Contributions

You are welcome to send pull requests to us - however, by doing so you agree that you are granting New Relic a non-exclusive, non-revokable, no-cost license to use the code, algorithms, patents, and ideas in that code in our products if we so choose. You also agree the code is provided as-is and you provide no warranties as to its fitness or correctness for any purpose.

Build

As with most Jenkins plugins,
this plugin uses Maven to build. This plugin was built within Eclipse but that is not a requirement. It can be built and tested just with or without Eclipse. Eclipse does provide certain advantages such as the ability to do step debugging of Jenkins plugins with little to no effort.

1. Clone this repository
2. Run `mvn package`

The Jenkins HPI file will be built and placed into the `target` directory, e.g. `[project-root]/newrelic-jenkins-plugin.hpi`

Using Eclipse

The standard Eclipse IDE for Java Developers
and Eclipse IDE for Enterprise
Java Developers packages come with built-in Maven integration. The following instructions are written with the assumption that either one of these packages is being used or that Java and Maven support have been manually configured in

Eclipse. In other words, downloading and setting up Eclipse with Java and Maven support is outside the scope of this document.

Credits

The following people have contributed to this project.

- Scott DeWitt

License

TBD