Audit To Database Plugin

Plugin Information
View Audit to Database on the plugin site for more information.

The current version of this plugin may not be safe to use. Please review the following warnings before use:

- Audit to Database Plugin stores credentials in plain text
- CSRF vulnerability and missing permission check allow connecting to arbitrary databases

Download latest release from jenkins-ci.org

Download latest continuous integration release from CloudBees **.**

Plugin Info

This plugin provides database audit functionality to Jenkins. It allows recording build information to database, including the build parameters (if any), the node where the build is executed, and the user who started the build.
Jenkins activity is already being stored in XML files. However, historic activity can be easily lost due to configuration (e.g. keep only the latest "n" builds), and is not easy to report off. As Jenkins is already much more than just a continuous integration platform, this plugin will help those organisations that have strict audit and reporting requirements. For example, if a job is meant to deploy artifacts to a production environment, organisations may want to record that job's activity and report off historical data for compliance purposes, showing when the job was executed, by whom, with what parameters, and on which Jenkins slave.

Installation

Database connections are established via JDBC, so you have to ensure a valid JDBC driver for your database can be found by this plugin. This can be accomplished in two ways:

Use the Jenkins classpath

If Jenkins is running as a standalone application, you can put the JDBC driver package in the `war/WEB-INF/lib` directory. If Jenkins is running inside a J2EE container (e.g. Tomcat) you can use the container's classpath instead (consult the container's documentation for details).

Use the plugin's classpath

Regardless of whether jenkins is running as a standalone application or as a web application inside a J2EE container, you can put the JDBC driver package in `$JENKINS_HOME/plugins/audit2db/WEB-INF/lib`. This directory will be created the first time you run the plugin inside Jenkins, so if you can't see it (and assuming you have actually already installed the audit to database plugin), then try restarting Jenkins.

MS SQL Server and Integrated Authentication

If you want to use MS SQL Server with Integrated Authentication, you might want to check this article from the MSDN for troubleshooting a common situation.

This plugin has been tested with the following JDBC drivers:

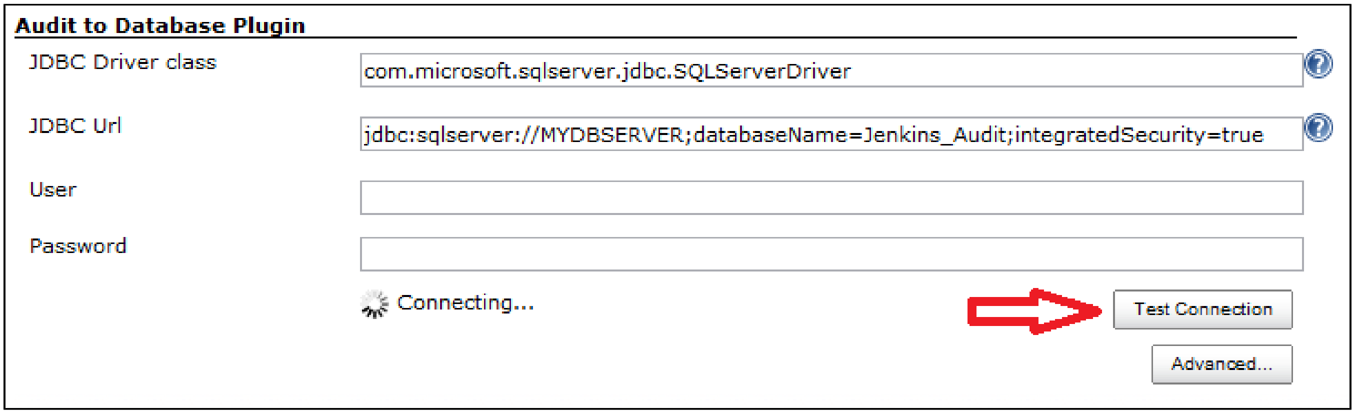
```
org.hsqldb.jdbc.JDBCDriver
oracle.jdbc.driver.OracleDriver
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

JDBC Driver Redistribution

This plugin does **NOT** ship with any JDBC drivers. Yes, it may be useful to ship supported drivers with the plugin, but it can quickly become a bit of a headache to manage redistribution licenses so we'll keep it simple instead... or at least until we get the time (and will) to study the relevant licensing terms.

Usage

Before you can use the plugin you need to set up the audit database. In the Jenkins global configuration page, enter the JDBC connection details for your audit database and test the connection.



Audit to Database Plugin

JDBC Driver class: com.microsoft.sqlserver.jdbc.SQLServerDriver

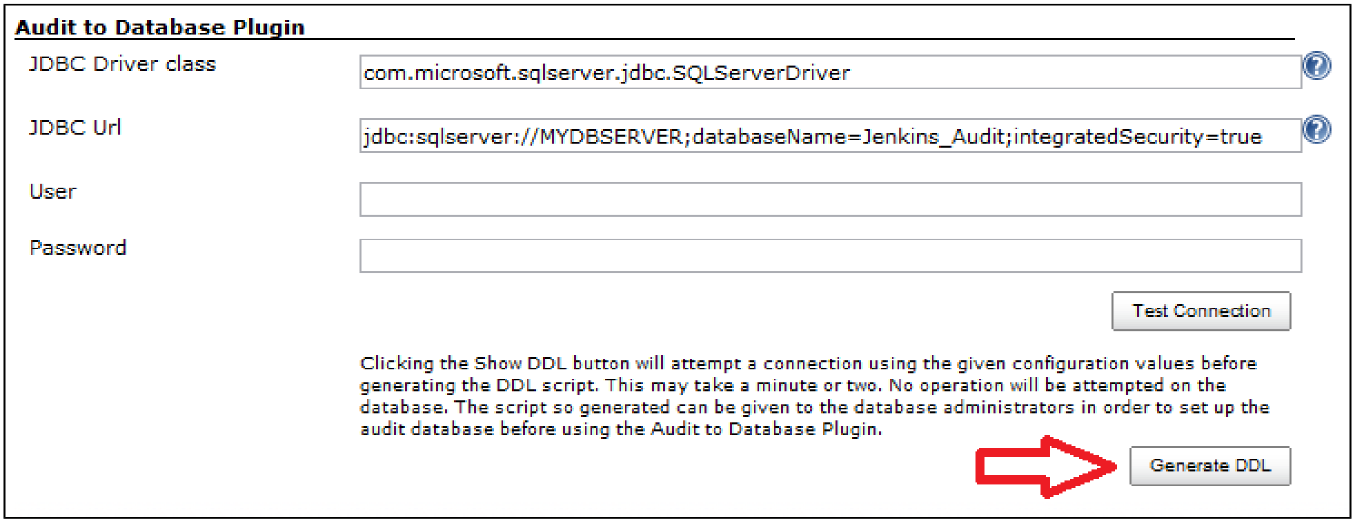
JDBC Url: jdbc:sqlserver://MYDBSERVER;databaseName=Jenkins_Audit;integratedSecurity=true

User

Password

Connecting...

If the connection is successful, click on the `Advanced` button and the `GenerateDDL` button will appear. This will allow you to generate the data definition script to set up the audit database.



Audit to Database Plugin

JDBC Driver class: com.microsoft.sqlserver.jdbc.SQLServerDriver

JDBC Url: jdbc:sqlserver://MYDBSERVER;databaseName=Jenkins_Audit;integratedSecurity=true

User

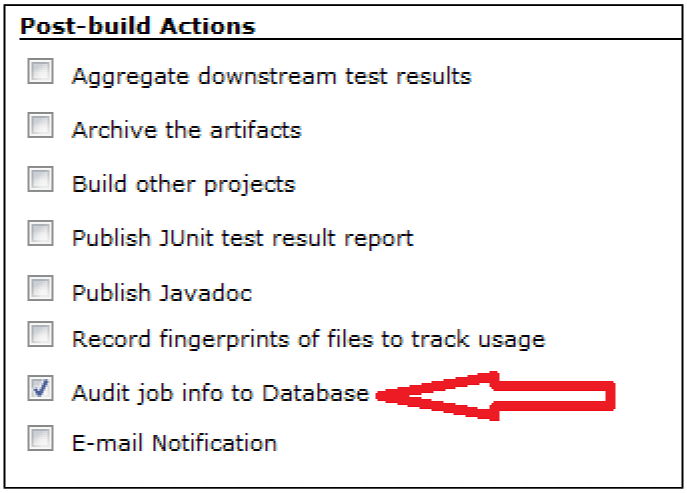
Password

Clicking the Show DDL button will attempt a connection using the given configuration values before generating the DDL script. This may take a minute or two. No operation will be attempted on the database. The script so generated can be given to the database administrators in order to set up the audit database before using the Audit to Database Plugin.

The DDL is generated as a temporary file in the plugin's home folder, and its contents displayed. The temporary file gets deleted right away. Because of this, the account used to run Jenkins must have full rights on the plugin's folder. This should already be the case, but it's something to check if an error occurs during the DDL generation. In any case, the audit database schema is given below on this page.

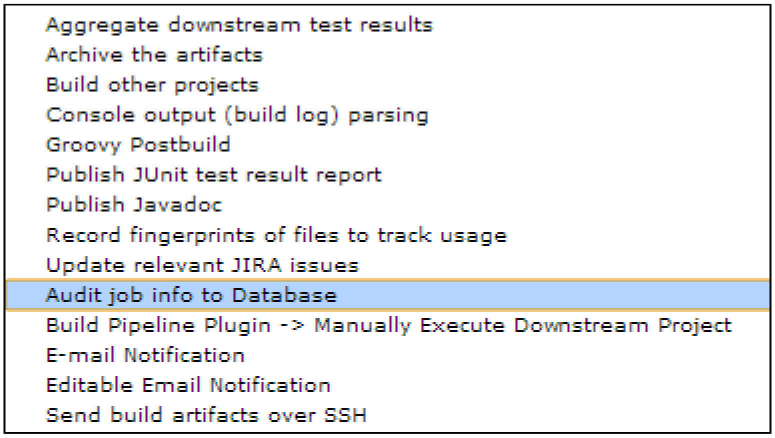
If you have any DBAs, it is a good idea to pass this script over to them now. It is also a good idea to discuss in detail your audit reporting requirements with your DBAs, so that they can configure the appropriate indexes on the audit tables and help you build your reporting queries.

Once your audit database is ready, you can enable this plugin in the `Post-build Actions` section of the job configuration. This step might look slightly differently from the screenshots below, depending on the version of Jenkins being used.



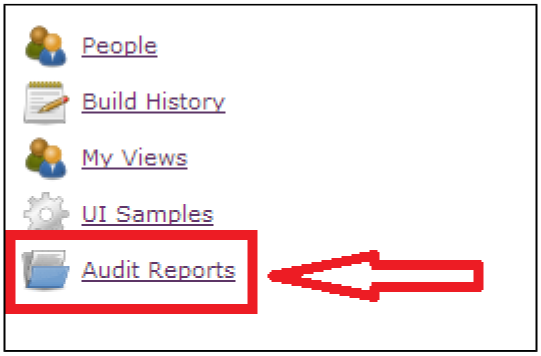
Post-build Actions

☐ Aggregate downstream test results
☐ Archive the artifacts
☐ Build other projects
☐ Publish JUnit test result report
☐ Publish Javadoc
☐ Record fingerprints of files to track usage
☑ Audit job info to Database
☐ E-mail Notification

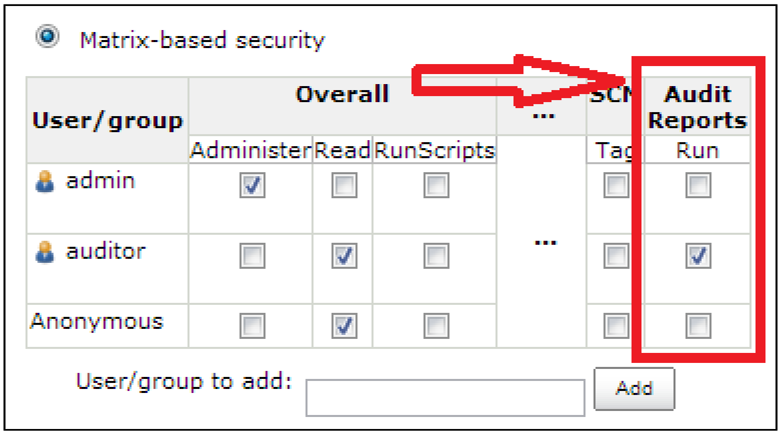

Aggregate downstream test results
Archive the artifacts
Build other projects
Console output (build log) parsing
Groovy Postbuild
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Update relevant JIRA issues
Audit job info to Database
Build Pipeline Plugin -> Manually Execute Downstream Project
E-mail Notification
Editable Email Notification
Send build artifacts over SSH

The Audit Reports

This plugin comes with some basic reports to help the audit activity.



By default, only users with administrative rights are able to access the reports. This can be changed by using the Jenkins configuration page and enabling Matrix-based security.



● Matrix-based security

User/group	Overall			...	SCM	Audit Reports
	Administer	Read	RunScripts		Tag	Run
admin	☑	□	□		□	□
auditor	□	☑	□	...	□	☑
Anonymous	□	☑	□		□	□

User/group to add: [ ] Add

The Audit Database Schema

If you need to build other reports, then you can build them outside of Jenkins by queryng the audit database. The audit database schema is extremely simple and becomes very clear once you generate the DDL script. It consists of 3 tables:

JENKINS_BUILD_DETAILS

Field Name	Type	Description
ID	String	Primary Key.
NAME	String	The name of the project that the build refers to.

FULLNAME	String	Full name of the build, includes the project name and build number. E.g. "TestProj #44".
NODE_URL	String	Foreign key on JENKINS_BUILD_NODE.
STARTDATE	Date	The build's start date.
ENDDATE	Date	The build's end date.
DURATION	Number	The build's duration in milliseconds.
USERID	String	The id of the user who started the build (NULL for Anonymous).
USERNAME	String	The name of the user who started the build.
RESULT	String	The build result.

JENKINS_BUILD_NODE

Field Name	Type	Description
URL	String	Primary Key.
NAME	String	The internal name of the node where builds get executed.
DISPLAYNAME	String	The name of the node as displayed to the user. This is not always the same as the node's internal name. For example, the master's `name` is blank, but its `displayname` is 'master'.
MASTERHOSTNAME	String	The hostname of the master that owns the build node.
MASTERADDRESS	String	The IP address of the master that owns the build node.
LABEL	String	The label associated with this node in Jenkins
DESCRIPTION	String	The description of this node in Jenkins

JENKINS_BUILD_PARAMS

Field Name	Type	Description
ID	String	Primary Key
NAME	String	The parameter name
VALUE	String	The parameter value
BUILDDETAILS_ID	String	Foreign key on JENKINS_BUILD_DETAILS

TODO List

- Support JNDI datasource.
- Write more audit reports to display in Jenkins.

Known Issues

- Plugin will not work with JDK 1.6_29 due to a bug in the JDK. See this article or this article for details.

Changelog

Version 0.5

Fixed a bug in Jobs By Date report that prevented users from applying filter criteria.

Version 0.4

Added Jobs By Param report to show jobs that have been executed with the same parameter value.

Version 0.3

Now scrambling datasource password in Jenkins configuraton XML file.

Added Jobs By Date report to show jobs executed between two dates (defaults to current month activity).

Added Audit Reports page to list all available reports.

Added audit reports RUN permission option in matrix-based Jenkins security.

Version 0.2

Added master hostname and IP address in the build node details.

Added build result in the build details.

Version 0.1

Initial working version.