Demystifying Debugging Through Logging

Ryan Smith & Michelle Fogwell

DEVOPS WORLD by CloudBees

Speakers

cloudbees®

- #1 contributor to The Jenkins Project
- Home of the largest group of Jenkins-Certified Engineers
- Offers Enterprise CI/CD products and services

Ryan Smith

- Global Escalation Manager
- Jenkins Performance and Stability Team Lead
- 10+ year industry veteran and Enterprise Java Evangelist
- ClipArt Enthusiast

Michelle Fogwell

- Senior Developer Support Engineer (DSE)
- Enthusiastic about the customer experience
- Received a 5 star rating on Animal Crossing in less than a month
- Makeup Obsessed

What We’ll Cover

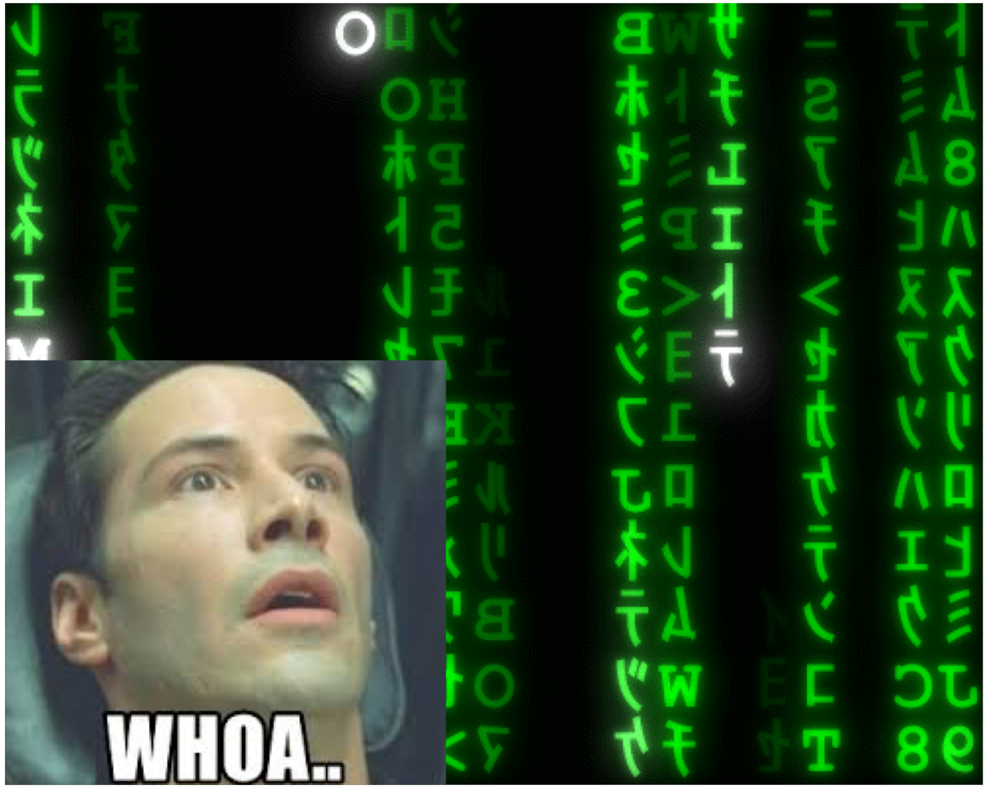
- Talk about Best Practices for Jenkins logging
- Enable you to trace log data back to the source code
- Demonstrate Real World Data use cases
- Add some powerful tools to your skillset
- Maximize your Jenkins Administration skills
- Interest in learning more about how to diagnose issues
- Want to reduce the cycle time to issue resolution

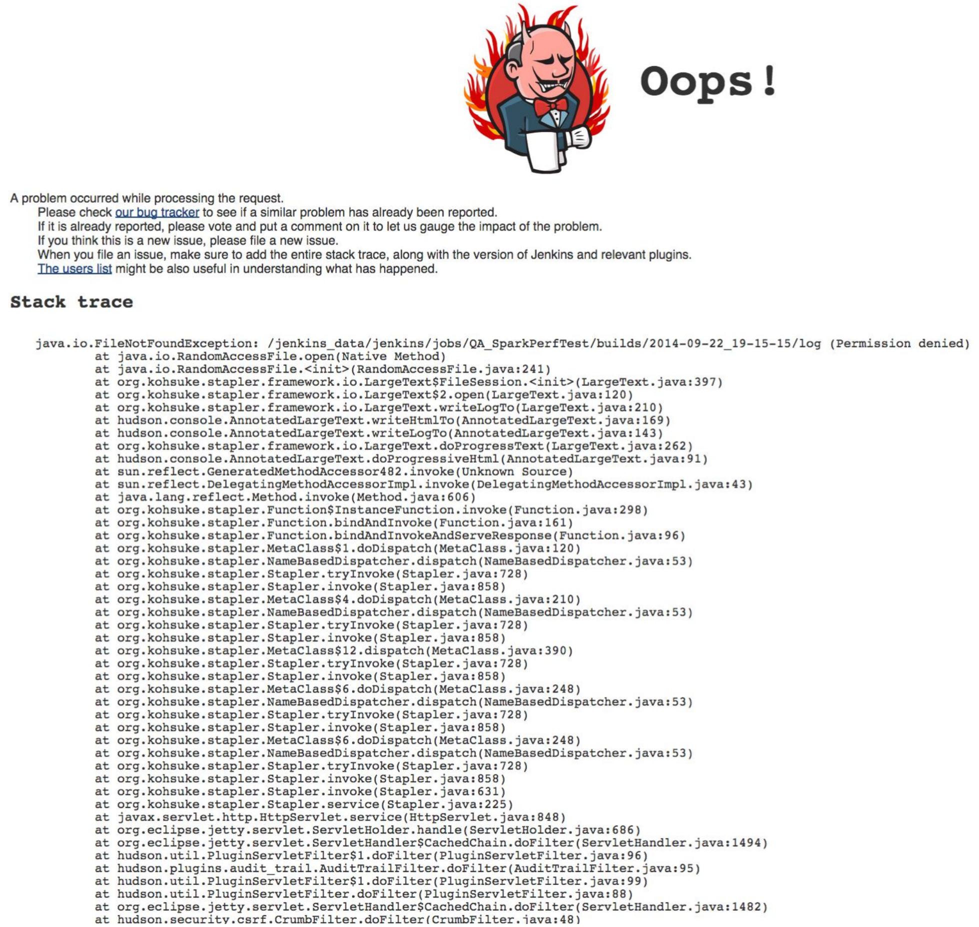
Demystifying Debugging Through Logging

Where to start with Log Diagnosis

DEVOPS WORLD by CloudBees

…Logging can be noisy and intimidating…





Oops!

A problem occurred while processing the request.
Please check our bug tracker to see if a similar problem has already been reported.
If it is already reported, please vote and put a comment on it to let us gauge the impact of the problem.
If you think this is a new issue, please file a new issue.
When you file an issue, make sure to add the entire stack trace, along with the version of Jenkins and relevant plugins.
The users list might be also useful in understanding what has happened.

Stack trace

```
java.io.FileNotFoundException: /jenkins_data/jenkins/jobs/QA_SparkPerfTest/builds/2014-09-22_19-15-15/log (Permission denied)
	at java.io.RandomAccessFile.open(Native Method)
	at java.io.RandomAccessFile.<init>(RandomAccessFile.java:241)
	at org.kohsuke.stapler.framework.io.LargeText$FileSession.<init>(LargeText.java:397)
	at org.kohsuke.stapler.framework.io.LargeText$2.open(LargeText.java:120)
	at org.kohsuke.stapler.framework.io.LargeText.writeLogTo(LargeText.java:210)
	at hudson.console.AnnotatedLargeText.writeHtmlTo(AnnotatedLargeText.java:169)
	at hudson.console.AnnotatedLargeText.writeLogTo(AnnotatedLargeText.java:143)
	at org.kohsuke.stapler.framework.io.LargeText.doProgressText(LargeText.java:262)
	at hudson.console.AnnotatedLargeText.doProgressiveHtml(AnnotatedLargeText.java:91)
	at sun.reflect.GeneratedMethodAccessor482.invoke(Unknown Source)
	at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
	at java.lang.reflect.Method.invoke(Method.java:606)
	at org.kohsuke.stapler.Function$InstanceFunction.invoke(Function.java:298)
	at org.kohsuke.stapler.Function.bindAndInvoke(Function.java:161)
	at org.kohsuke.stapler.Function.bindAndInvokeAndServeResponse(Function.java:96)
	at org.kohsuke.stapler.MetaClass$1.doDispatch(MetaClass.java:120)
	at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:53)
	at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:728)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:858)
	at org.kohsuke.stapler.MetaClass$4.doDispatch(MetaClass.java:210)
	at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:53)
	at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:728)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:858)
	at org.kohsuke.stapler.MetaClass$12.dispatch(MetaClass.java:390)
	at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:728)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:858)
	at org.kohsuke.stapler.MetaClass$6.doDispatch(MetaClass.java:248)
	at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:53)
	at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:728)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:858)
	at org.kohsuke.stapler.MetaClass$6.doDispatch(MetaClass.java:248)
	at org.kohsuke.stapler.NameBasedDispatcher.dispatch(NameBasedDispatcher.java:53)
	at org.kohsuke.stapler.Stapler.tryInvoke(Stapler.java:728)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:858)
	at org.kohsuke.stapler.Stapler.invoke(Stapler.java:631)
	at org.kohsuke.stapler.Stapler.service(Stapler.java:225)
	at javax.servlet.http.HttpServlet.service(HttpServlet.java:848)
	at org.eclipse.jetty.servlet.ServletHolder.handle(ServletHolder.java:686)
	at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1494)
	at hudson.util.PluginServletFilter$1.doFilter(PluginServletFilter.java:96)
	at hudson.plugins.audit_trail.AuditTrailFilter.doFilter(AuditTrailFilter.java:95)
	at hudson.util.PluginServletFilter$1.doFilter(PluginServletFilter.java:99)
	at hudson.util.PluginServletFilter.doFilter(PluginServletFilter.java:88)
	at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1482)
	at hudson.security.csrf.CrumbFilter.doFilter(CrumbFilter.java:48)
```

Logs:

Jenkins Logs:

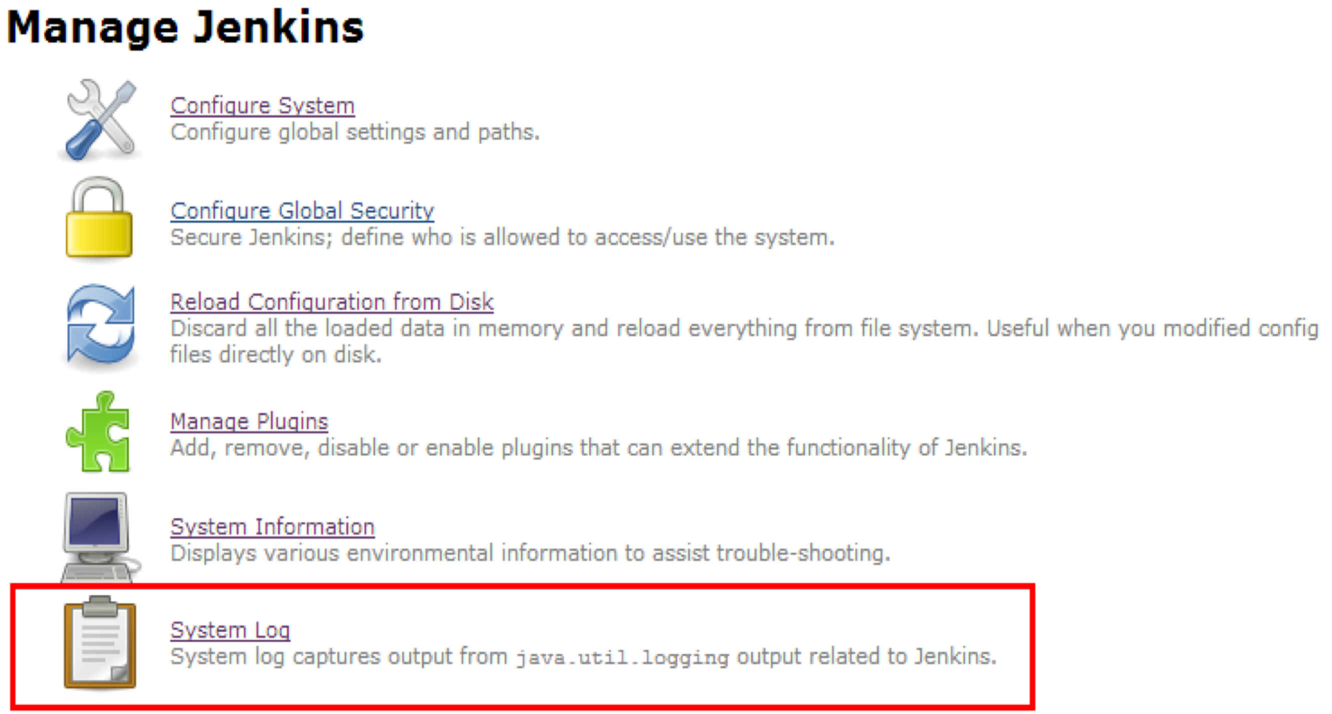
- Configurable
- Dependant upon operating system
- Support Core Plugin adds logging to filesystem in a defined location
- JVM logging added via Best Practices

Default Locations:

- Linux:
  - /var/log/jenkins/jenkins.log
- Windows:
  - %JENKINS_HOME%/jenkins.out
- Docker: **docker logs containerId**
  - (Or defaulted to underlying OS)

System Logs:

- Jenkins uses **`java.util.logging`** for logging
- By default sends every log above **INFO** to stdout
- Most servlet containers alter this behavior,
- Servlet containers bundle all the logs into a single place.



Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

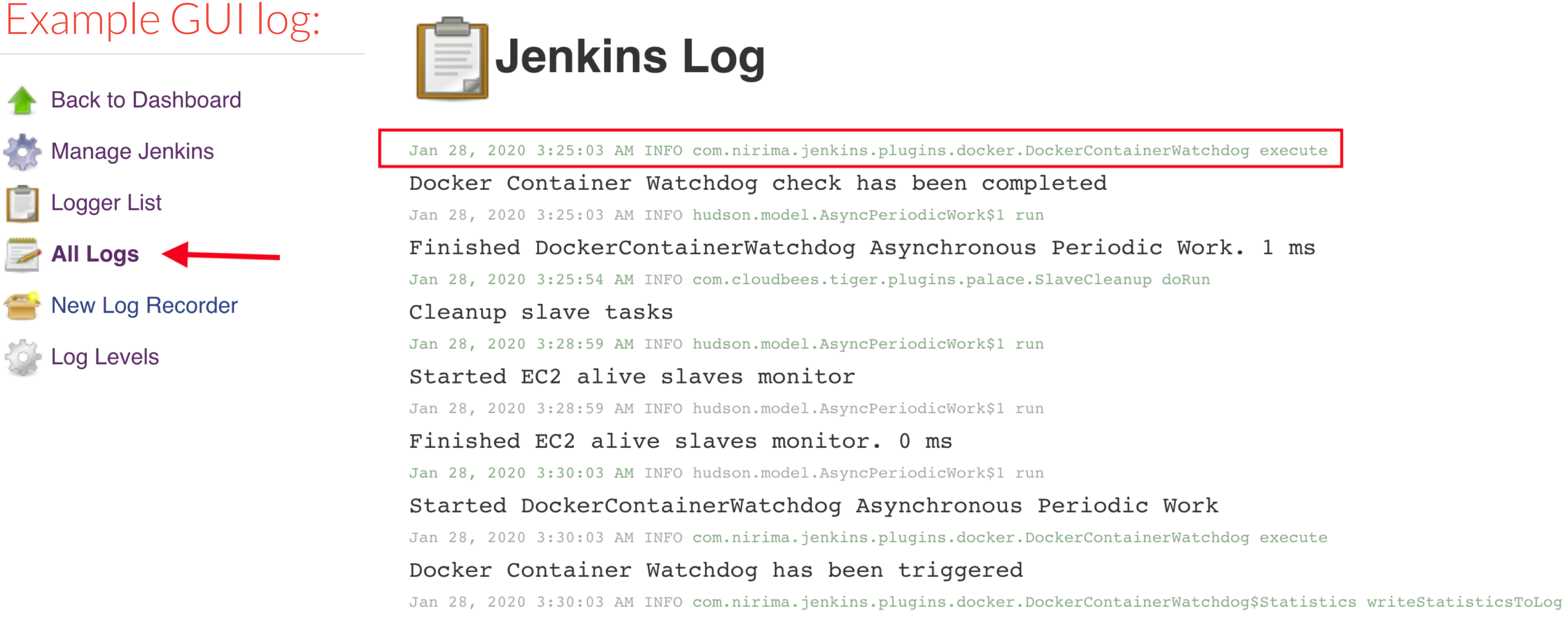
System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Jenkins is equipped with a GUI for configuring/collecting/reporting log records of your choosing!

System Logs:

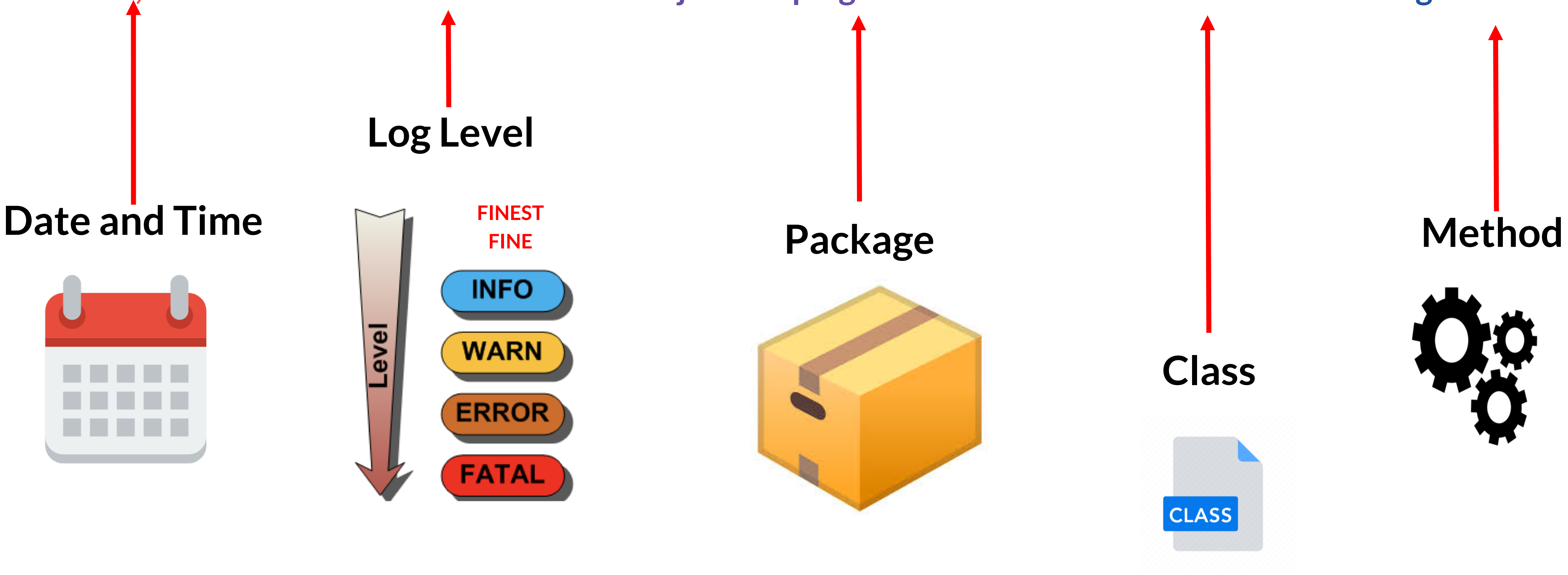
Example GUI log:



```
Jan 28, 2020 3:25:03 AM INFO com.nirima.jenkins.plugins.docker.DockerContainerWatchdog execute
Docker Container Watchdog check has been completed
Jan 28, 2020 3:25:03 AM INFO hudson.model.AsyncPeriodicWork$1 run
Finished DockerContainerWatchdog Asynchronous Periodic Work. 1 ms
Jan 28, 2020 3:25:54 AM INFO com.cloudbees.tiger.plugins.palace.SlaveCleanup doRun
Cleanup slave tasks
Jan 28, 2020 3:28:59 AM INFO hudson.model.AsyncPeriodicWork$1 run
Started EC2 alive slaves monitor
Jan 28, 2020 3:28:59 AM INFO hudson.model.AsyncPeriodicWork$1 run
Finished EC2 alive slaves monitor. 0 ms
Jan 28, 2020 3:30:03 AM INFO hudson.model.AsyncPeriodicWork$1 run
Started DockerContainerWatchdog Asynchronous Periodic Work
Jan 28, 2020 3:30:03 AM INFO com.nirima.jenkins.plugins.docker.DockerContainerWatchdog execute
Docker Container Watchdog has been triggered
Jan 28, 2020 3:30:03 AM INFO com.nirima.jenkins.plugins.docker.DockerContainerWatchdog$Statistics writeStatisticsToLog
```

System Logs:

```
Jan 28, 2020 3:25:03 AM INFO com.nirima.jenkins.plugins.docker.DockerContainerWatchdog execute
```



Date and Time

Log Level

Level

FINEST

FINE

INFO

WARN

ERROR

FATAL

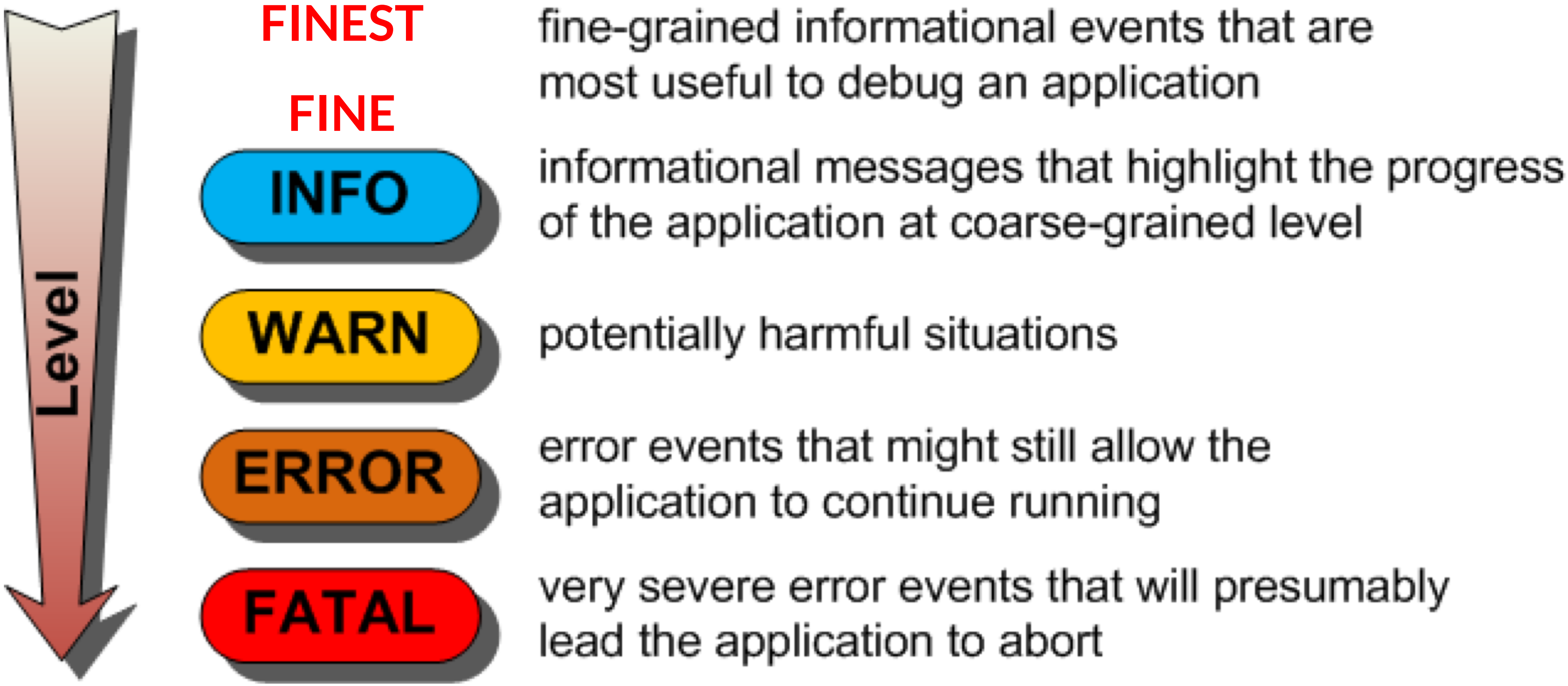
Package

Class

CLASS

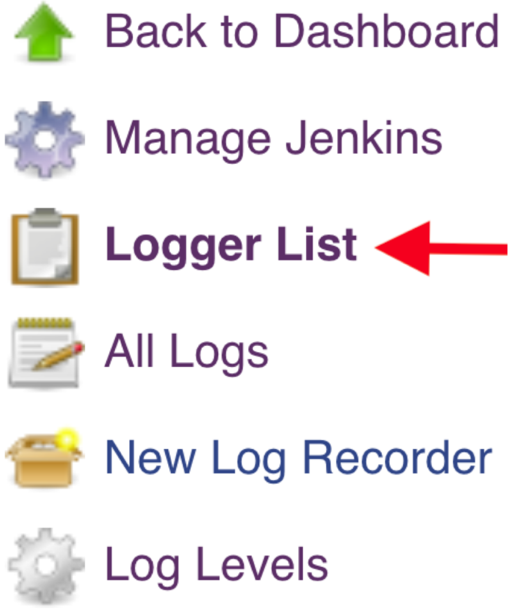
Method

Log Levels:

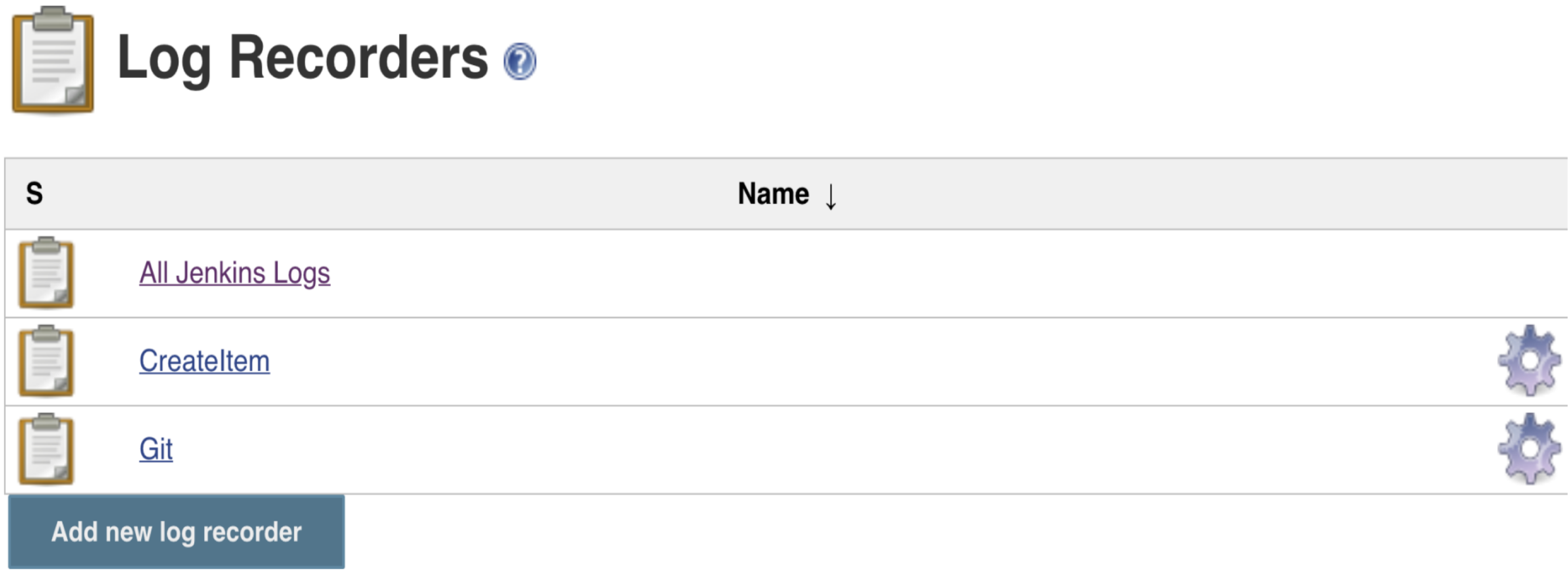


FINEST FINE	fine-grained informational events that are most useful to debug an application
INFO	informational messages that highlight the progress of the application at coarse-grained level
WARN	potentially harmful situations
ERROR	error events that might still allow the application to continue running
FATAL	very severe error events that will presumably lead the application to abort

System Logs:



Example Logger List:



Log Recorders

S	Name ↓
	All Jenkins Logs
	CreateItem
	Git

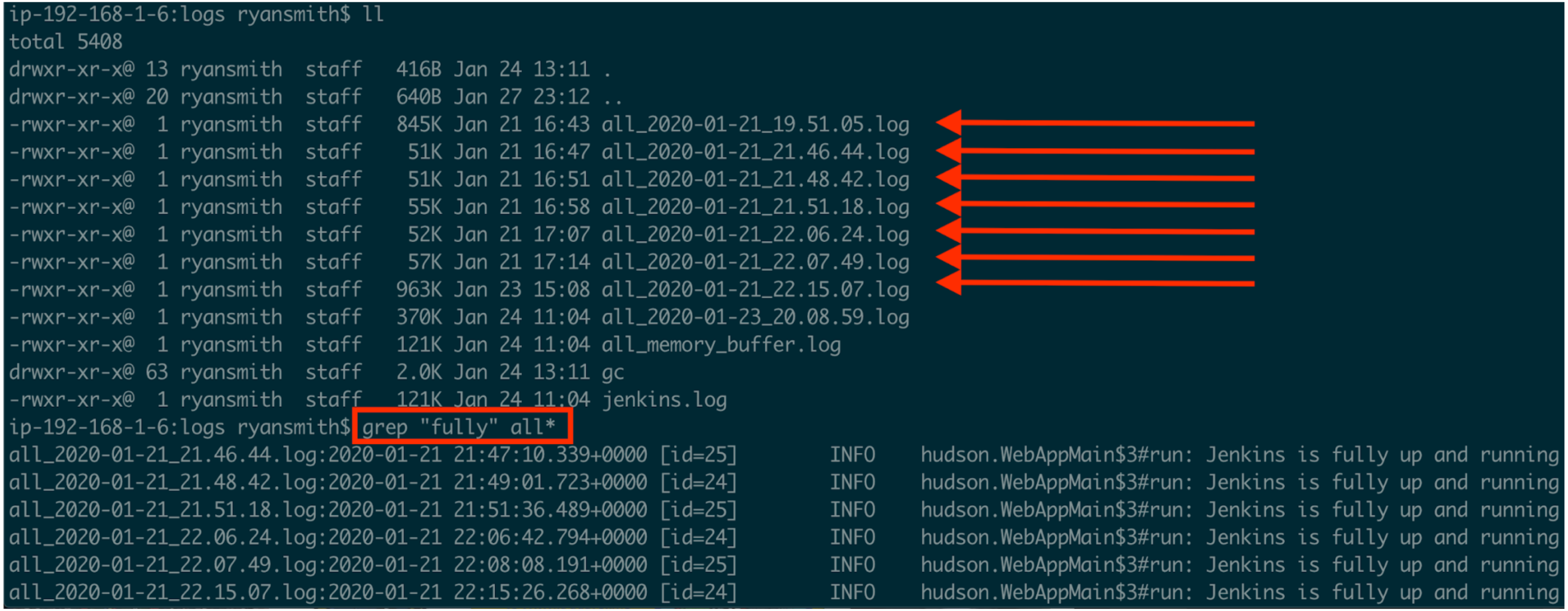
System Logs:

What’s missing from the GUI?

- GUI is limited to browser search
- Does not tail
- Limited to browser refresh
- Cannot aggregate to a log server e.g. Splunk
- Cannot efficiently search for timestamp
- Limited historic display
- Using a grep search is infinitely more effective

System Logs:

Powerful grep search example:



```
ip-192-168-1-6:logs ryansmith$ ll
total 5408
drwxr-xr-x@ 13 ryansmith  staff   416B Jan 24 13:11 .
drwxr-xr-x@ 20 ryansmith  staff   640B Jan 27 23:12 ..
-rwxr-xr-x@  1 ryansmith  staff   845K Jan 21 16:43 all_2020-01-21_19.51.05.log
-rwxr-xr-x@  1 ryansmith  staff    51K Jan 21 16:47 all_2020-01-21_21.46.44.log
-rwxr-xr-x@  1 ryansmith  staff    51K Jan 21 16:51 all_2020-01-21_21.48.42.log
-rwxr-xr-x@  1 ryansmith  staff    55K Jan 21 16:58 all_2020-01-21_21.51.18.log
-rwxr-xr-x@  1 ryansmith  staff    52K Jan 21 17:07 all_2020-01-21_22.06.24.log
-rwxr-xr-x@  1 ryansmith  staff    57K Jan 21 17:14 all_2020-01-21_22.07.49.log
-rwxr-xr-x@  1 ryansmith  staff   963K Jan 23 15:08 all_2020-01-21_22.15.07.log
-rwxr-xr-x@  1 ryansmith  staff   370K Jan 24 11:04 all_2020-01-23_20.08.59.log
-rwxr-xr-x@  1 ryansmith  staff   121K Jan 24 11:04 all_memory_buffer.log
drwxr-xr-x@ 63 ryansmith  staff   2.0K Jan 24 13:11 gc
-rwxr-xr-x@  1 ryansmith  staff   121K Jan 24 11:04 jenkins.log
ip-192-168-1-6:logs ryansmith$ grep "fully" all*
all_2020-01-21_21.46.44.log:2020-01-21 21:47:10.339+0000 [id=25]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
all_2020-01-21_21.48.42.log:2020-01-21 21:49:01.723+0000 [id=24]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
all_2020-01-21_21.51.18.log:2020-01-21 21:51:36.489+0000 [id=25]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
all_2020-01-21_22.06.24.log:2020-01-21 22:06:42.794+0000 [id=24]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
all_2020-01-21_22.07.49.log:2020-01-21 22:08:08.191+0000 [id=25]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
all_2020-01-21_22.15.07.log:2020-01-21 22:15:26.268+0000 [id=24]        INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
```

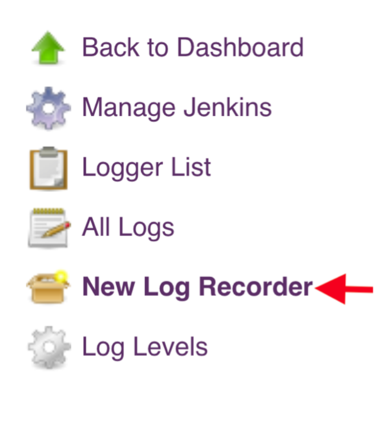
DEVOPSWORLD
by CloudBees

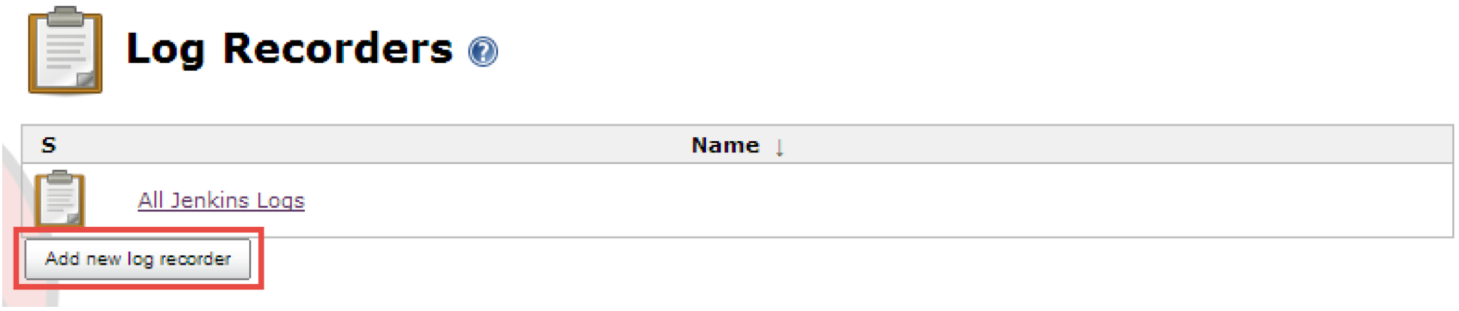
Demystifying Debugging Through Logging

Enabling Custom Loggers

DEVOPS WORLD by CloudBees

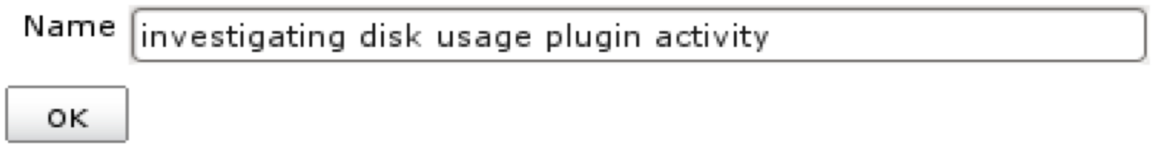
Enable Custom Loggers



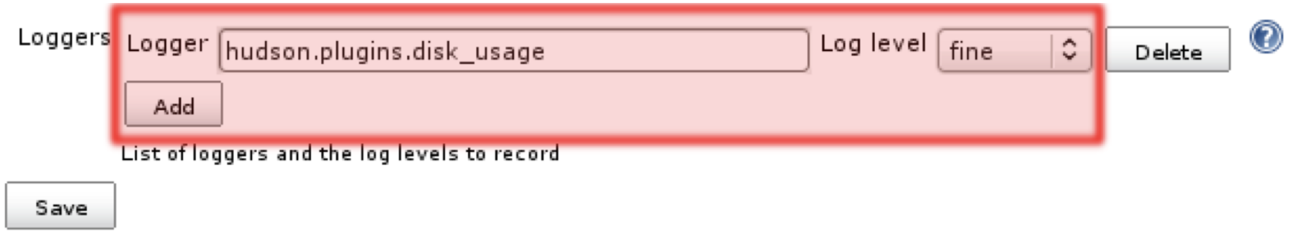

Log Recorders

S	Name ↓
	All Jenkins Logs

- **Create a log name that makes sense:**


Name: investigating disk usage plugin activity

- **Add the relevant logging class:**

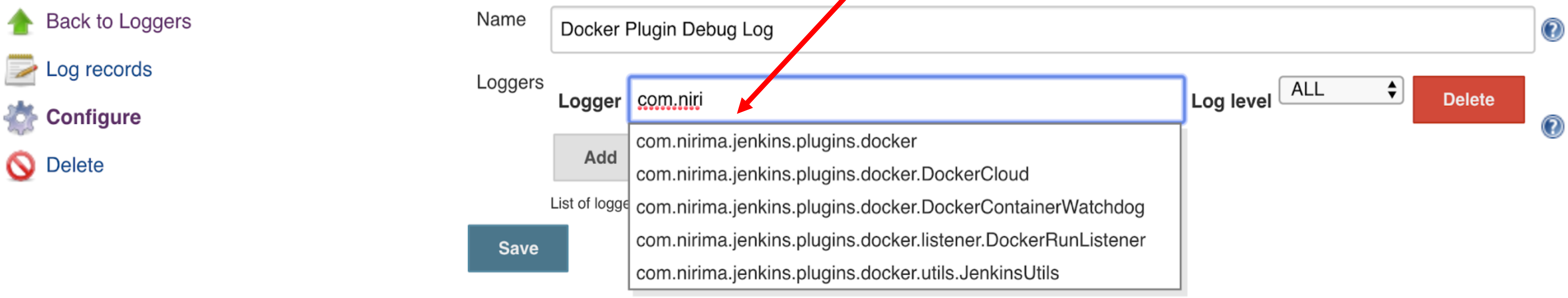

Loggers — Logger: hudson.plugins.disk_usage — Log level: fine

List of loggers and the log levels to record

Enable Custom Loggers

Jan 28, 2020 3:25:03 AM INFO com.nirima.jenkins.plugins.docker.DockerContainerWatchdog execute

- Note the Autocomplete Function!



Name: Docker Plugin Debug Log

Loggers

Logger: com.niri

Log level: ALL

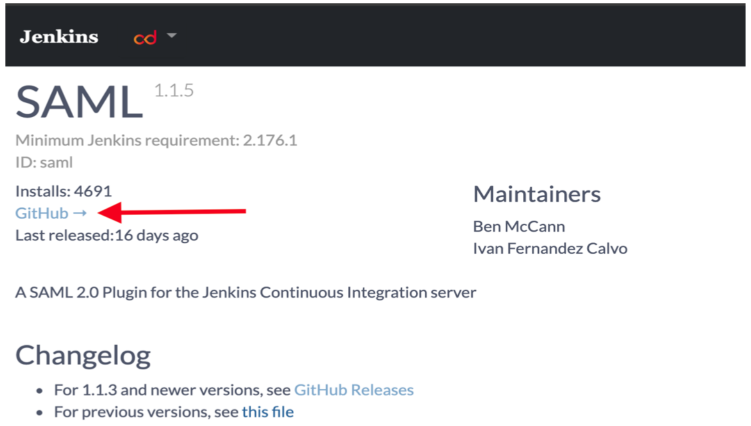
```
com.nirima.jenkins.plugins.docker
com.nirima.jenkins.plugins.docker.DockerCloud
com.nirima.jenkins.plugins.docker.DockerContainerWatchdog
com.nirima.jenkins.plugins.docker.listener.DockerRunListener
com.nirima.jenkins.plugins.docker.utils.JenkinsUtils
```

List of logge

Enable Custom Loggers

How to find a package:

Step 1: Visit the Plugin Page in the Plugin Site and follow the Github Link:



SAML 1.1.5

Minimum Jenkins requirement: 2.176.1
ID: saml

Installs: 4691
GitHub →
Last released:16 days ago

Maintainers

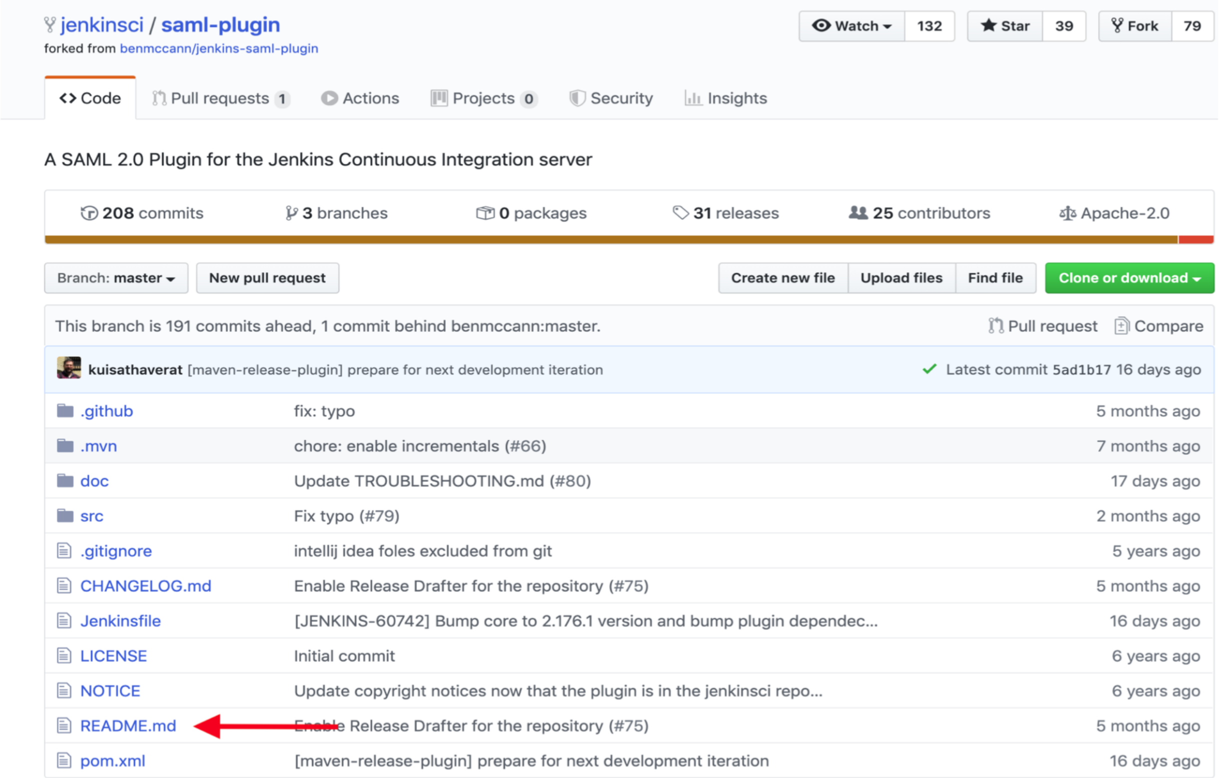
Ben McCann
Ivan Fernandez Calvo

A SAML 2.0 Plugin for the Jenkins Continuous Integration server

Changelog

- For 1.1.3 and newer versions, see GitHub Releases
- For previous versions, see this file

Step 2: Check the Readme



jenkinsci / saml-plugin
forked from benmccann/jenkins-saml-plugin

A SAML 2.0 Plugin for the Jenkins Continuous Integration server

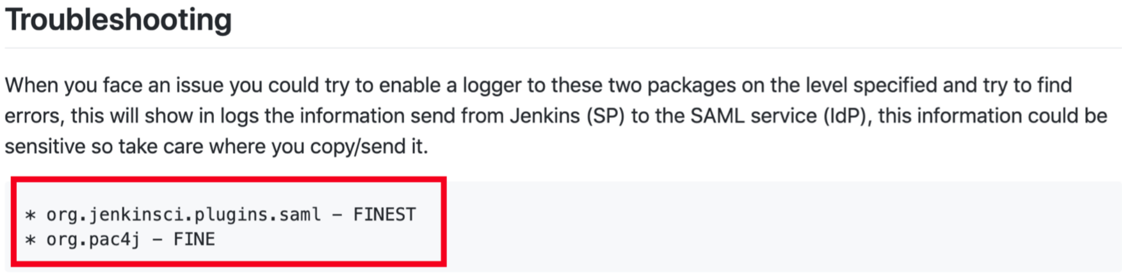
208 commits · 3 branches · 0 packages · 31 releases · 25 contributors · Apache-2.0

This branch is 191 commits ahead, 1 commit behind benmccann:master.

kuisathaverat [maven-release-plugin] prepare for next development iteration — Latest commit 5ad1b17 16 days ago

.github	fix: typo	5 months ago
.mvn	chore: enable incrementals (#66)	7 months ago
doc	Update TROUBLESHOOTING.md (#80)	17 days ago
src	Fix typo (#79)	2 months ago
.gitignore	intellij idea foles excluded from git	5 years ago
CHANGELOG.md	Enable Release Drafter for the repository (#75)	5 months ago
Jenkinsfile	[JENKINS-60742] Bump core to 2.176.1 version and bump plugin dependec…	16 days ago
LICENSE	Initial commit	6 years ago
NOTICE	Update copyright notices now that the plugin is in the jenkinsci repo…	6 years ago
README.md	Enable Release Drafter for the repository (#75)	5 months ago
pom.xml	[maven-release-plugin] prepare for next development iteration	16 days ago

Step 3: Read the Manual:



Troubleshooting

When you face an issue you could try to enable a logger to these two packages on the level specified and try to find errors, this will show in logs the information send from Jenkins (SP) to the SAML service (IdP), this information could be sensitive so take care where you copy/send it.

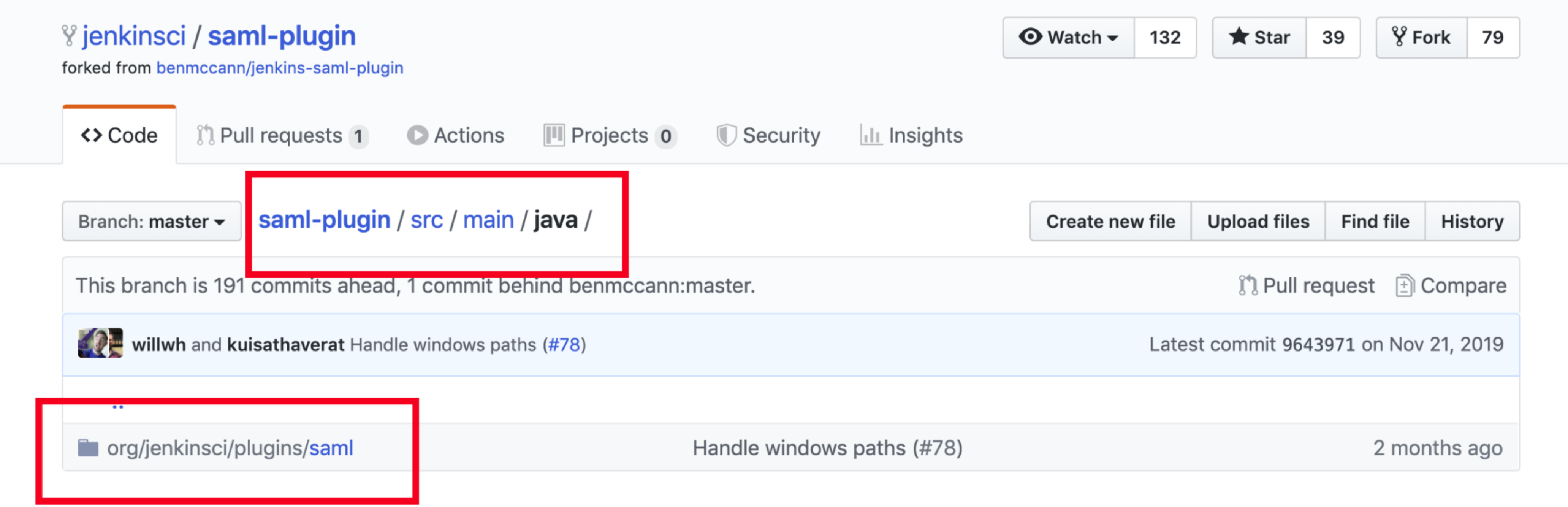
```
* org.jenkinsci.plugins.saml - FINEST
* org.pac4j - FINE
```

DEVOPSWORLD
by CloudBees

Enable Custom Loggers

How to find a package:

If all else fails, explore the repo



jenkinsci / saml-plugin

forked from benmccann/jenkins-saml-plugin

saml-plugin / src / main / java /

This branch is 191 commits ahead, 1 commit behind benmccann:master.

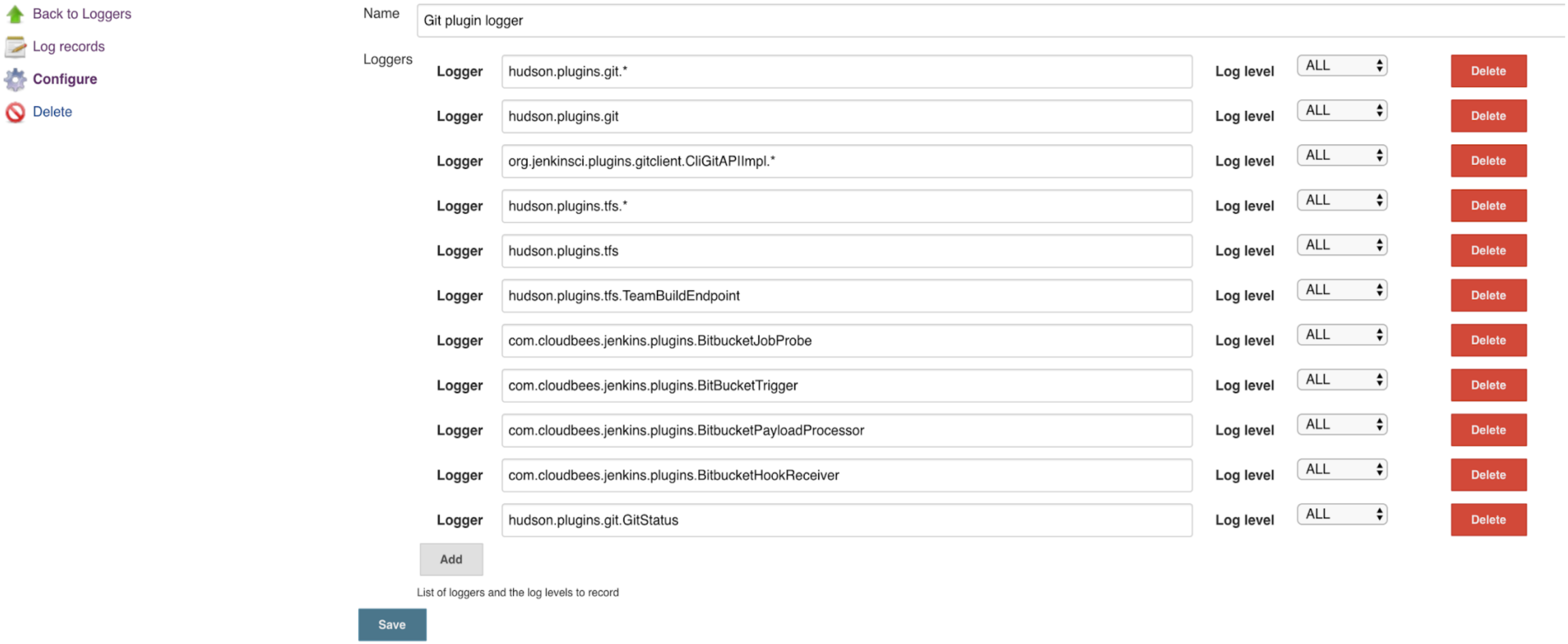
willwh and kuisathaverat Handle windows paths (#78) — Latest commit 9643971 on Nov 21, 2019

..

org/jenkinsci/plugins/saml — Handle windows paths (#78) — 2 months ago

Enable Custom Loggers

You can log multiple classes within the same logger



Name: Git plugin logger

Loggers

Logger: hudson.plugins.git.*	Log level: ALL
Logger: hudson.plugins.git	Log level: ALL
Logger: org.jenkinsci.plugins.gitclient.CliGitAPIImpl.*	Log level: ALL
Logger: hudson.plugins.tfs.*	Log level: ALL
Logger: hudson.plugins.tfs	Log level: ALL
Logger: hudson.plugins.tfs.TeamBuildEndpoint	Log level: ALL
Logger: com.cloudbees.jenkins.plugins.BitbucketJobProbe	Log level: ALL
Logger: com.cloudbees.jenkins.plugins.BitBucketTrigger	Log level: ALL
Logger: com.cloudbees.jenkins.plugins.BitbucketPayloadProcessor	Log level: ALL
Logger: com.cloudbees.jenkins.plugins.BitbucketHookReceiver	Log level: ALL
Logger: hudson.plugins.git.GitStatus	Log level: ALL

List of loggers and the log levels to record

Enable Custom Loggers

- Create a memorable logging name e.g. Git plugin log
- Remove your custom logs when not debugging an issue
- Logging has overhead to the I/O which can affect performance
- You can bring down the JVM with excessive logging
- Be careful with setting log levels on production systems

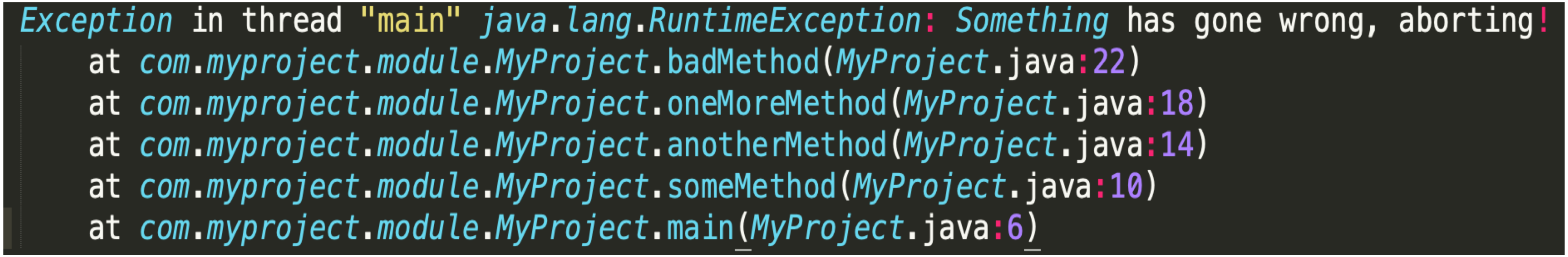
Demystifying Debugging Through Logging

Understanding a Java Stack Trace

Java Stack Trace

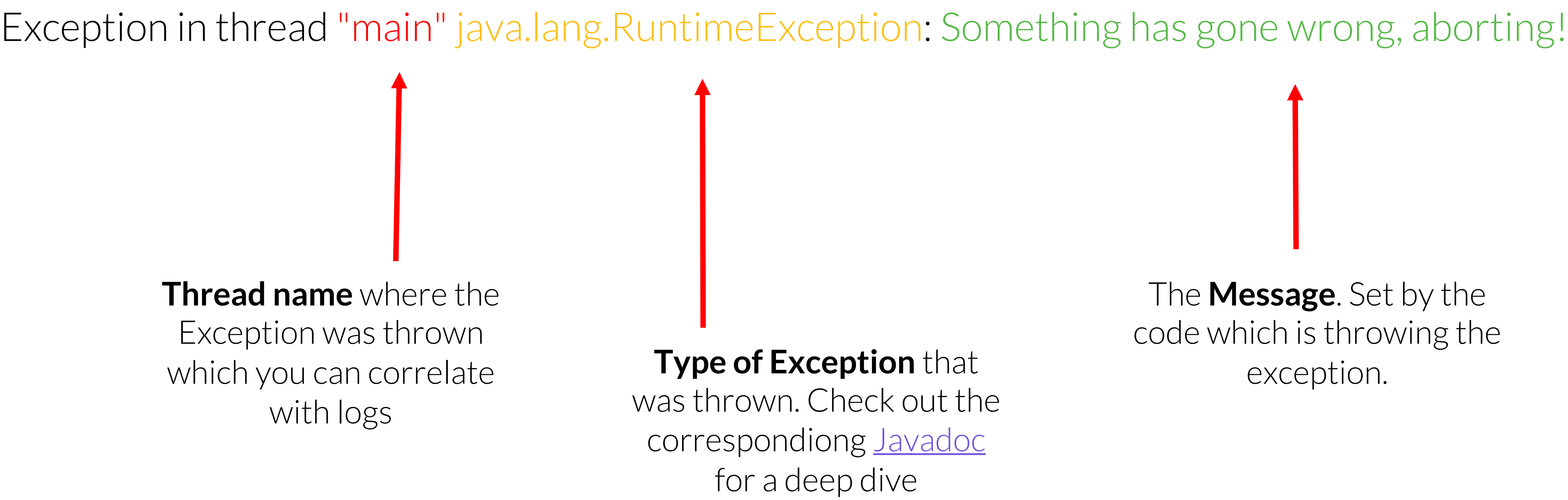
The Stacktrace contains:

- The exception type
- The message
- List of all the method calls which were in progress



```
Exception in thread "main" java.lang.RuntimeException: Something has gone wrong, aborting!
    at com.myproject.module.MyProject.badMethod(MyProject.java:22)
    at com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)
    at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)
    at com.myproject.module.MyProject.someMethod(MyProject.java:10)
    at com.myproject.module.MyProject.main(MyProject.java:6)
```

Java Stack Trace



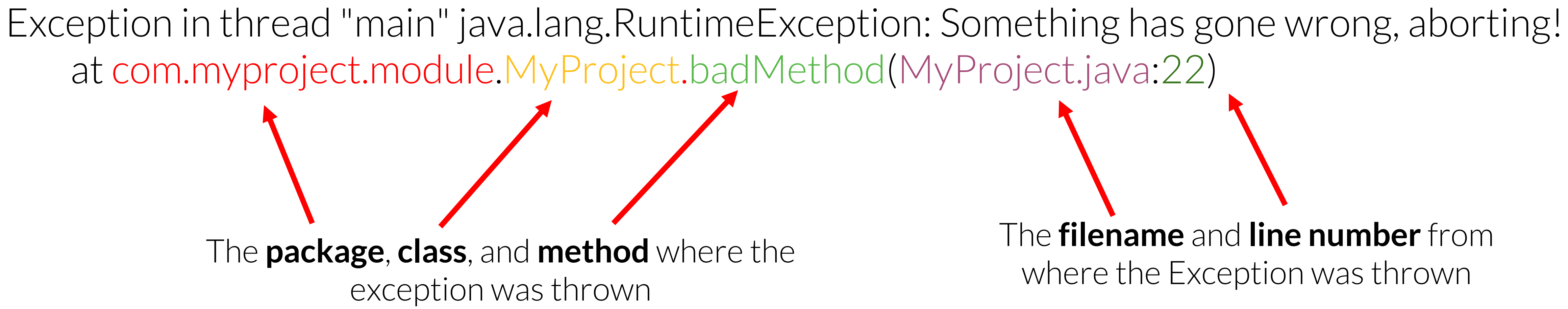
Exception in thread "main" java.lang.RuntimeException: Something has gone wrong, aborting!

Thread name where the Exception was thrown which you can correlate with logs

Type of Exception that was thrown. Check out the correspondiong Javadoc for a deep dive

The **Message**. Set by the code which is throwing the exception.

Java Stack Trace



```
Exception in thread "main" java.lang.RuntimeException: Something has gone wrong, aborting!
        at com.myproject.module.MyProject.badMethod(MyProject.java:22)
```

The **package**, **class**, and **method** where the exception was thrown

The **filename** and **line number** from where the Exception was thrown

But what part of the code called badMethod?

```
Exception in thread "main" java.lang.RuntimeException: Something has gone wrong, aborting!
        at com.myproject.module.MyProject.badMethod(MyProject.java:22)
        at com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)
```

Reading the stack trace from bottom to top traces the path!!

Java Stack Trace

```
Exception in thread "main" java.lang.RuntimeException: Something has gone wrong, aborting!
        at com.myproject.module.MyProject.badMethod(MyProject.java:22)
        at com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)
        at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)
        at com.myproject.module.MyProject.someMethod(MyProject.java:10)
        at com.myproject.module.MyProject.main(MyProject.java:6)
        …
        …
        …
        …
        …
        at java.lang.Thread.run
```

It all starts with the **main** method of the application!

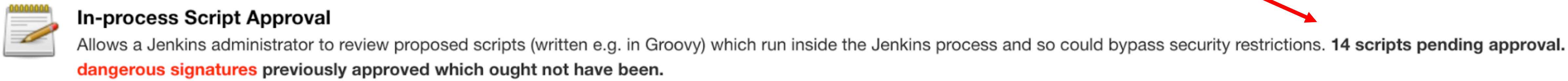
Java Stack Trace

Let’s try with a Jenkins specific stack trace!

```
2020-01-23 18:02:29.554+0000 [id=83054]         WARNING
          o.j.p.w.cps.CpsVmExecutorService#reportProblem: Unexpected exception in CPS VM thread:
CpsFlowExecution[Owner[Test-jobs/Developer/master/20:Test-jobs/Developer-JavaDoc-Test/master #20]]
org.jenkinsci.plugins.scriptsecurity.sandbox.RejectedAccessException: Scripts not permitted to use new
foo.BarConfiguration
          at org.jenkinsci.plugins.scriptsecurity.sandbox.whitelists.StaticWhitelist.rejectNew(StaticWhitelist.java:184)
          at
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SandboxInterceptor.onNewInstance(SandboxInterceptor.java:148)
          at org.kohsuke.groovy.sandbox.impl.Checker$3.call(Checker.java:197)
          at org.kohsuke.groovy.sandbox.impl.Checker.checkedConstructor(Checker.java:202)
          at org.kohsuke.groovy.sandbox.impl.Checker$checkedConstructor$1.callStatic(Unknown Source)
          …
          …
          at java.lang.Thread.run(Thread.java:748)
```

Java Stack Trace

Sure enough!
I have 14 pending approvals!



In-process Script Approval

Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions. **14 scripts pending approval.** **dangerous signatures previously approved which ought not have been.**

Demystifying Debugging Through Logging

Real World Example #1 “The JIRA Hook”

The JIRA Hook Issue

- This is an all too common scenario
- Where do you start diagnosis when communication between two systems is not working?
- How do you address OSS Community Plugins?
    - CloudBees is the #1 Contributor to the Jenkins Project
    - You can contribute to any OSS project!

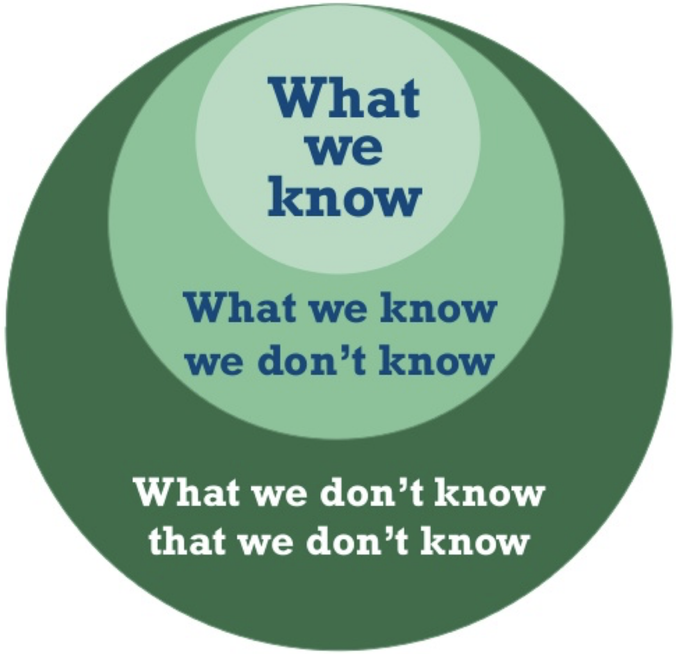
The JIRA Hook Issue

Information provided to CloudBees Support:

- “We are trying to trigger a build in jenkins from Jira based on a user comment or status change, but are unable to trigger it”
- We have configured exactly as per the documentation.
- Client Error - response code '403'

Data requested in response:

- Logs from JIRA
- Jenkins Support Bundle data provided by the Support Core Plugin
  - Plugin Versions
  - Jenkins Version
  - Logs


What we know
What we know we don’t know
What we don’t know that we don’t know

The JIRA Hook Issue

Data Anlaysis:

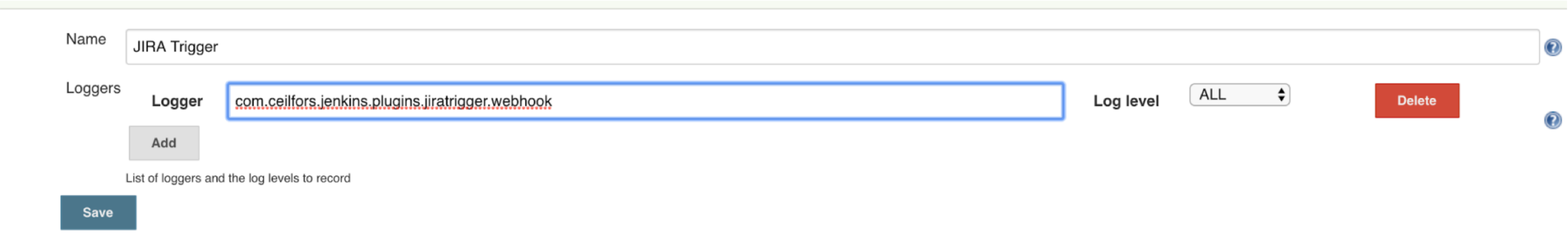
The Support Bundle log clearly show WARNING logs:

```
2020-01-06 22:29:57.737+0000 [id=2316384]   WARNING o.e.j.s.h.ContextHandler$Context#log: Error while serving https://business.com/jira-trigger-webhook-receiver/
org.codehaus.jettison.json.JSONException: JSONObject["name"] not found.
   at org.codehaus.jettison.json.JSONObject.get(JSONObject.java:360)
   at org.codehaus.jettison.json.JSONObject.getString(JSONObject.java:487)
   at com.atlassian.jira.rest.client.internal.json.JsonParseUtil.parseBasicUser(JsonParseUtil.java:192)
   at com.atlassian.jira.rest.client.internal.json.CommentJsonParser.parse(CommentJsonParser.java:37)
   at com.atlassian.jira.rest.client.internal.json.CommentJsonParser$parse.call(Unknown Source)
   at com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser.parse(WebhookCommentEventJsonParser.groovy:40)
   at com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser$parse.call(Unknown Source)
   at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.processEvent(JiraWebhook.groovy:71)
   at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook$processEvent.callCurrent(Unknown Source)
   at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.doIndex(JiraWebhook.groovy:51)
   at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
   at org.kohsuke.stapler.Function$MethodFunction.invoke(Function.java:396)
         Caused: java.lang.reflect.InvocationTargetException
```

The webhook is expecting a JSON object called name and JIRA is not passing it in the JSON request. It's possible that this requires a configuration change in JIRA or JIRA itself has made a breaking change and the plugin will need to be updated.

The JIRA Hook Issue

Confirming with Custom Logs



Name: JIRA Trigger

Loggers

Logger: com.ceilfors.jenkins.plugins.jiratrigger.webhook

Log level: ALL

After enabling the following custom logger we saw the full JSON output looking like this:

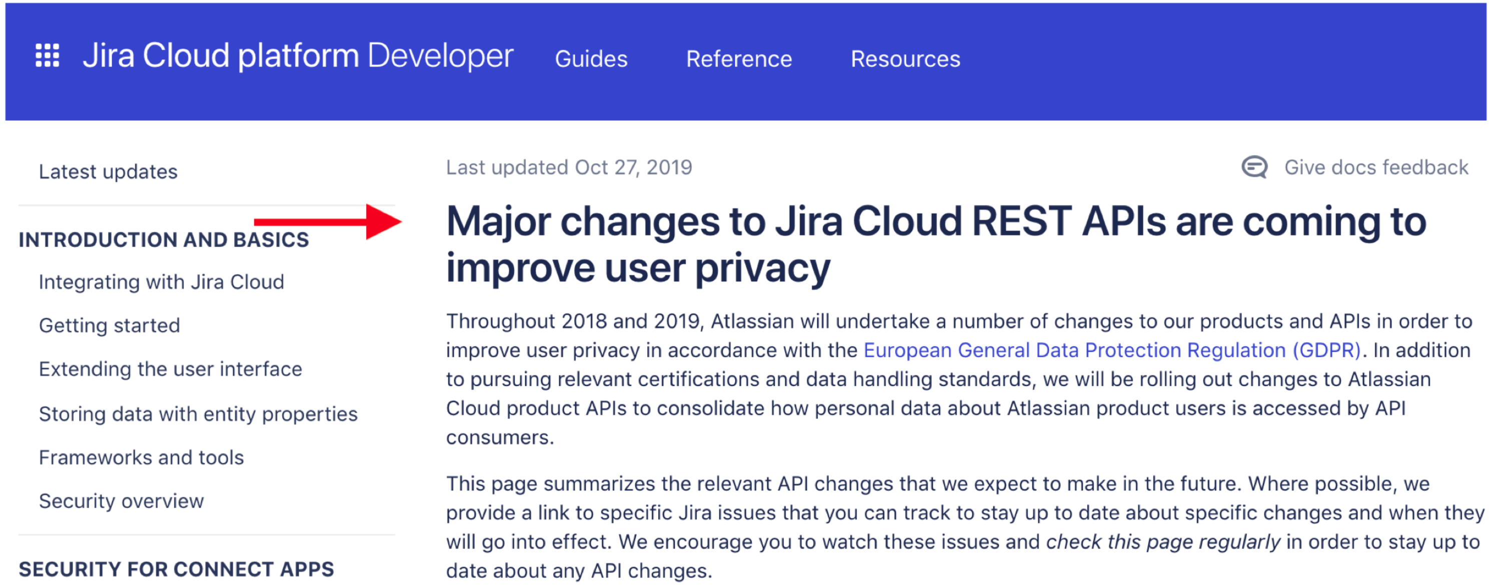
```
2020-01-07 17:13:54.175+0000 [id=14]   FINEST        java_util_logging_Logger$finest$0#call: {
            "timestamp": 1578416675038,
            "webhookEvent": "jira:issue_updated",
            "issue_event_type_name": "issue_generic",
            "user": {
               "self": "$URL",
               "accountId": "5d81f5c401e2cb0c301fb9ea",
               "avatarUrls": {...
               },
               "displayName": "$USER",
               "active": true,
               "timeZone": "America/New_York",
        .... (continuing from here)
```

Inside of that JSON response, nowhere was there a top level `name` object which is required to properly populate the webhook data. Therefore the error is thrown.

The JIRA Hook Issue

- Searching for Known Issues in the [Community Tracker]()
- Searching the for a matching Stacktrace in [CloudBees Knowledge Base]()
- Consulting with Third Party Vendor to resolve the issue

Searching led us to a security announcement:



Jira Cloud platform Developer
Guides
Reference
Resources
Latest updates
INTRODUCTION AND BASICS
Integrating with Jira Cloud
Getting started
Extending the user interface
Storing data with entity properties
Frameworks and tools
Security overview
SECURITY FOR CONNECT APPS
Give docs feedback

Last updated Oct 27, 2019

Major changes to Jira Cloud REST APIs are coming to improve user privacy

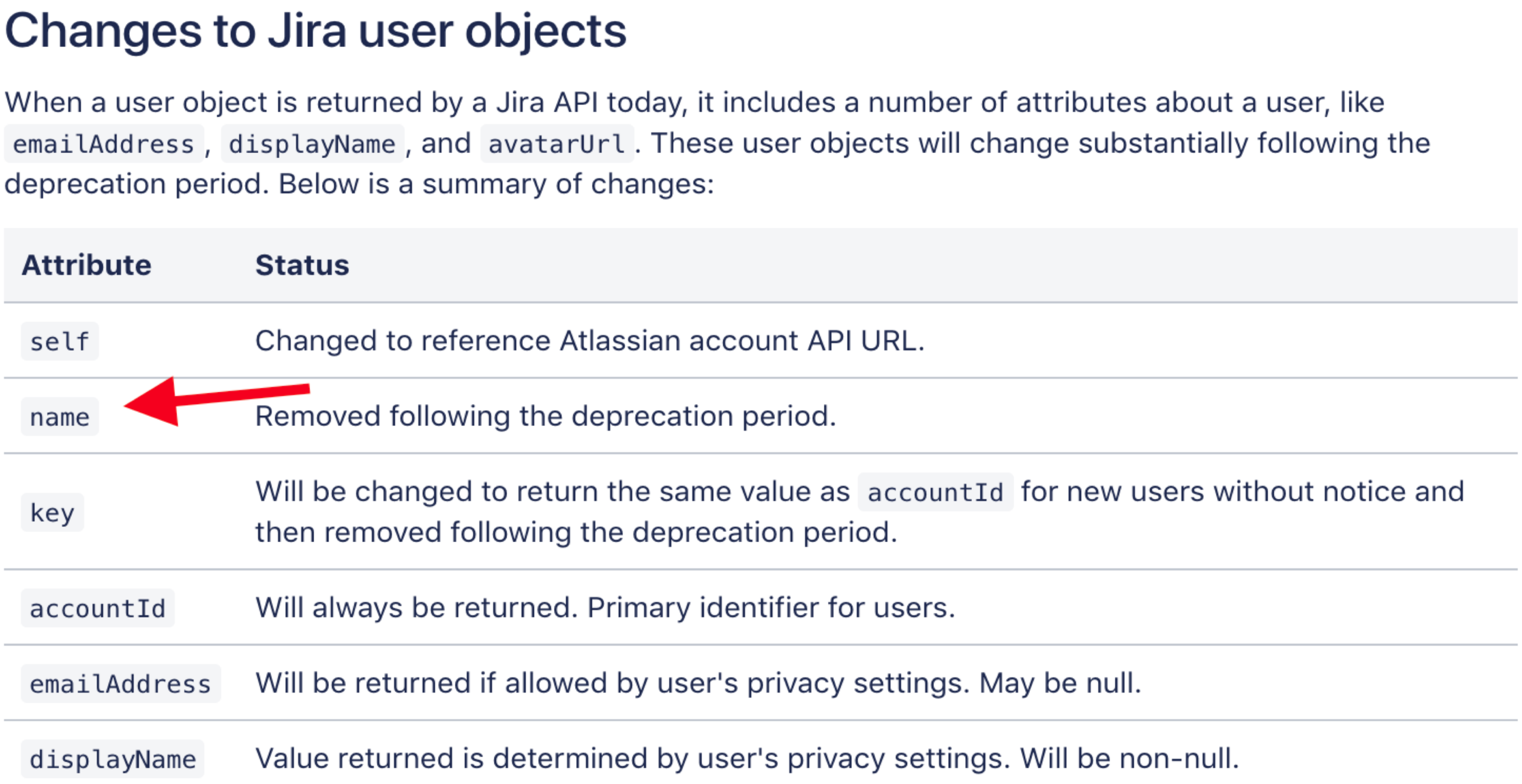
Throughout 2018 and 2019, Atlassian will undertake a number of changes to our products and APIs in order to improve user privacy in accordance with the European General Data Protection Regulation (GDPR). In addition to pursuing relevant certifications and data handling standards, we will be rolling out changes to Atlassian Cloud product APIs to consolidate how personal data about Atlassian product users is accessed by API consumers.

This page summarizes the relevant API changes that we expect to make in the future. Where possible, we provide a link to specific Jira issues that you can track to stay up to date about specific changes and when they will go into effect. We encourage you to watch these issues and *check this page regularly* in order to stay up to date about any API changes.

DEVOPSWORLD by CloudBees

The JIRA Hook Issue

- This matches what we are seeing!



Changes to Jira user objects

When a user object is returned by a Jira API today, it includes a number of attributes about a user, like `emailAddress`, `displayName`, and `avatarUrl`. These user objects will change substantially following the deprecation period. Below is a summary of changes:

Attribute	Status
`self`	Changed to reference Atlassian account API URL.
`name`	Removed following the deprecation period.
`key`	Will be changed to return the same value as `accountId` for new users without notice and then removed following the deprecation period.
`accountId`	Will always be returned. Primary identifier for users.
`emailAddress`	Will be returned if allowed by user's privacy settings. May be null.
`displayName`	Value returned is determined by user's privacy settings. Will be non-null.

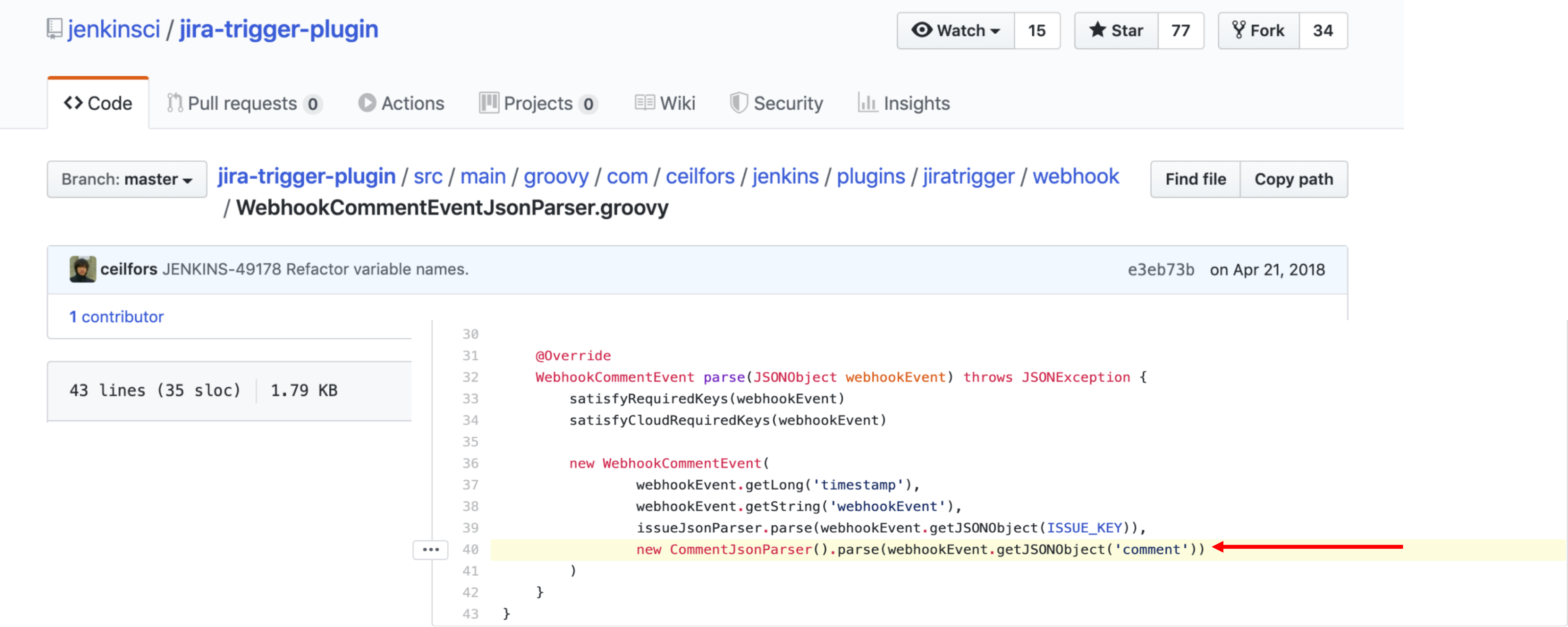
The JIRA Hook Issue

- Let me show you where I found the code!

```
2020-01-07 17:30:53.262+0000 [id=833]       WARNING o.e.j.s.h.ContextHandler$Context#log: Error while serving
https://jenkinscb2.bcbsma.com/jira-trigger-webhook-receiver/
org.codehaus.jettison.json.JSONException: JSONObject["name"] not found.
            at org.codehaus.jettison.json.JSONObject.get(JSONObject.java:360)
            at org.codehaus.jettison.json.JSONObject.getString(JSONObject.java:487)
            at com.atlassian.jira.rest.client.internal.json.JsonParseUtil.parseBasicUser(JsonParseUtil.java:192)
            at com.atlassian.jira.rest.client.internal.json.CommentJsonParser.parse(CommentJsonParser.java:37)
            at com.atlassian.jira.rest.client.internal.json.CommentJsonParser$parse.call(Unknown Source)
            at
com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser.parse(WebhookCommentEventJsonParser.groovy:40)
            at com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser$parse.call(Unknown Source)
            at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.processEvent(JiraWebhook.groovy:71)
            at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook$processEvent.callCurrent(Unknown Source)
            at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.doIndex(JiraWebhook.groovy:51)
            at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
            at org.kohsuke.stapler.Function$MethodFunction.invoke(Function.java:396)
Caused: java.lang.reflect.InvocationTargetException
```

https://github.com/jenkinsci/jira-trigger-plugin/blob/master/src/main/groovy/com/ceilfors/jenkins/plugins/jiratrigger/webhook/WebhookCommentEventJsonParser.groovy#L40

The JIRA Hook Issue



jira-trigger-plugin / src / main / groovy / com / ceilfors / jenkins / plugins / jiratrigger / webhook / WebhookCommentEventJsonParser.groovy

ceilfors JENKINS-49178 Refactor variable names. e3eb73b on Apr 21, 2018

43 lines (35 sloc) 1.79 KB

```

    @Override
    WebhookCommentEvent parse(JSONObject webhookEvent) throws JSONException {
        satisfyRequiredKeys(webhookEvent)
        satisfyCloudRequiredKeys(webhookEvent)

        new WebhookCommentEvent(
                webhookEvent.getLong('timestamp'),
                webhookEvent.getString('webhookEvent'),
                issueJsonParser.parse(webhookEvent.getJSONObject(ISSUE_KEY)),
                new CommentJsonParser().parse(webhookEvent.getJSONObject('comment'))
        )
    }
}
```

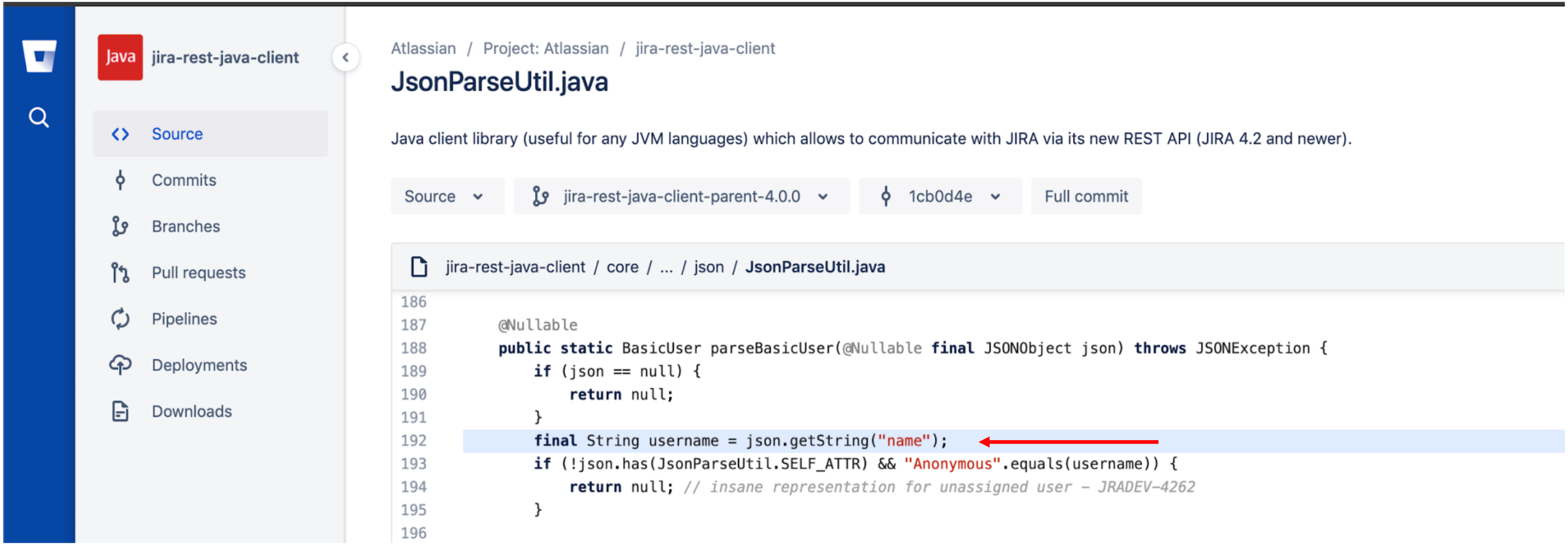
The JIRA Hook Issue

- Let me show you where I found the code!

```
2020-01-07 17:30:53.262+0000 [id=833]	WARNING o.e.j.s.h.ContextHandler$Context#log: Error while serving
https://jenkinscb2.bcbsma.com/jira-trigger-webhook-receiver/
org.codehaus.jettison.json.JSONException: JSONObject["name"] not found.
		at org.codehaus.jettison.json.JSONObject.get(JSONObject.java:360)
		at org.codehaus.jettison.json.JSONObject.getString(JSONObject.java:487)
		at com.atlassian.jira.rest.client.internal.json.JsonParseUtil.parseBasicUser(JsonParseUtil.java:192)
		at com.atlassian.jira.rest.client.internal.json.CommentJsonParser.parse(CommentJsonParser.java:37)
		at com.atlassian.jira.rest.client.internal.json.CommentJsonParser$parse.call(Unknown Source)
		at
com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser.parse(WebhookCommentEventJsonParser.groovy:40)
		at com.ceilfors.jenkins.plugins.jiratrigger.webhook.WebhookCommentEventJsonParser$parse.call(Unknown Source)
		at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.processEvent(JiraWebhook.groovy:71)
		at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook$processEvent.callCurrent(Unknown Source)
		at com.ceilfors.jenkins.plugins.jiratrigger.webhook.JiraWebhook.doIndex(JiraWebhook.groovy:51)
		at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
		at org.kohsuke.stapler.Function$MethodFunction.invoke(Function.java:396)
Caused: java.lang.reflect.InvocationTargetException
```

https://github.com/jenkinsci/jira-trigger-plugin/blob/master/src/main/groovy/com/ceilfors/jenkins/plugins/jiratrigger/webhook/WebhookCommentEventJsonParser.groovy#L40

The JIRA Hook Issue


Java
jira-rest-java-client
Source
Commits
Branches
Pull requests
Pipelines
Deployments
Downloads
Atlassian / Project: Atlassian / jira-rest-java-client

JsonParseUtil.java

Java client library (useful for any JVM languages) which allows to communicate with JIRA via its new REST API (JIRA 4.2 and newer).

Source
jira-rest-java-client-parent-4.0.0
1cb0d4e
Full commit
jira-rest-java-client / core / ... / json / JsonParseUtil.java

```

    @Nullable
    public static BasicUser parseBasicUser(@Nullable final JSONObject json) throws JSONException {
        if (json == null) {
            return null;
        }
        final String username = json.getString("name");
        if (!json.has(JsonParseUtil.SELF_ATTR) && "Anonymous".equals(username)) {
            return null; // insane representation for unassigned user - JRADEV-4262
        }

```

Demystifying Debugging Through Logging

Real World Example #2 “Agents Not Connecting”

Agent’s Not Connecting Issue

- Common Support issue
- Where do you start diagnosis?
    - Networking
    - Ports
    - Firewalls
    - Proxy Server
    - SSL Certificates
    - Misconfiguration
- Start with the logs

Agent’s Not Connecting Issue 1

Data provided to CloudBees Support:

Upon running the following command:

```
C:\jenkins>java -jar slave.jar -jnlpUrl https://$JENKINS_URL/
computer/$AGENT_NAME/slave-agent.jnlp -secret b4b49e3d93a4d
f2d333908c8fb2e26d18ac31bb890dae2d366fb60d14129d9df
```

We see the following error:

```
INFO: Connecting to $JENKINS_URL:59988 (retrying:2)
java.net.ConnectException: Connection timed out: connect
```

Information provided:

Jenkins Support Bundle provided by the Support Core Plugin

Agent’s Not Connecting Issue 1

Steps to Debug:

```
INFO: Connecting to $JENKINS_URL:59988 (retrying:2)
java.net.ConnectException: Connection timed out: connect
```

- Error is: “java.net.ConnectException: Connection timed out: connect”
  - Is there any information on the master side.
- Where is it trying to connect to?
  - Can we `curl` or `netcat` the $JENKINS_URL?
  - Can we `curl` or `netcat` the $JENKINS_URL at port 59988?
- Is the port correct?

Agent’s Not Connecting Issue 1

Resolution:

Based on the error message from netcat/curl we know that the agent can reach the Jenkins URL but not at the port which was given(59988). This means either that port needs to be opened or the wrong port is being used.

We then took a look at the Jenkins bundle we could see from the included Jenkins config file “config.xml” the item “<AgentPort>$PORT_NUMBER</AgentPort>” was missing. This means it was set to a Random Port which is the default Jenkins behavior. The expected behavior from the customer was that port to be set to 50000 as that is what was open on the firewall.

Once that port was set back to 50000, the agent connected with no issues and the message “INFO: Connecting to $JENKINS_URL:50000” came up in the agent logs

Agent’s Not Connecting Issue 2

Data provided to CloudBees Support:

Upon running the following command:

```
C:\$DIRECTORY>java -jar agent.jar -jnlpUrl https://$JENKINS_URL/computer/$AGENT_NAM/slave-agent.jnlp -workDir
"C:\$DIRECTORY"
```

We see the following error:

```
Jan 02, 2020 10:50:55 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\$DIRECTORY\remoting as a remoting work directory
Both error and output logs will be printed to C:\$DIRECTORY\remoting
Exception in thread "main" java.io.IOException: Failed to validate a server certificate. If you are using a self-signed cer
tificate, you can use the -noCertificateCheck option to bypass this check.
at hudson.remoting.Launcher.parseJnlpArguments(Launcher.java:548)
at hudson.remoting.Launcher.run(Launcher.java:322)
at hudson.remoting.Launcher.main(Launcher.java:283)
Caused by: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.s
ecurity.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
at sun.security.ssl.Alerts.getSSLException(Unknown Source)
...
Caused by: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPath
BuilderException: unable to find valid certification path to requested target
...
```

As well as Jenkins Support Bundle provided by the Support Core Plugin including:

- Plugin Versions
- Jenkins Version
- Master Logs
- Jenkins Configuration files

Agent’s Not Connecting Issue 2

Steps to Debug:

- “unable to find valid certification path to requested target”
  - Is there any information on the master side?
- Where is it trying to connect to?
- How to resolve the certificate error?
  - Start the agent with the “-noCertificateCheck” option
  - Make sure the certificate is in the correct location for Java to find it

```
Jan 02, 2020 10:50:55 AM
org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\$DIRECTORY\remoting as a remoting work directory
Both error and output logs will be printed to C:\$DIRECTORY\remoting
Exception in thread "main" java.io.IOException: Failed to validate a server
certificate. If you are using a self-signed cer
tificate, you can use the -noCertificateCheck option to bypass this check.
at hudson.remoting.Launcher.parseJnlpArguments(Launcher.java:548)
at hudson.remoting.Launcher.run(Launcher.java:322)
at hudson.remoting.Launcher.main(Launcher.java:283)
Caused by: javax.net.ssl.SSLHandshakeException:
sun.security.validator.ValidatorException: PKIX path building failed: sun.s
ecurity.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
at sun.security.ssl.Alerts.getSSLException(Unknown Source)
...
Caused by: sun.security.validator.ValidatorException: PKIX path building
failed: sun.security.provider.certpath.SunCertPath
BuilderException: unable to find valid certification path to requested
target
...
```

Agent’s Not Connecting Issue 2

Resolution:

Based on the given error message we know the reason the agent would not connect to be caused by a certificate issue. To fix this we can either ignore certificate errors or put the certificate in place.

For this particular case because the agent was needed quickly so the error was resolved with the -noCertificateCheck option. Because builds were able to function immediately, this then gave us time to add the self-signed certificate from the Jenkins master to the agent directly.

To add the certificate, we downloaded the cert to the windows machine and then ran the following command “keytool -import -trustcacerts -keystore /$JAVA_HOME/lib/security/cacerts -storepass changeit -noprompt -alias mycert -file /tmp/examplecert.crt” which imported the downloaded Jenkins certificate into the default cacerts file. Once that was added we were able to remove the -noCertificateCheck option and it connected successfully as the certificate was now “authorized”

Demystifying Debugging Through Logging

Additional Reading

DEVOPS WORLD
by CloudBees

Additional Reading and Viewing

- Matthew Gilliard’s blog: “How to read and understand a Java Stacktrace
- How to Get the Most from Jenkins
- Download Jenkins Health Advisor

Thanks!

Q&A

DEVOPS WORLD by CloudBees