Documentation

IBM Workload Scheduler integration with Splunk

Written by : **Miguel Sanders** Uniforce	**Date :** **August 18 2017**

Table of Contents

CHANGE HISTORY

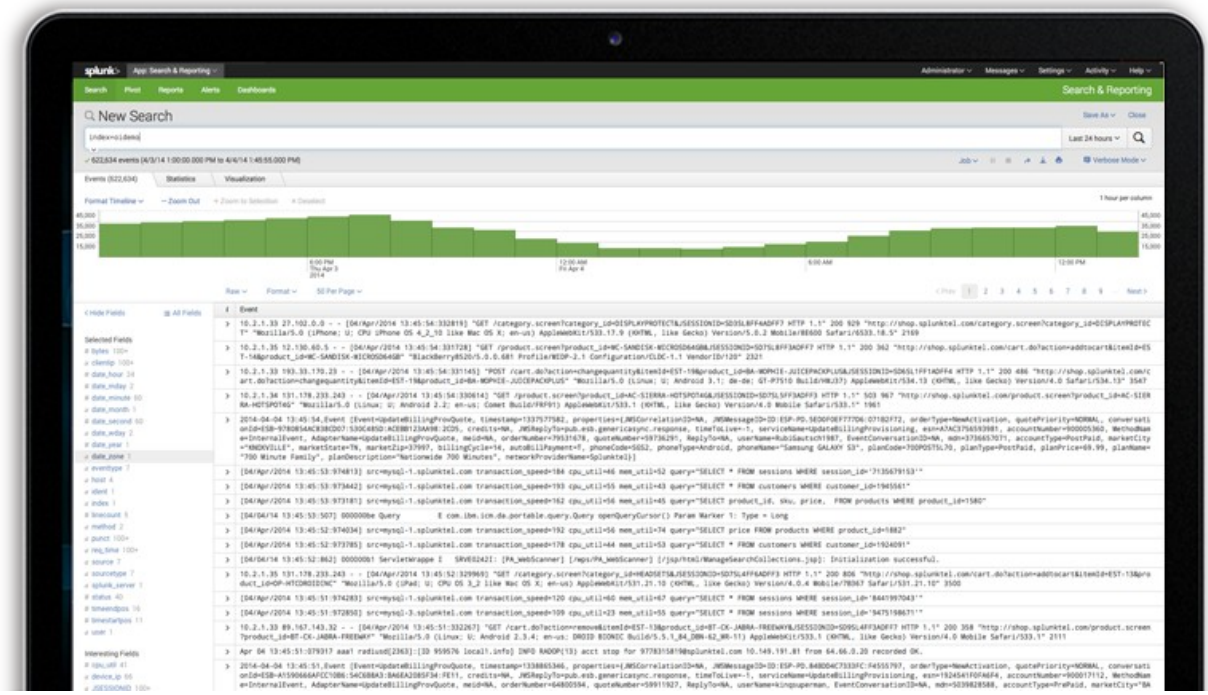
Version	Date of change	Change detail
1.0	August 18 2017	Initial version by M. Sanders

1. INTRODUCTION

Splunk Enterprise and Splunk Cloud monitor and analyze machine data from any source to deliver Operational Intelligence to optimize your IT, security and business performance. With intuitive analysis features, machine learning, packaged applications and open APIs, Splunk Enterprise and Splunk Cloud are flexible platforms that scale from focused use cases to an enterprise-wide analytics backbone. Features include the following:

- Collects and indexes log and machine data from any source
- Powerful search, analysis and visualization capabilities empower users of all types
- Apps provide solutions for security, IT ops, business analysis and more
- Enables visibility across on premise, cloud and hybrid environments
- Delivers the scale, security and availability to suit any organization
- Available as software or as a cloud service

Using the IBM Workload Scheduler plug-in for Splunk, you will be able to schedule reports in Splunk Enterprise and Splunk Cloud.



2. INSTALLING AND CONFIGURING THE PLUG-IN FOR SPLUNK

The following prerequisites must be met in order to use the IBM Workload Scheduler plug-in for Splunk.

- IBM Workload Scheduler 9.1 or later
- IBM Dynamic Workload Console 9.1 or later
- Splunk 6.6 or later

To install and configure the IBM Workload Scheduler plug-in for Splunk, perform the following steps:

- Copy *com.ibm.scheduling.agent.splunk_<version>.jar* to the *<TWA_HOME>/TWS/applicationJobPlugIn* folder on either the Master Domain Manager or a Dynamic Domain Manager
- Copy *com.ibm.scheduling.agent.splunk _<version>.jar* to the *<TWA_HOME>/TWS/JavaExt/eclipse/plugins* folder on the Dynamic Agent that will run the Splunk jobs
- Modify *config.ini* located in *<TWA_HOME>/TWS/JavaExt/eclipse/configuration* on the Dynamic Agent that will run the Splunk jobs. At the end of the line that starts with "osgi.bundles=", add the following: *",com.ibm.scheduling.agent.splunk@4:start"*
- Restart the WebSphere Application Server of either the Master Domain Manager or the Dynamic Domain Manager.
- Restart the Dynamic Agent
- Restart the Dynamic Workload Console

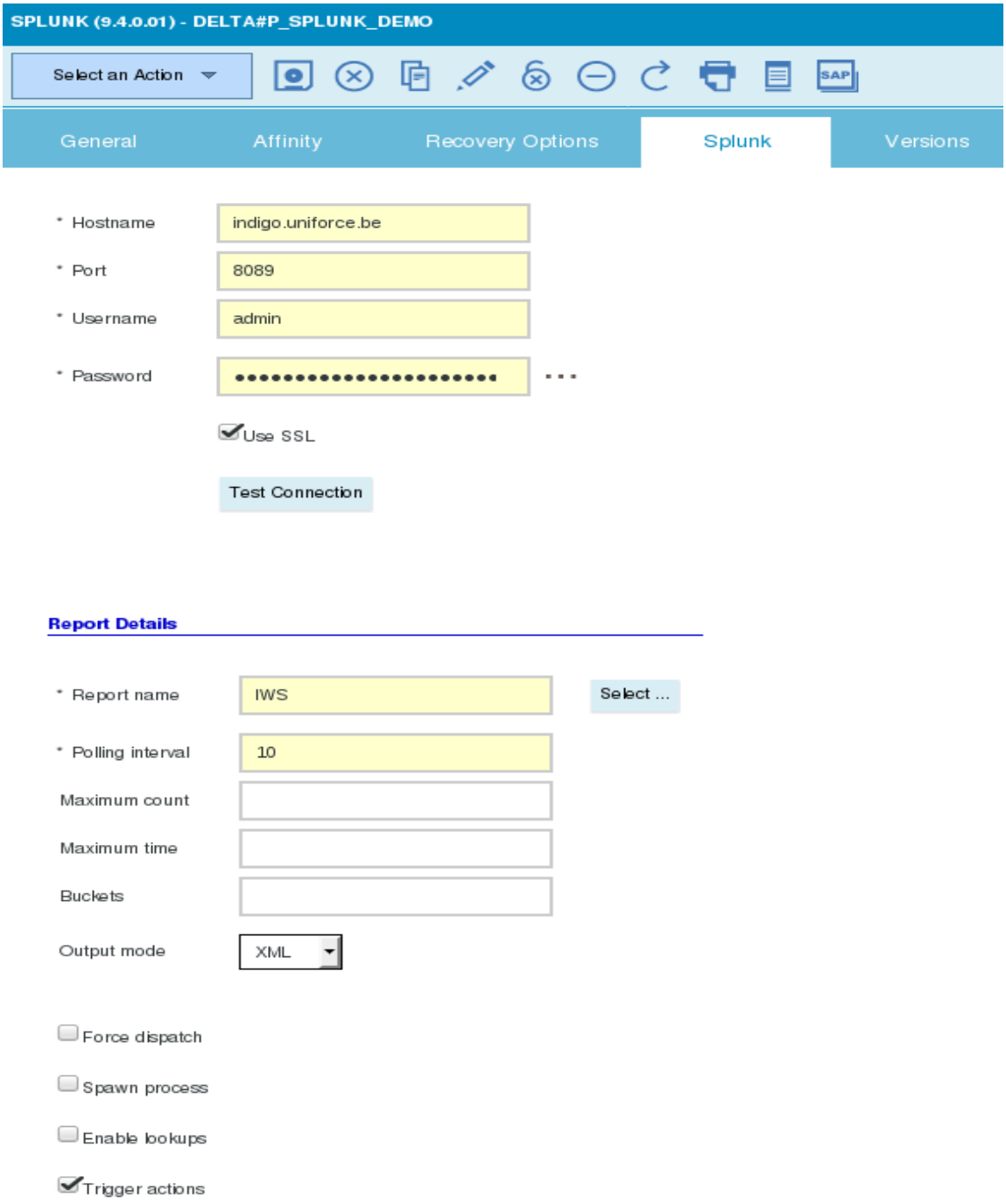
3. DEFINING A SPLUNK JOB

From the Dynamic Workload Console, you can define a Splunk job as follows:

- In the Dynamic Workload Console navigation tree, expand *Administration* and select *Manage Workload Definitions*.
- Specify the name of the engine. Subsequently, the Workload Designer is displayed.
- In the Working List panel, select *New -> Job Definition -> Business Analytics -> Splunk*.
- On the *Splunk* panel, fill in the job details
    - **Hostname** : Hostname of the Splunk Enterprise Server or Splunk Cloud. For Self-service Splunk Cloud deployments, use the following URL

      input-<deployment-name>.cloud.splunk.com
    - **Username** : The username to log on to Splunk Enterprise or Splunk Cloud. For Splunk Cloud, submit a support case with Splunk to get non-SAML user credentials.
    - **Password** : The password of the user that will log on to Splunk Enterprise or Splunk Cloud.
    - **Use SSL** : Select whether SSL/TLS is used for the connection to Splunk Enterprise or Splunk Cloud. For Splunk Cloud, this is mandatory.

      To validate the connection, click *Test Connection*.
    - **Report name** : The name of the report that you want to schedule. Use the "*Select...*" button to get a picklist which shows all defined reports.
    - **Polling interval** : The monitoring frequency determines how often the job is monitored. The default value is 10 seconds..
    - **Maximum count** : Limit the maximum number of results before finalizing the search.
    - **Maximum time** : Limit the maximum amount of time before finalizing the search.
    - **Buckets** : Sets the maximum number of timeline buckets.
    - **Output mode** : Select the output mode for the search result (XML, JSON or CSV).
    - **Force dispatch** : Indicates whether to start a new search, even if another instance of this search is already running.
    - **Spawn process** : Indicates whether the search should run in a separate spawned process. Searches against indexes must run in a separate process.
    - **Enable lookups** : Indicates whether to enable lookups for this search.
    - **Trigger actions** : Indicates whether to trigger the alert actions that are associated with the report.

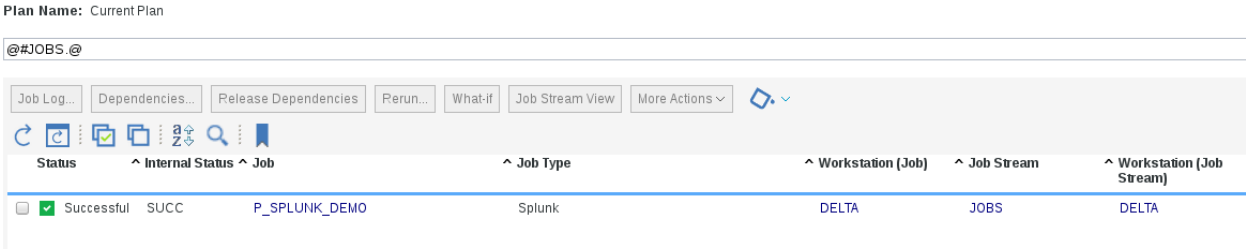

SPLUNK (9.4.0.01) - DELTA#P_SPLUNK_DEMO
Select an Action
General
Affinity
Recovery Options
Splunk
Versions
* Hostname
indigo.uniforce.be
* Port
8089
* Username
admin
* Password
Use SSL
Test Connection
Report Details
* Report name
IWS
Select ...
* Polling interval
10
Maximum count
Maximum time
Buckets
Output mode
XML
Force dispatch
Spawn process
Enable lookups
Trigger actions

Alternatively, the *composer* command line can be used to define the job. Example:

```
DELTA#P_SPLUNK_DEMO
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlsplunk="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlsplunk" name="SPLUNK">
  <jsdl:application name="splunk">
    <jsdlsplunk:splunk>
        <jsdlsplunk:SplunkParameters>
               <jsdlsplunk:SplunkParms>
                      <jsdlsplunk:serverInformation>
                             <jsdlsplunk:hostname>indigo.uniforce.be</jsdlsplunk:hostname>
                             <jsdlsplunk:port>8089</jsdlsplunk:port>
                             <jsdlsplunk:username>admin</jsdlsplunk:username>
                             <jsdlsplunk:password>{aes}UG/QqqXCFjk=</jsdlsplunk:password>
                             <jsdlsplunk:ssl/>
                      </jsdlsplunk:serverInformation>
                      <jsdlsplunk:reportDetails>
                             <jsdlsplunk:reportName>IWS</jsdlsplunk:reportName>
                             <jsdlsplunk:pollingInterval>10</jsdlsplunk:pollingInterval>
                             <jsdlsplunk:maximumCount/>
                             <jsdlsplunk:maximumTime/>
                             <jsdlsplunk:buckets/>
                             <jsdlsplunk:outputMode>outputModeXml</jsdlsplunk:outputMode>
                             <jsdlsplunk:triggerActions/>
                      </jsdlsplunk:reportDetails>
               </jsdlsplunk:SplunkParms>
        </jsdlsplunk:SplunkParameters>
     </jsdlsplunk:splunk>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

4. MONITORING A SPLUNK JOB

Like regular jobs, you can monitor Splunk jobs by using either the Dynamic Workload Console or the *conman* command line.



Extended job properties can be displayed using either the Dynamic Workload Console or the conman command line using the «*;props*» option

'P_SPLUNK_DEMO' Properties

Action	
▼ Deadline	
Length of Time	
Action	
Repeat Range	
▼ Extra Information	
Disk Usage	1675264
Dispatch State	DONE
Done Progress	1.0
Drop Count	0
Event Available Count	33075
Event Count	33075
Event Field Count	6
Label	IWS
Priority	5
Result Count	33075
Run Duration	1.003
Scan Count	33075
Search Identifier	admin__admin__search__IWS_at_1503052151_910
Time to live	86400
▼ Splunk	
buckets	
forceDispatch	false
hostname	indigo.uniforce.be
lookups	false
maximumCount	
maximumTime	
outputMode	outputModeXml
password	{aes}UG/QqqXCFjmnmhthzZfXv33vAv3vXcfY9PubhegSxZk=
pollingInterval	10
port	8089
reportName	IWS
spawnProcess	false
ssl	true
triggerActions	true
username	admin