Rsyslog Build Documentation

Summary

Rsyslog is used in the CommIT environment to enable centralized logging. This allows the instances to use ephemeral storage as well as providing an copy of all logs on a different server in the event of a security breach. This document details how rsyslog is deployed in this environment.

Installation

rsyslog is installed by default on Amazon Linux and presumably most other Linux variations in EC2. The current deployed version is 5.8.10. This is important as the syntax of the configuration file, /etc/rsyslog.conf, has changed dramatically over the later versions. The current version is 8.X and the default documentation on http://www.rsyslog.com is for that version. Documentation for the 5.X version can be found here: http://www.rsyslog.com/doc/v5-stable/index.html.

Since we are using version 5.X, we use what are now called Legacy configuration directives.

An rsyslog server instance has been deployed and given an Elastic IP of **54.214.22.10**.

Server Configuration

The rsyslog server is configured to accept syslog messages from remote systems. This functionality is configured in /etc/rsyslog.conf. The important lines in that file are as follows.

```
# Provides TCP syslog reception
$ModLoad imtcp

...

$RuleSet REMOTE_Ruleset

$template REMOTE_Messages,"/var/log/remote-syslogs/%FROMHOST%/messages"
$template REMOTE_Secure,"/var/log/remote-syslogs/%FROMHOST%/secure"
$template REMOTE_Maillog,"/var/log/remote-syslogs/%FROMHOST%/maillog"
$template REMOTE_Cron,"/var/log/remote-syslogs/%FROMHOST%/cron"
$template REMOTE_Catalina,"/var/log/remote-syslogs/%FROMHOST%/catalina.out"
$template REMOTE_389ds_Access,"/var/log/remote-syslogs/%FROMHOST%/389ds_access"
$template REMOTE_389ds_Errors,"/var/log/remote-syslogs/%FROMHOST%/389ds_errors"

if \
        $syslogseverity <= '6' \
        and $syslogfacility-text != 'mail' \
        and $syslogfacility-text != 'authpriv' \
        and $syslogfacility-text != 'cron' \
        and $syslogfacility-text != 'local1' \
then    ?REMOTE_Messages
if \
        $syslogfacility-text == 'authpriv' \
then    ?REMOTE_Secure
if \
        $syslogfacility-text == 'mail' \
then    -?REMOTE_Maillog
if \
        $syslogfacility-text == 'cron' \
then    ?REMOTE_Cron
if \
        $syslogfacility-text == 'local1' \
then    ?REMOTE_Catalina
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-access' \
then    ?REMOTE_389ds_Access
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-errors' \
then    ?REMOTE_389ds_Errors
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-access' \
then    ?REMOTE_389ds_Access
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-errors' \
then    ?REMOTE_389ds_Errors

# These two lines must be defined *after* the ruleset it is defined (above)
$InputTCPServerBindRuleset REMOTE_Ruleset
$InputTCPServerRun 10514

$RuleSet RSYSLOG_DefaultRuleset
```

Note, the rsyslog server was configured to run in TCP mode and listen on port 10514 for connections from remote servers.

After making these changes, the rsyslogd process on the server must be restarted using /etc/init.d/rsyslog restart

Syslog EBS Volume

An EBS volume has been attached to and mounted on mount point /mnt/remote-syslogs on the rsyslog server. A symlink has been created from /var/log /remote-syslogs -> /mnt/remote-syslogs. rsyslog.conf configuration file entries point to the /var/log/remote-syslogs directory tree. To separate the log files received from each instance, the %FROMHOST% property used in the path to the instance log directory, e.g.

/var/log/remote-syslogs/ip-10-252-20-14.us-west-2.compute.internal/messages

Client Configuration

rsyslogd Configuration

Clients must be configured to fork syslog messages to the rsyslog server. This functionality is configured in /etc/rsyslog.conf on the client instances. The important lines in that file are as follows.

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

...

$WorkDirectory /var/lib/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList   # run asynchronously
$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
*.* @@ec2-54-214-22-10.us-west-2.compute.amazonaws.com:10514# ### end of the forwarding rule ###
# ### end of the forwarding rule ###
```

The FQDN of the rsyslog server's Elastic IP, namely **ec2-54-214-22-10.us-west-2.compute.amazonaws.com**, is configured above. Note, that the TCP protocol, as denoted by two '@@' versus one '@' for UDP, is used and the communication takes place over port 10514.

After editing this file, as root, **/etc/init.d/rsyslog restart**

It may seem odd that we've also configured the ***client*** to listen for UDP communication on port 514. This is ***ONLY*** necessary on clients that wish to use the log4j syslog appender. This appender, used to support remote logging of Tomcat's catalina.out, connects via UDP on 514 on the localhost (client) by default and there doesn't appear to be a way to override that. Tomcat log messages are routed through the local rsyslogd process and on to the rsyslog server by the configuration lines shown above.

Tomcat Log4j Configuration

The instructions below assume that the Tomcat 6.0.39 is already installed in /opt/apache-tomcat-6.0.39. Two "extra" jar files files plus the log4j jar with version > 1.2 are required. These can be downloaded from an apache mirror as shown below.

```
<as root, do the following>


cd
mkdir tmp
cd tmp
wget http://mirrors.gigenet.com/apache/tomcat/tomcat-6/v6.0.39/bin/extras/tomcat-juli-adapters.jar
wget http://mirrors.gigenet.com/apache/tomcat/tomcat-6/v6.0.39/bin/extras/tomcat-juli.jar
wget http://mirrors.gigenet.com/apache/logging/log4j/1.2.17/log4j-1.2.17.tar.gz
tar xzf log4j-1.2.17.tar.gz

cd apache-tomcat-6.0.39/
cp ~/tmp/apache-log4j-1.2.17/log4j-1.2.17.jar libcd /opt/apache-tomcat-6.0.39/
cp ~/tmp/apache-log4j-1.2.17/log4j-1.2.17.jar lib/
cp ~/tmp/tomcat-juli-adapters.jar lib/

mv bin/tomcat-juli.jar bin/tomcat-juli.jar.orig
cp ~/tmp/tomcat-juli.jar bin/
mv conf/logging.properties conf/logging.properties.orig
cp ~/tmp/tomcat-juli-adapters.jar lib
mv bin/tomcat-juli.jar bin/tomcat-juli.jar.origcp ~/tmp/tomcat-juli.jar binmv conf/logging.properties conf
/logging.properties.ori
```

Finally, a /opt/apache-tomcat-6.0.39/lib/log4j.properties file should be created and populated with the following:

```
log4j.rootLogger=INFO, CATALINA, SYSLOG

# Define all the appenders
log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender
log4j.appender.SYSLOG.syslogHost=localhost
log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout
log4j.appender.SYSLOG.layout.ConversionPattern=%-4r [%t] %-5p %c %x - %m%n
log4j.appender.SYSLOG.Header=true
log4j.appender.SYSLOG.Facility=LOCAL1

log4j.appender.CATALINA=org.apache.log4j.DailyRollingFileAppender
log4j.appender.CATALINA.File=${catalina.base}/logs/catalina.log
log4j.appender.CATALINA.Append=true
log4j.appender.CATALINA.Encoding=UTF-8
# Roll-over the log once per day
log4j.appender.CATALINA.DatePattern='.'yyyy-MM-dd'.log'
log4j.appender.CATALINA.layout = org.apache.log4j.PatternLayout
log4j.appender.CATALINA.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.LOCALHOST=org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOCALHOST.File=${catalina.base}/logs/localhost.log
log4j.appender.LOCALHOST.Append=true
log4j.appender.LOCALHOST.Encoding=UTF-8
log4j.appender.LOCALHOST.DatePattern='.'yyyy-MM-dd'.log'
log4j.appender.LOCALHOST.layout = org.apache.log4j.PatternLayout
log4j.appender.LOCALHOST.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.MANAGER=org.apache.log4j.DailyRollingFileAppender
log4j.appender.MANAGER.File=${catalina.base}/logs/manager.log
log4j.appender.MANAGER.Append=true
log4j.appender.MANAGER.Encoding=UTF-8
log4j.appender.MANAGER.DatePattern='.'yyyy-MM-dd'.log'
log4j.appender.MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.MANAGER.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.HOST-MANAGER=org.apache.log4j.DailyRollingFileAppender
log4j.appender.HOST-MANAGER.File=${catalina.base}/logs/host-manager.log
log4j.appender.HOST-MANAGER.Append=true
log4j.appender.HOST-MANAGER.Encoding=UTF-8
log4j.appender.HOST-MANAGER.DatePattern='.'yyyy-MM-dd'.log'
log4j.appender.HOST-MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.HOST-MANAGER.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Encoding=UTF-8
log4j.appender.CONSOLE.layout = org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

# Configure which loggers log to which appenders
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost]=INFO, LOCALHOST
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager]=INFO, MANAGER
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager]=INFO, HOST-MANAGER
```

Note the '**log4j.appender.SYSLOG.Facility=LOCAL1**' bolded text in the configuration above. This configures the log4j syslog appender to use the 'local1' syslog facility.

When these messages are sent to the remote rsyslog server, they are handled by the following section of '/etc/rsyslog.conf on the rsyslog server:

```
if \
      $syslogfacility-text == 'local1' \
then  ?REMOTE_Catalina
```

and thus sent to REMOTE_Catalina template, defined as /var/log/remote-syslogs/%FROMHOST%/catalina.out, on the rysslog server.

Sending Designated Log Files to the Rsyslog Server

Rsyslog supports "watching" designated files and sending copies of the updates to the rsyslog server shortly (default is 10 seconds) after they occur. The following example was used to configured the 389-ds servers to forward access and error log messages to the rsyslog server. This template, however, can be used to support log file that is desired to be replicated on the rsyslog server.

```
# Start - Added for 389ds rsyslog support - CWF
$ModLoad imfile

$InputFileName /var/log/dirsrv/slapd-i2commitdev/access
$InputFileTag 389ds-access
$InputFileStateFile state-389ds-access
$InputFileSeverity info
$InputFileFacility local4
$InputRunFileMonitor

$InputFileName /var/log/dirsrv/slapd-i2commitdev/errors
$InputFileTag 389ds-errors
$InputFileStateFile state-389ds-errors
$InputFileSeverity info
$InputFileFacility local4
$InputRunFileMonitor
# End - Added for 389ds rsyslog support - CWF

...

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none;local4.none    /var/log/messages
```

The rsyslog.conf configuration above on the client server, instructs rsyslog to load its imfile module, watch /var/log/dirsrv/slapd-i2commitdev/access and /var/log/dirsrv/slapd-i2commitdev/errors and create syslog messages at facility local4 and severity info. Note the bold **local4.none** addition to the default /var/log/messages line so the log messages don't get duplicated locally in that log file. Assuming the client's rsyslog.conf file contains the necessary configuration lines, shown above under rsyslogd configuration section, to forward syslog messages to the remote syslog server, the logging messages will be sent to the rsyslog server.

When these messages are received by the remote rsyslog server, they are handled by the following section of '/etc/rsyslog.conf on the rsyslog server:

```
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-access' \
then    ?REMOTE_389ds_Access
if \
        $syslogfacility-text == 'local4' \
        and $syslogtag contains '389ds-errors' \
then    ?REMOTE_389ds_Errors
```

and thus sent to REMOTE_389ds_Access and REMOTE_389ds_Errors templates, defined as /var/log/remote-syslogs/%FROMHOST%/389ds_access and /var/log/remote-syslogs/%FROMHOST%/389ds_errors respectively, on the rysslog server.

Common Commands

Starting and Stopping Rsyslog

[root@ip-10-252-20-14 ~]# /etc/init.d/rsyslog start

[root@ip-10-252-20-14 ~]# /etc/init.d/rsyslog stop

Enabling Debugging

export RSYSLOG_DEBUG=Debug

rsyslogd -d -n

rsyslogd -d -n