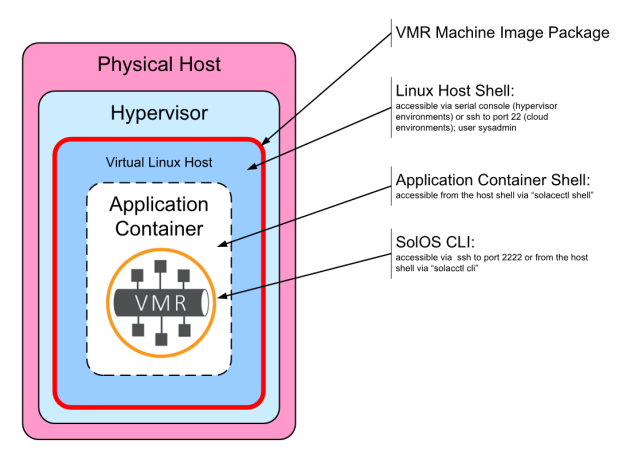
**Architecture – GTDC (SOLACE POC)**



**Build VMRs Cluster running as docker containers on VM’s**

VM Infrastructure composed of following Virtual Machines.

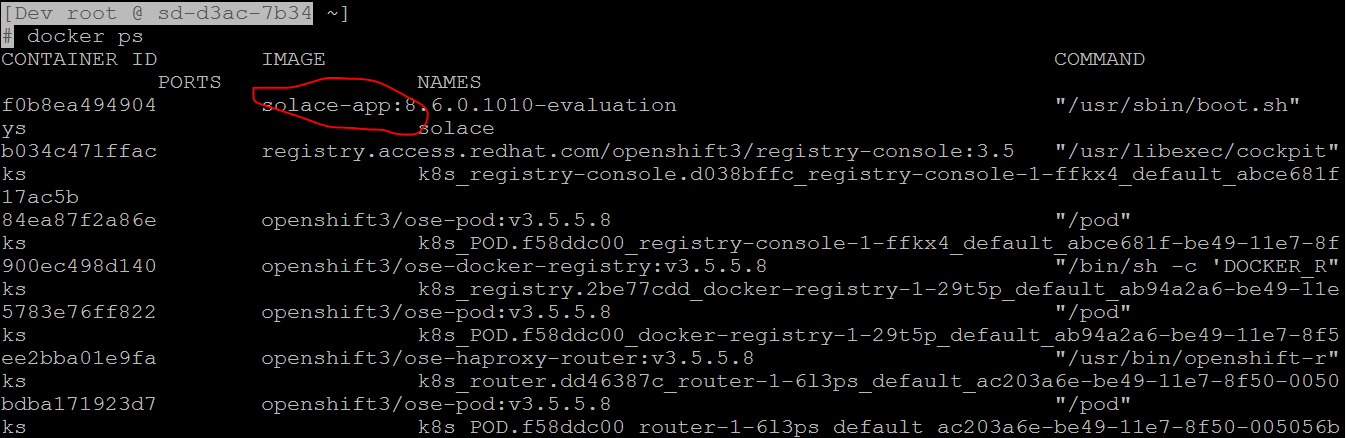
Sd-d3ac-7b34 (Primary VMR running on this)

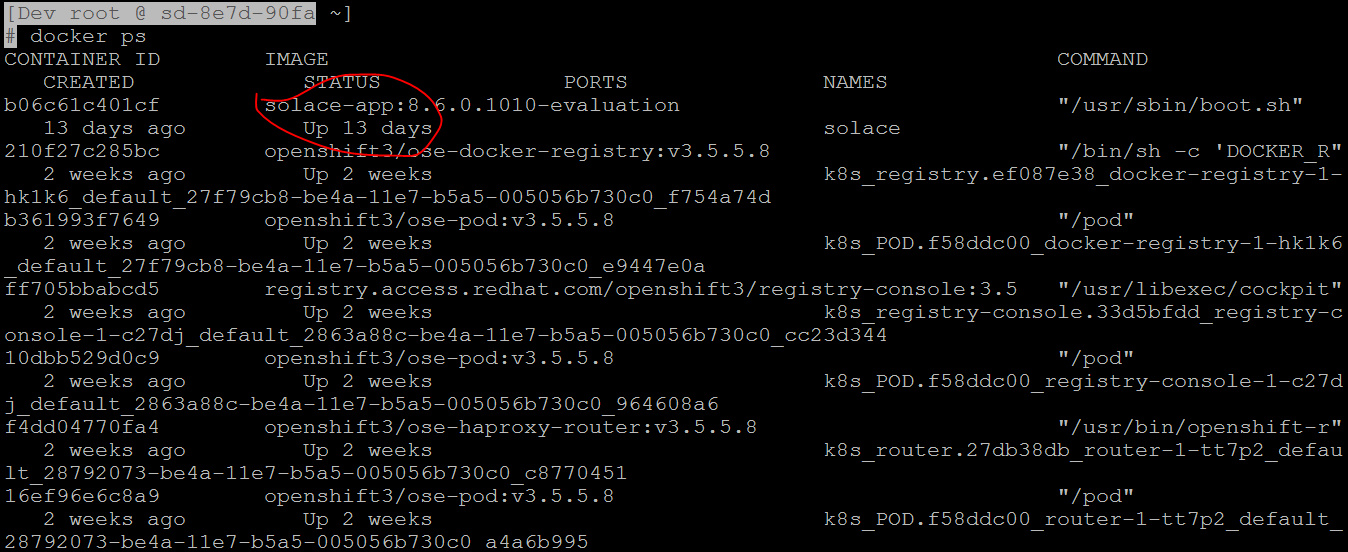
Sd-8e7d-90fa (Backup VMR running on this)

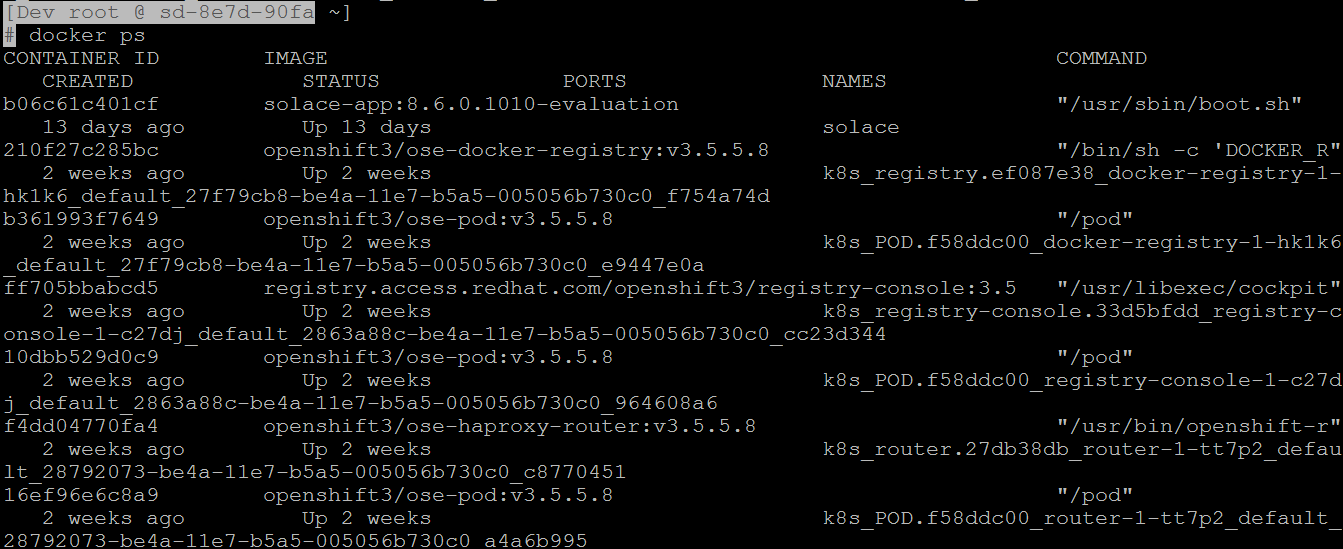
Sd-fd2a-23cb (Monitor VMR)

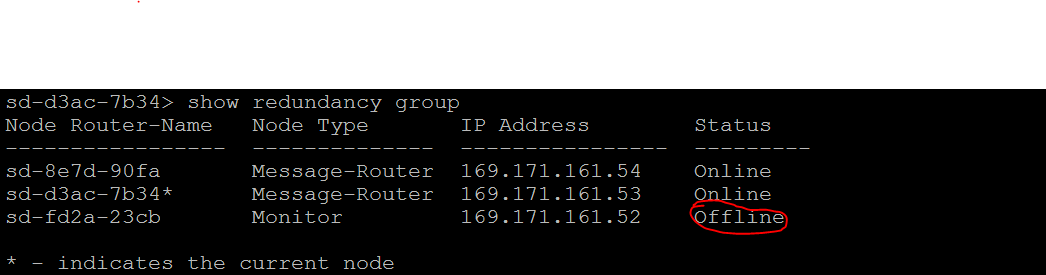
11/20

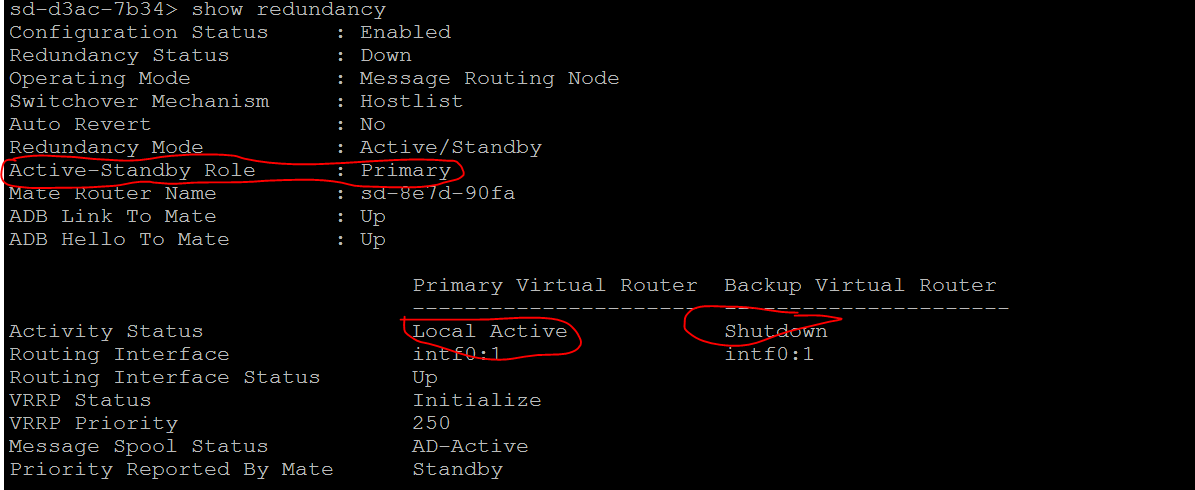
Each Solace VMRs is running as docker container on the each VM’s

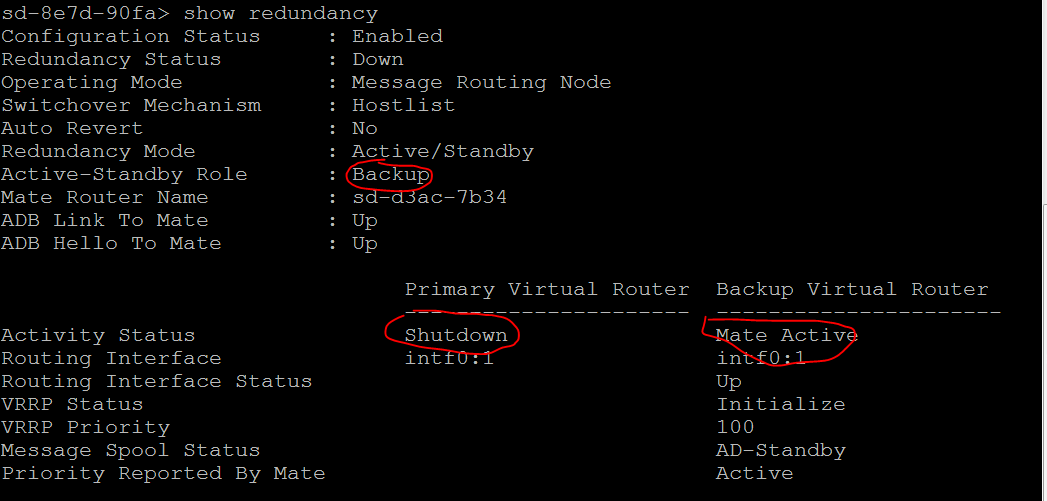




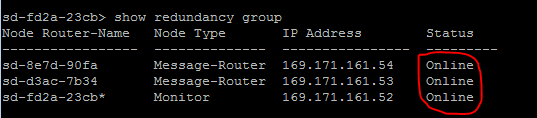


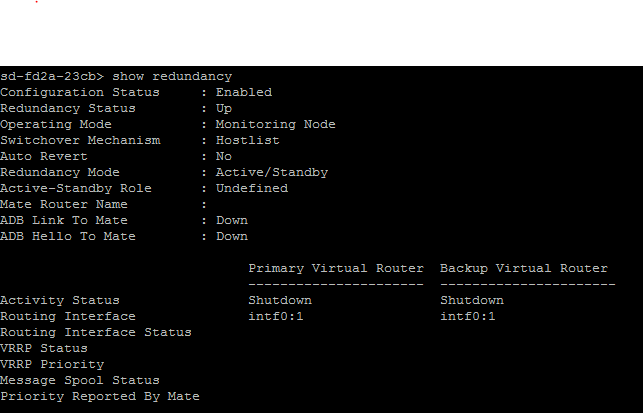






11/21

Configured the monitor node to make cluster healthy. 



**Integrating Solace with Splunk**

Description – Test the connectivity of Solace with Splunk

**Steps Taken –**

**11/29 –** Email communication initiated with Cloud Analytics team (Kitzman, Rick) requesting DEV index from splunk to point solace using JMS Messaging Modular Input (aka jms\_ta).

**12/04 –** A ServiceNow Request has been submitted with the cloud Analytics team for the work.

**Refer:** REQU0004532177 (Kitzman, Rick assigned this ticket to further work)

**12/05 –** Solace configuration is completed with reference to the below link.

**Refer :** <https://solace.com/blog/devops/splunking-solace-events-jms-modular-input>

A kickoff meeting was scheduled to discuss solace integration with splunk. (Shah, Navneet, Desai, Parth, Mamidi, Ashok Kumar and Kitzman, Rick)

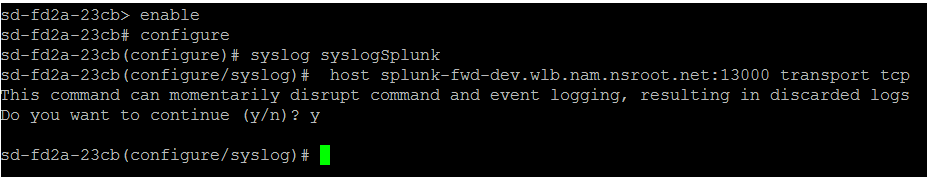
**12/06 –** Team had some concerns on the kickoff meeting and discussed to setup a call with solace team. A Meeting is scheduled with Overton, Ken to discuss further.

(Overton, Ken, Shah, Navneet, Desai, Parth, Mamidi, Ashok Kumar and Kitzman, Rick).

* Ken suggested to go with the syslog forwarding approach for this use-case. Solace configuration is completed according to the below URL from solace.

<https://solace.com/blog/devops/syslog-forwarding-integrate-solace-with-monitoring-systems>

1) Below CLI configuration (On all the three VMR’s) to forward syslog messages on the event facility to a remote host with host:splunk-fwd-dev.wlb.nam.nsroot.net:13000 using transport tcp:



sd-fd2a-23cb> enable

sd-fd2a-23cb# configure

sd-fd2a-23cb(configure)# syslog syslogSplunk

sd-fd2a-23cb(configure/syslog) # host splunk-fwd-dev.wlb.nam.nsroot.net:13000 transport tcp

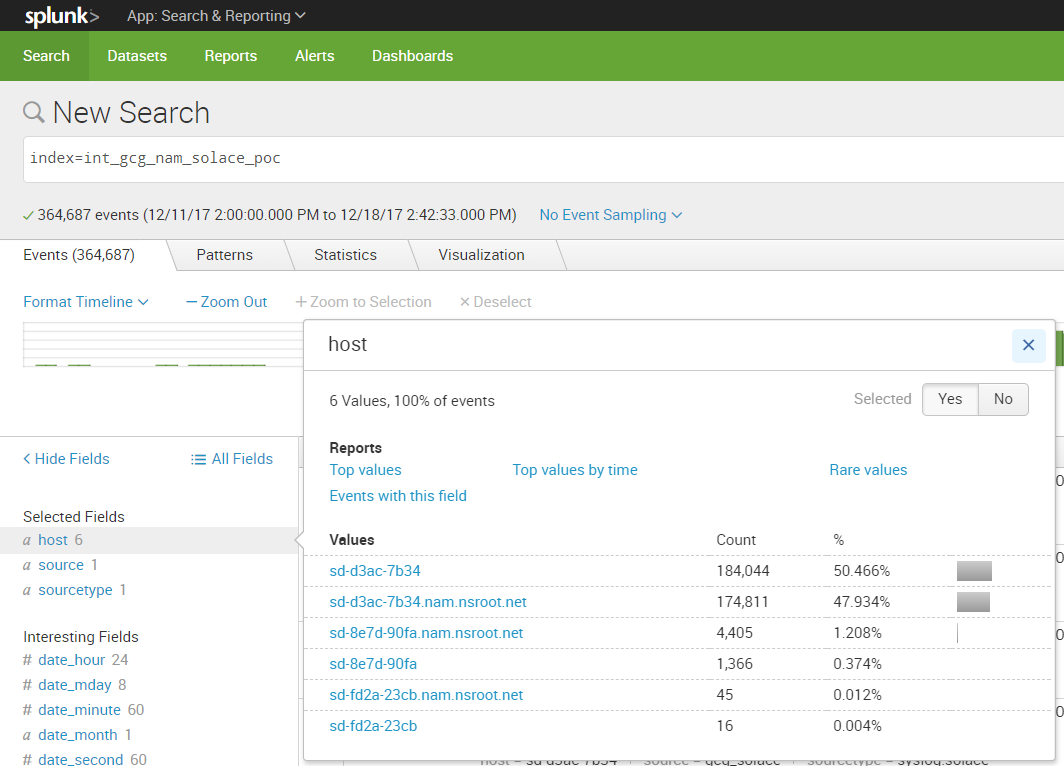
This command can momentarily disrupt command and event logging, resulting in discarded logs

Do you want to continue (y/n)? y

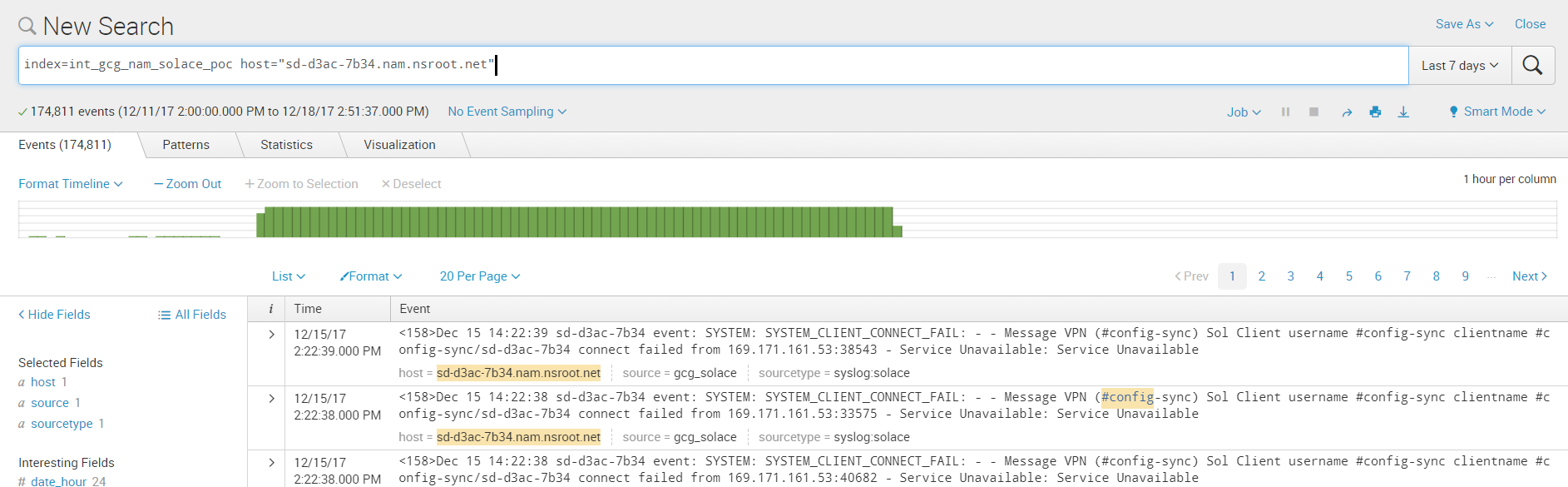
2) Post configuration on the remote end, Syslog events are forwarded from all the 3 nodes as below to Splunk.

Splunk Dev URL: <https://sd-1a1b-a62b:8000/en-US/app/search/search?q=search%20index%3Dint_gcg_nam_solace_poc&display.page.search.mode=smart&dispatch.sample_ratio=1&earliest=-60m%40m&latest=now&sid=1512748864.220171>

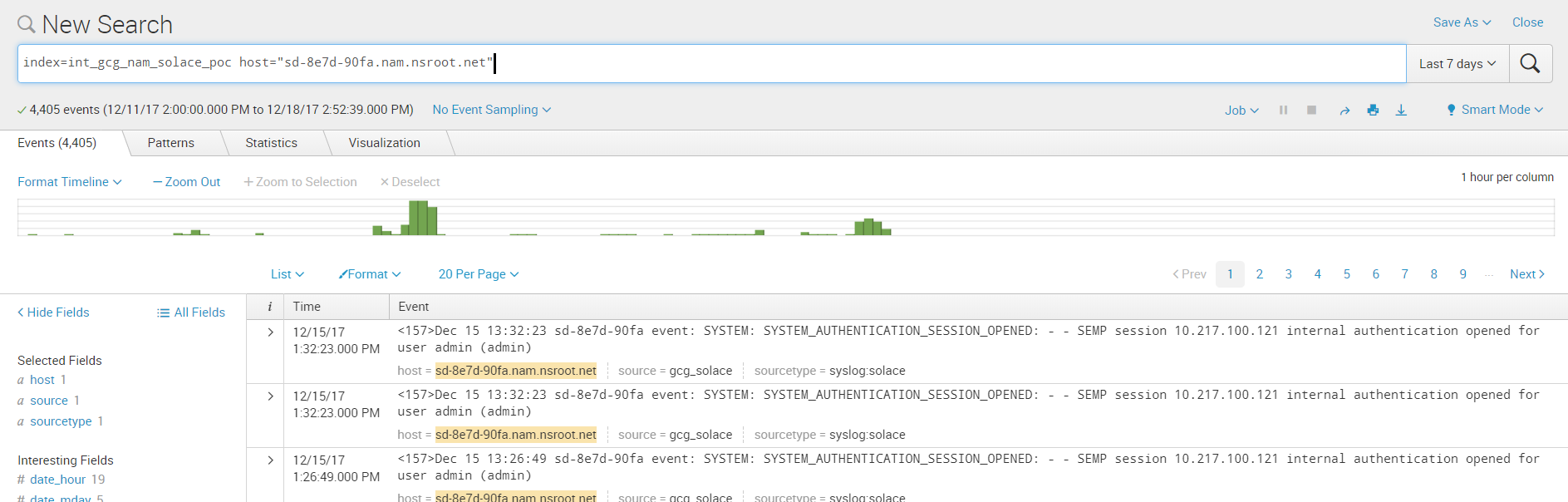
Splunk Index for POC: index=int\_gcg\_nam\_solace\_poc



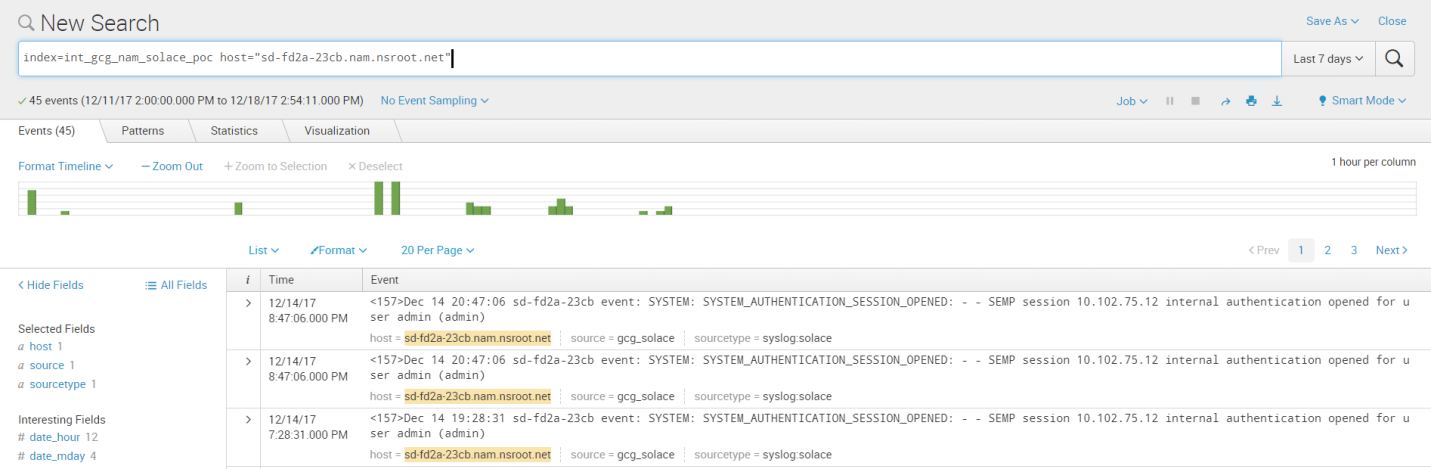
3) Below Snippet shows the primary router (sd-d3ac-7b34 169.171.161.53) sending event logs since last 7 days.



4) Below Snippet shows the backup router (sd-8e7d-90fa 169.171.161.54) sending event logs since last 7 days.



5) Below Snippet shows the monitor router (sd-fd2a-23cb 169.171.161.52) sending event logs since last 7 days.



**Integrating Solace with AppD**

Description – Test the connectivity of solace with AppD and review the metrics.

**Steps Taken –**

11/29 – Kickoff meeting to disucss solace integration with AppD

(Webb, Russell – CATE, Al Naser, Ihab – AppD, Overton, Ken – Solace)

Ken from Solace is working on creating Java extension after discussion with Ihab Al Naser

Ken requested an AppD environment for testing the Java extension. No issues from Citi AppD team.

12/5 – Ken is still working on extension. May need some support from AppD.

**JMS connectivity testing from Tibco BW application with Solace**

Description – Test the JMS connectivity from Tibco BW application with Solace

**Step 1)**

Copy the Jars to below location.

**TIBCO\_HOME/5.X/tpcl/5.X/lib/**

sol-jms-10.2.0-javadoc.jar

sol-common-10.2.0.jar

sol-jcsmp-10.2.0.jar

commons-lang-2.6.jar

commons-logging-1.1.3.jar

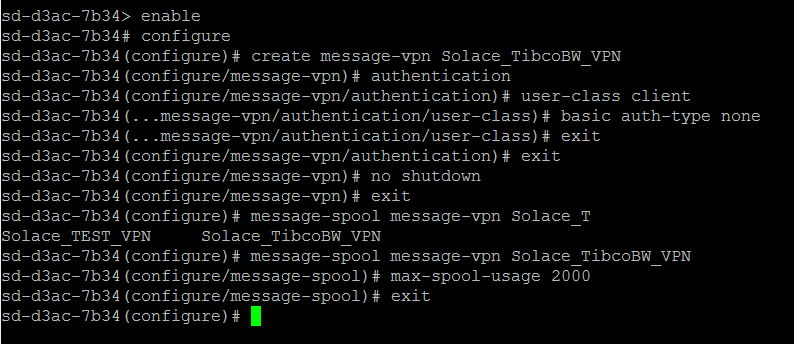
geronimo-jms\_1.1\_spec-1.1.1.jar

org.osgi.annotation-6.0.0.jar

sol-jms-10.2.0.jar

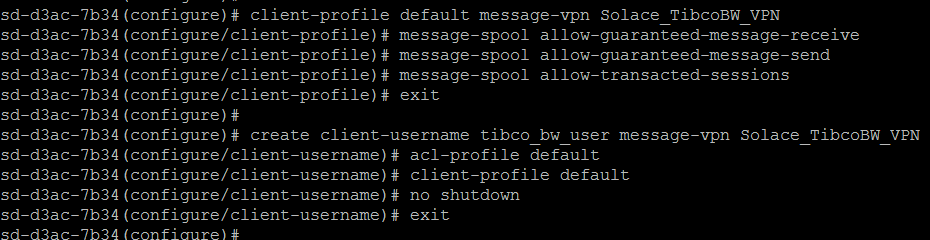
**Step 2)**

Create new message VPN on Solace



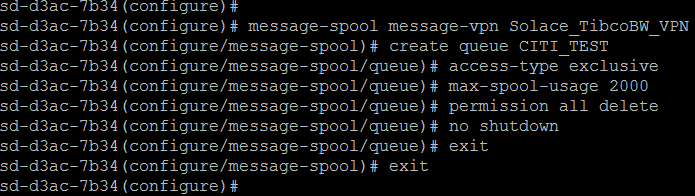
**Step 3)**

Set Client profile to message-vpn and create client-username for tibco bw application to connect with solace



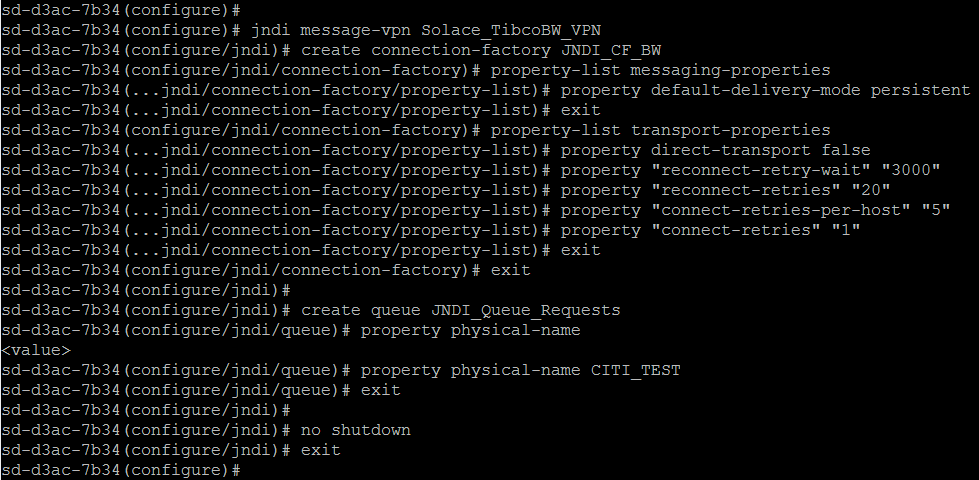
**Step 4)**

Create request queue on Solace



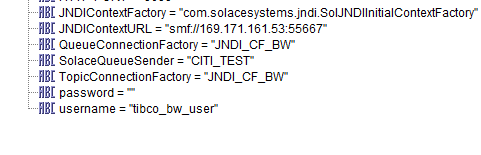
**Step 5)**

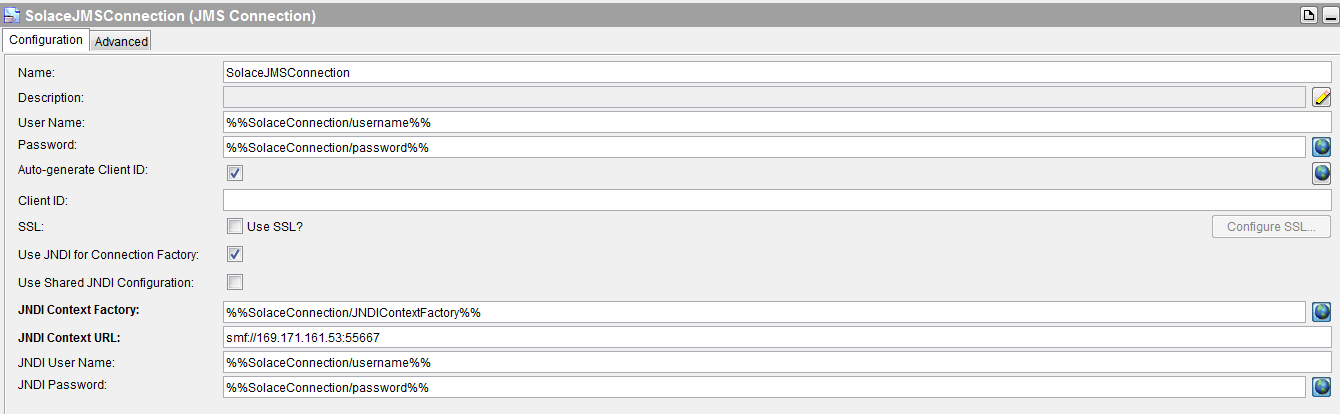
Create JNDI factories and set properties on Solace

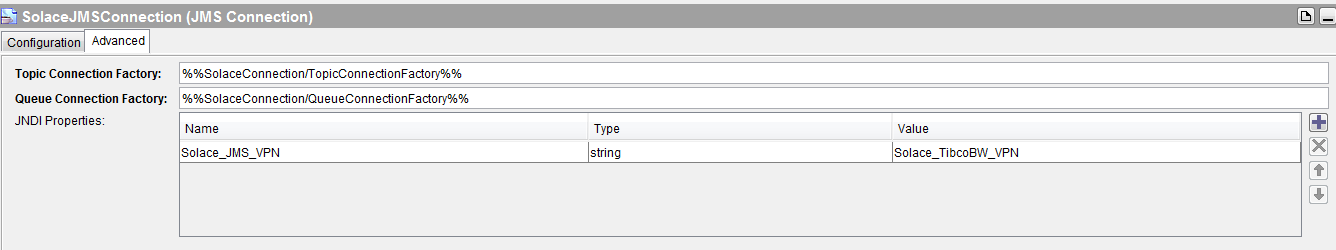


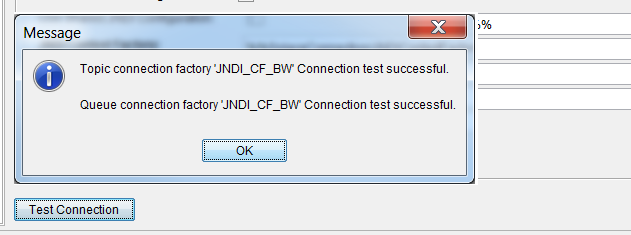
**Step 6)**

Test JMS connectivity from Tibco BW









**JMS Connectivity from Tibco BE with Solace**

1. Place solace JAR files in any specific folder

sol-jms-10.2.0-javadoc.jar

sol-common-10.2.0.jar

sol-jcsmp-10.2.0.jar

commons-lang-2.6.jar

commons-logging-1.1.3.jar

geronimo-jms\_1.1\_spec-1.1.1.jar

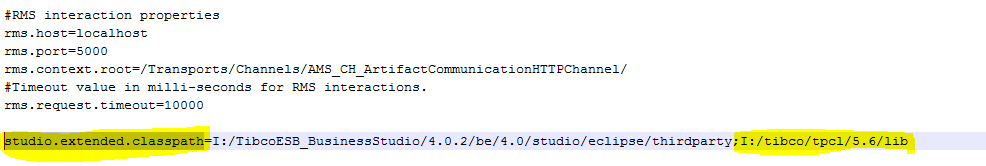
org.osgi.annotation-6.0.0.jar

sol-jms-10.2.0.jar

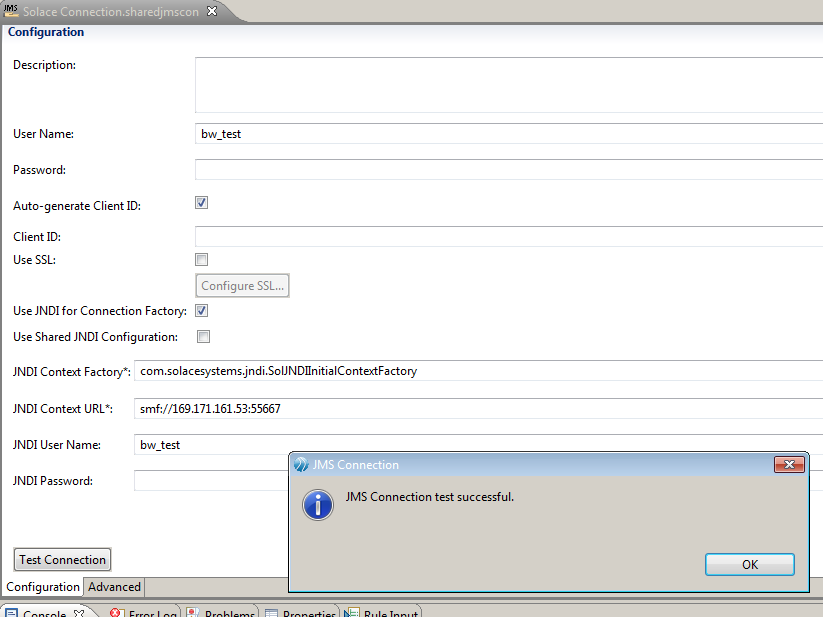
1. Open studio.TRA file from below path (Adapt the path as needed. Might vary as per your local directory structure) :

BE\_HOME/4.0/studio/eclipse/configuration/studio.TRA

1. Append environment variable “studio.extended.classpath” with reference to solace JAR file path (Placed in step 1) as below:



1. Validate JMS Connection from Tibco BE:



**JMS Connectivity from TIBCO iProcess with solace**

1. Create the solace objects such as message-vpn, connection factories, endpoints, username and client acl properties as mentioned in BW connectivity section.
2. Update the below entries in the file $SWDIR/etc/iapjms\_classpath.properties

classpath.internal.common=common\_swprocess\_library.jar,common\_bootstrap\_library.jar,common\_utils\_library.jar,commons-lang-2.6.jar,commons-logging-1.1.3.jar

classpath.ems.1=jms.jar,tibjms.jar,tibjmsadmin.jar,tibcrypt.jar,slf4j-api-1.4.2.jar,slf4j-simple-1.4.2.jar,sol-jms-10.2.2.jar,sol-jcsmp-10.2.2.jar,sol-common-10.2.2.jar

1. Comment the existing EMS configuration and add the solace VMR details in $SWDIR/etc/iapjms.properties

IAPJMSConnect.InitialContextFactory=com.solacesystems.jndi.SolJNDIInitialContextFactory

IAPJMSConnect.InitialURL=smf://10.40.57.174:55555

IAPJMSConnect.TopicConnectionFactory=JNDI/CF/ipe

IAPJMSConnect.SecurityPrinciple=ipe\_user

IAPJMSConnect.SecurityCredentials=

IAPJMSConnect.SecurityEncryption=PLAIN

IAPJMSConnect.Persistent=Y

IAPJMSConnect.Priority=1

IAPJMSConnect.TimeToLive=0

1. Place the below jars in the location $SWDIR/jmslib/ems/

sol-common-10.2.2.jar

sol-jcsmp-10.2.2.jar

sol-jms-10.2.2.jar

1. Place the below jars in the location $SWDIR/jar/

commons-lang-2.6.jar

commons-logging-1.1.3.jar

geronimo-jms\_1.1\_spec-1.1.1.jar

org.osgi.annotation-6.0.0.jar

1. Place the below jars in the location $SWDIR/eaijava/libs/repository/emsclient

sol-common-10.2.2.jar

sol-jcsmp-10.2.2.jar

sol-jms-10.2.2.jar

commons-lang-2.6.jar

commons-logging-1.1.3.jar

geronimo-jms\_1.1\_spec-1.1.1.jar

org.osgi.annotation-6.0.0.jar

log4j-1.2.17.jar

1. Update iProcess EAI JAVA EMS details in DB as below.

Location = $SWDIR/\_otherInfo/Hlocal/delta/

bash-4.1$ cat provider.txt

[provider]

NAME=RELIPRO

URL=smf://10.40.57.174:55555

JNDI\_USER=ipe\_user@iprocess\_vpn

FACTORY=com.solacesystems.jndi.SolJNDIInitialContextFactory

[end]

[provider]

NAME=RELEMS

URL=smf://10.40.57.174:55555

JNDI\_USER=ipe\_user@iprocess\_vpn

FACTORY=com.solacesystems.jndi.SolJNDIInitialContextFactory

[end]

bash-4.1$ cat dest.txt

[dest]

EndPointName=RELIPRO

JNDIQuqueName=TestQueueiPe3

FACTORY=JNDI/CF/ipe

ProviderName=RELIPRO

[end]

[dest]

EndPointName=RELEMS

JNDIQuqueName=TestQueueiPe4

FACTORY=JNDI/CF/ipe

ProviderName=RELEMS

[end]

Location = $SWDIR/\_otherInfo/Hlocal/bin

Execute below command

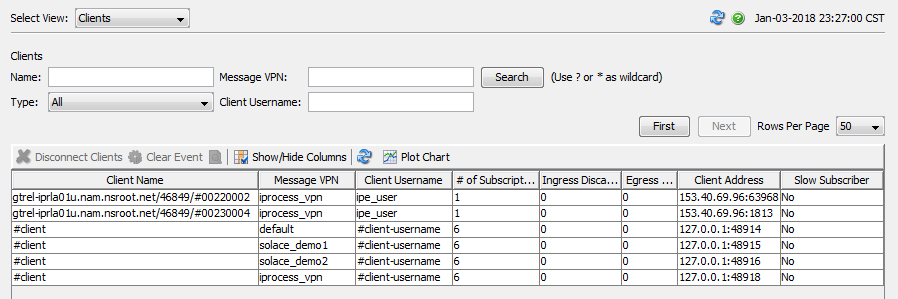
./iProcStart

1. Execute the below command to update message-vpn

$SWDIR/util/swadm set\_attribute 0 IAPJMS 0 JVMPROPS -DSolace\_JMS\_VPN=iprocess\_vpn

1. Shutdown the iProcess node and start it with clean logs.
2. Check the IAPJMS connection status at Solace VMR once the node is up and running fine.

Connection from IAPJMS will be established at Solace VMR side.



1. Initiate some EAI actions from iProcess workspace or CI/SOA eclipse front end portal to check the EAI JAVA call.

if the EAI actions has JMS call, then we can see the message is getting posted at the respective queue which will be waiting to be get consumed by BW.

