# SYSLOG in ACI

## Overview, Configuration, Troubleshooting, and Caveats\Issues

Created by Tomas de Leon (ACI Solutions Delivery Team)

*Version 1, created December 1, 2016*

# Table of Contents

# Table of Contents (cont.)

# ACI SYSLOG Overview

Syslog is a method to collect messages from devices to a server running a syslog daemon. Logging to a central syslog server helps in aggregation of logs and alerts. *During operation, a fault or event in the Cisco Application Centric Infrastructure (ACI) system can trigger the sending of a system log (syslog) message to the console and, optionally, to a logging server on another system. A system message typically contains a subset of information about the fault or event, and the message is can sent by using syslog feature in the ACI system.*

# ACI SYSLOG Overview

## About System Messages

Syslog is a method to collect messages from devices to a server running a syslog daemon. Logging to a central syslog server helps in aggregation of logs and alerts. During operation, a **fault** or **event** in the Cisco Application Centric Infrastructure (ACI) system can trigger the sending of a system log (syslog) message to the console and, optionally, to a logging server on another system. A system message typically contains a subset of information about the fault or event, and the message is can sent by using syslog feature in the ACI system.

Many system messages are specific to the action that a user is performing or the object that a user is configuring or administering. These messages can be the following:

- ***Informational messages***, *providing assistance and tips about the action being performed*
- ***Warning messages***, *providing information aboutsystem errors related to an object, such as a user account or service profile, that the user is configuring or administering*
- ***Finite state machine (FSM) status messages***, *providing information about the status of an FSM stage*

A system message can contain one or more variables. The information that the system uses to replace these variables depends upon the context in which you see the message. Some messages can be generated by more than one type of condition.

# ACI SYSLOG Overview (cont.)

## Fault Syslogs

Fault-generated system messages are triggered by these mechanisms:

- *A fault rule*
- *A threshold crossing*
- *A failure of a task or finite state machine (FSM) sequence*

The fault-generated system messages are described in the *Cisco APIC Management Information Model Reference*, which is a web-based application. Under the **Syslog Messages** navigation tab, select **Syslog Faults** or **Syslog FSM Transitions**.

# ACI SYSLOG Overview (cont.)

## Event Syslogs

Event-generated system messages are triggered by these mechanisms:

- *An event rule*
- *An event in the NX-OS operating system of a leaf or spine switch*

The event rule-generated system messages are described in the *Cisco APIC Management Information Model Reference*, which is a web-based application. Under the **Syslog Messages** navigation tab, select **Syslog Events**.

The NX-OS operating system event messages are listed in the *Cisco ACI System Messages Reference Guide.*

*Note: Not all system messages indicate problems with your system. Some messages are purely informational, while others may help diagnose problems with communications lines, internal hardware, or the system software.*

# ACI SYSLOG Overview (cont.)

## ACI System Message Structure

System messages generated by ACI components other than NX-OS are structured as follows:

*timestamp host %LOG_LOCALn-severity-SYSTEM_MSG [code][lifecycle state][rule][severity text] [DN of affected MO]*

*Message-text*

The fields in the messsage are as follows:

**timestamp**
The year, month, date, and time of day when the message was generated.

**host**
The hostname or IP address of the host that generated the message, such as ‘apic1’, ‘leaf1’, or ‘spine1.’

# ACI SYSLOG Overview (cont.)

## ACI System Message Structure (cont.)

additional fields in the messsage are as follows:

**%LOG_LOCALn**

The local facility code 'n' is a single-digit code from 0 to 7 that reflects indicate the local facility of the message. This number can be configured and is used to sort received messages.

**severity**

The severity level is a single-digit code from 1 to 5 that reflects the severity of the condition. The lower the number, the more serious the situation. Unlike NX-OS system messages, ACI system messages follow the ITU Perceived Severity values described in **RFC5674**. The following Table lists the message severity levels.

# ACI SYSLOG Overview (cont.)

## ACI System Message Structure (cont.)

The following Table lists the ACI message severity levels.

| | Severity Level (NX-OS) | ITU Level (ACI) | Description |
|---|---|---|---|
| 0 | emergency | | System is unusable |
| 1 | alert | Critical | Immediate action required |
| 2 | critical | Major | Critical condition |
| 3 | error | Minor | Error condition |
| 4 | warning | Warning | Warning condition |
| 5 | notification | Cleared | Normal but significant condition |
| 6 | informational | (Info) | Informational message only |
| 7 | debugging | (Not used) | Message that appears during debugging only |

# ACI SYSLOG Overview (cont.)

## ACI System Message Structure (cont.)

additional fields in the messsage are as follows:

**code**
The unique fault or event code.

**lifecycle state**
The current lifecycle state of the fault. Faults are stateful, and a fault transitions through more than one state during its life cycle. Events are stateless, and this field is omitted in event system messages.

**rule**
The action or condition that caused the event, such as a component failure or a threshold crossing.

# ACI SYSLOG Overview (cont.)

## ACI System Message Structure (cont.)

additional fields in the messsage are as follows:

**severity text**
The text translation of the numeric severity value. For example “major.”

**DN of affected MO (managed object)**
The distinguished name (DN) of the managed object (MO) affected by the fault condition or event.

**Message-text**
Message-text is a text string that briefly describes the condition. The text string sometimes contains detailed information about the fault or event, including interface port numbers or network addresses.

# ACI SYSLOG Overview (cont.)

## ACI System Message Examples

### Example: Fault system message

The following example shows an ACI system message generated by a fabric node failure:

<1026> Aug 01 15:40:25 **fab2-p1-apic1** %LOG_LOCAL0-**2**-SYSTEM_MSG [**F0321**][**soaking**][**unhealthy**][critical][**topology/pod-1/node-1/av/node-1**/**fault-F0321**] **Controller 1 is unhealthy because: Data Layer Partially Diverged**

In this system message:

- "**fab2-p1-apic1**" indicates that this message is generated by the controller.
- "**2**" (Major) is the severity level, indicating a "**Critical**" condition
- "**F0321**" is the fault code, which we can look up as "**fltInfraWiNodeHealth**" in the API documentation.
- "**soaking**" is the current lifecycle state of the fault.
- "**unhealthy**" is the cause of the fault.
- "**topology/pod-1/node-1/av/node-1**" is the DN of the affected object, which is node 1 in pod 1.
- "**fault-F0321**" is the DN of the fault object, which is a child of the affected object.

*Tip: Using the Visore object browser, you can inspect the properties of the fault object for more details about the fault condition.*

- "**Controller 1 is unhealthy because: Data Layer Partially Diverged**" is the message text.

# ACI SYSLOG Overview (cont.)

## ACI System Message Examples

### Example: Event system message

The following example shows an ACI system message generated by a fabric node failure:

<158> Aug 01 16:00:04 **fab2-p1-apic1** %LOG_LOCAL3-**6**-SYSTEM_MSG [**E4208219**][**link-state-change**][info] [**topology/pod-1/lnkcnt-216/lnk-2101-1-9-to-216-1-49**] **Link State of Fabric Link is set to ok**

In this system message:

- "**fab2-p1-apic1**" indicates that this message is generated by the controller.
- "**6**" (Info) is the severity level, indicating an "informational" message.
- "**E4208219**" is the fault code, which we can look up as "**fabric_Link_linkStateChange**" in the API documentation.
- "**link-state-change**" is the cause of the fault.
- "**topology/pod-1/lnkcnt-216/lnk-2101-1-9-to-216-1-49**" is the DN of the affected object, which is a link.

*Tip: Using the Visore object browser, you can inspect the properties of the fault object for more details about the fault condition.*

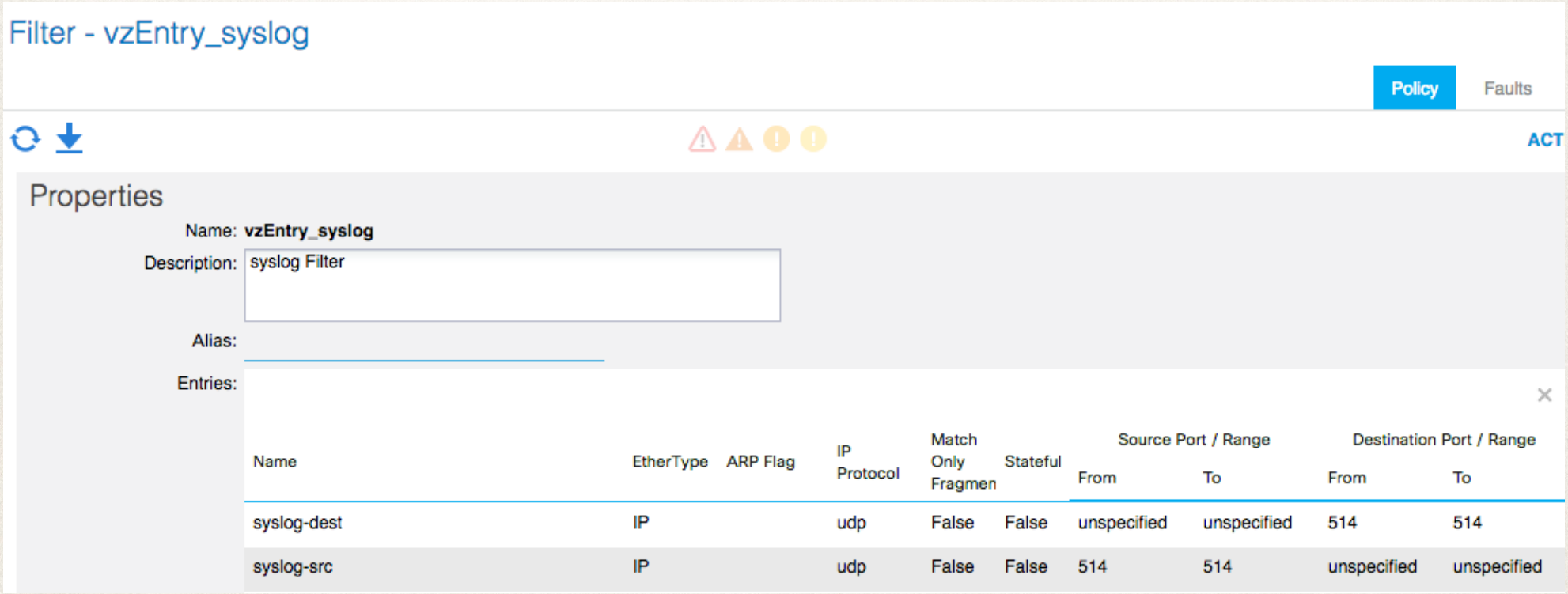
- "**Link State of Fabric Link is set to ok**" is the message text.

# SYSLOG Support in the ACI (cont.)

- Management Contracts required for SYSLOG
  - SYSLOG on APIC using OOB management EPG **does not require** an explicit "**Out-Of-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**). That said, it is a good practice to go ahead and create a specific filter for SYSLOG and add it to the filter list in your OOB Contract Subject configuration. *Note: In earlier versions of ACI firmware, certain ports were always open and a contract was not needed for SYSLOG support on the Leaf and Spine nodes.*
  - SYSLOG on APIC using INB management EPG **requires** an explicit "**In-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**).
  - Unless the contract is created, The SYSLOG packets will be dropped by the Border Leaf with the L3 Out used by the fabric for MGMT Access. ***This is different from enabling/disabling the SYSLOG protocol in monitoring policies.***

# SYSLOG Support in the ACI (cont.)

- Example of Syslog Filter and Contract Subject



Filter - vzEntry_syslog

Policy | Faults

ACT

Properties

Name: **vzEntry_syslog**

Description: syslog Filter

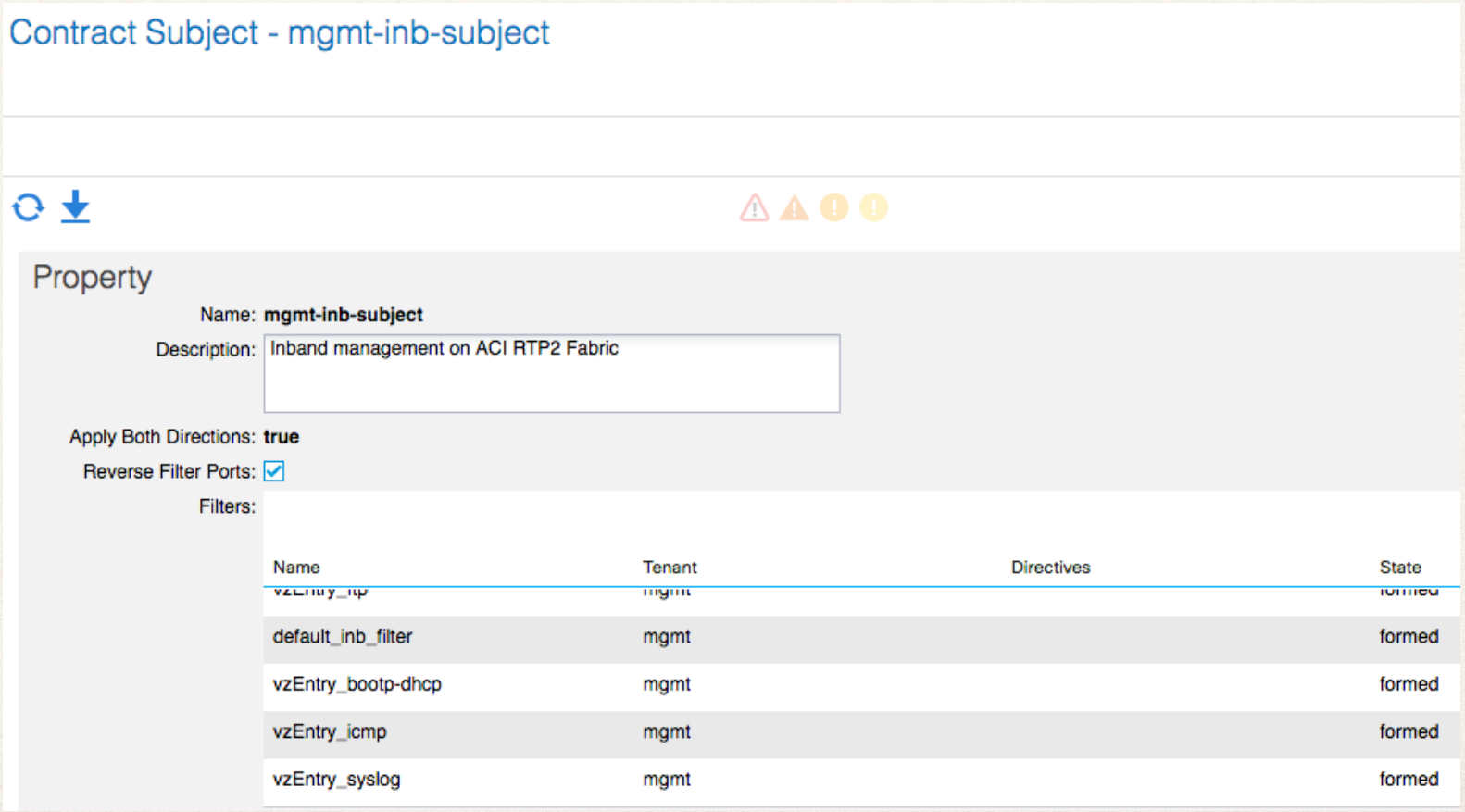
Alias:

Entries:

| Name | EtherType | ARP Flag | IP Protocol | Match Only Fragmen | Stateful | Source Port / Range | | Destination Port / Range | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | From | To | From | To |
| syslog-dest | IP | | udp | False | False | unspecified | unspecified | 514 | 514 |
| syslog-src | IP | | udp | False | False | 514 | 514 | unspecified | unspecified |

# SYSLOG Support in the ACI (cont.)

- Example of Syslog Filter and Contract Subject (cont.)



# About this Technote on SYSLOG in ACI

The following document will use examples from using a "SYSLOG" utility or CLI commands to gather information about the Cisco ACI fabric system. The "SYSLOG" utility will receive SYSLOG messages sent by the individual leaf & spine switches and APIC controllers. Not all SYSLOG messages indicate problems with your system. Some messages are purely informational, while others may help diagnose problems with communications lines, internal hardware, or the system software. This document will not cover configuring 3rd Party SYSLOG monitoring utilities. *Just make sure the SYSLOG Utility has the ACI nodes IP addresses (INB & OOB) configured as SYSLOG input sources & the correct UDP port used in the ACI Fabric for SYSLOG messages.*

In this technote, I will show examples of configuring SYSLOG **utilizing the APIC Admin GUI**. In ACI version 1.2(xx) or later, there are two modes for the APIC Admin GUI. For this document, **I will use examples from the "ADVANCED" GUI Mode**. In addition to the APIC Admin GUI, SYSLOG can be configured using the APIC iNXOS CLI Mode and by using a REST API client. **[2] [3]**

# About this Technote on SYSLOG in ACI

**Note**: At the time of writing this document, configuring SYLOG using the APIC iNXOS CLI Mode is similar to using the "Basic" GUI mode. For more advanced configuration settings, the SYLOG configuration should be configured via the Advanced GUI or the Rest API. In regards to the REST API, you can open the API inspector console from the APIC GUI. The API inspector displays the Rest API POST requests used for the tasks performed. The "Post" Requests in the API inspector can be used for sending requests to APIC controllers.

For Rest API examples listed in this document, there is an assumption made that you have a REST CLIENT (like POSTMAN) installed on your workstation. This is a sample tool that can be used for executing REST API requests to an APIC Controller.

# ACI SYSLOG Configuration

In this technote, I will show examples of configuring SYSLOG **utilizing the APIC Admin GUI**. In ACI version 1.2(xx) or later, there are two modes for the APIC Admin GUI. For this document, **I will use examples from the "ADVANCED" GUI Mode**. In addition to the APIC Admin GUI, SYSLOG can be configured using the APIC iNXOS CLI Mode and by using a REST API client. **[2] [3]**

# ACI SYSLOG Configuration

## Configure using the APIC iNXOS CLI

In ACI version 1.2(xx) or later, SYSLOG can be configured using the APIC iNXOS CLI Mode. It has been noted that configuring SYSLOG using the CLI is limited and can be incomplete. Basic Syslog configuration can be configured and send messages successfully using the APIC iNXOS CLI mode . **[2] [3]**

# Configure a SYSLOG using the APIC iNXOS CLI

- For this configuration example, the goal is to configure your ACI Fabric for sending Syslog messages using the APIC iNXOS CLI.
  - *Configure two Syslog Server destinations.*
  - *Send Syslog messages using the In-Band & Out-of-Band Mgmt EPGs*

- **<u>Configuration Steps (using the APIC iNXOS CLI):</u>**
  1. Create a Syslog Monitoring Destination Group
  2. Add Remote Server Destinations to the Syslog Monitoring Destination Group
  3. Configure a Syslog Monitoring Source to use the Syslog Monitoring Destination Group
  4. Verify the Syslog configuration
  5. Test the ACI Fabric Syslog configuration using the CLI Syslog Test Feature

*NOTE: These CLI configuration steps will configure the ACI Fabric to send Syslog messages to remote servers, but there are some other parameters that can be configured via the APIC Admin GUI. These parameters can further customize the ACI Fabrics Syslog configuration.*

# TASK 1:
# Create a SYSLOG Monitoring Destination Group

- Configuration Steps (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1
  2. Use the CLI mode "configure" to enter configuration mode
  3. Use the "logging server-group" command to create Syslog Destination Group
  4. Add a "description" for the Syslog Destination Group
  5. Add a "server" with parameters for severity, port, facility, and mgmt EPGS. Add multiple servers so that one server utilizes the **In-Band mgmt EPG** and the other utilizes the **Out-of-Band EPG**.

CLI configuration Example:

```
apic1# configure
apic1(config)# logging server-group deadbeef-syslogGrp
apic1(config-logging)# description "Syslog Server for the deadbeef network"
apic1(config-logging)# server 10.122.254.251 severity information port 514 facility local7 mgmtepg oob
apic1(config-logging)# server 10.117.67.29 severity information port 514 facility local5 mgmtepg inb
apic1(config-logging)# exit
```

# TASK 2:
# Configure a Syslog Monitoring Source to use the Syslog Monitoring Destination Group

- Configuration Steps (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1
  2. Use the CLI mode "configure" to enter configuration mode
  3. Use the "syslog common" command to configure a Syslog Monitoring Source.
  4. Add a "description" for the Syslog Monitoring Source.
  5. Use the "logging severity" command to set the minimum severity level to monitor.
  6. Also configure the Syslog Monitoring Source to include mulitple information sources in the syslog messages. Use the "logging" command to configure which information sources to include in Syslog Messages. Configure the Syslog Monitoring Source to include audit, event, fault, and session information.
  7. Use the "logging server-group" command to configure which Syslog Monitoring Destination Group to use.

CLI configuration Example:

```
apic1(config)# syslog common
apic1(config-syslog)# logging description "Syslog Policy created by deadbeef"
apic1(config-syslog)# logging severity information
apic1(config-syslog)# logging audit
apic1(config-syslog)# logging event
apic1(config-syslog)# logging fault
apic1(config-syslog)# logging session
apic1(config-syslog)# logging server-group deadbeef-syslogGrp
```

# TASK 3:
# Verify the Syslog configuration

- Steps to Verify Configuration (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1
  2. Use the "**show running-config syslog**" & "**show running-config logging server-group**" commands to verify Syslog.

CLI configuration Example:

```
apic1# show running-config syslog
# Command: show running-config syslog
 syslog common
   logging description "Syslog Policy created by deadbeef"
   logging audit
   logging event
   logging fault
   logging session
   logging severity information
   logging server-group deadbeef-syslogGrp
   exit

apic1# show running-config logging server-group deadbeef-syslogGrp
# Command: show running-config logging server-group deadbeef-syslogGrp
 logging server-group deadbeef-syslogGrp
   description "Syslog Server for the deadbeef network"
   logfile
   console
   server 10.117.67.29 severity information facility local5 mgmtepg inb port 514
   server 10.122.254.251 severity information facility local7 mgmtepg oob port 514
   exit
```

# TASK 4:
# Test the Syslog configuration using the CLI Syslog Test Feature

- Test Syslog Configuration Steps (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1.
  2. Use the "**logit**" CLI command to verify Syslog configuration.
     *Note: The Syslog "**logit**" test comand will be available in ACI versions from Congo Maintance Releases and later.*
  3. Perform a "**logit**" test for each configured remote destination. Use Node 1 (APIC1) for each test.

**Command Syntax:**

# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message>
# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message> node <id>

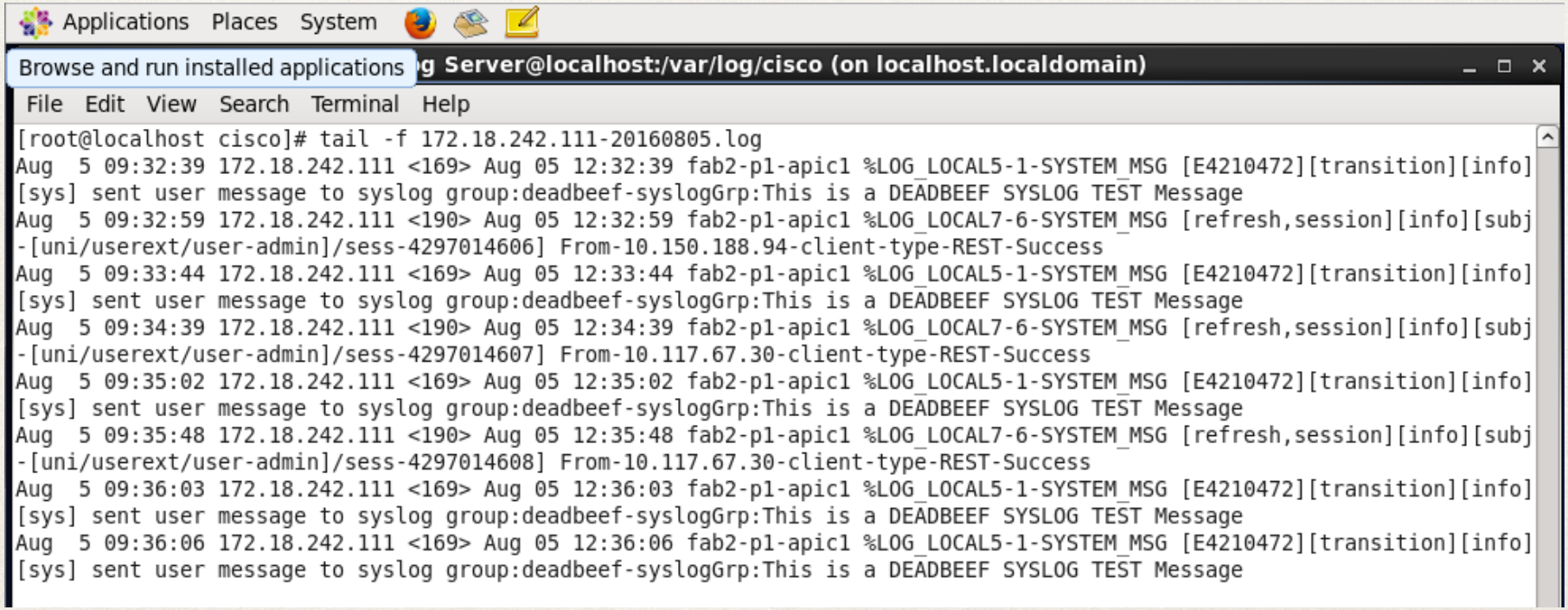
*Remember to run a test for each configured Syslog remote destination.*

# TASK 4:
# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **INB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.117.67.29 "This is a DEADBEEF SYSLOG TEST Message"** node **1**

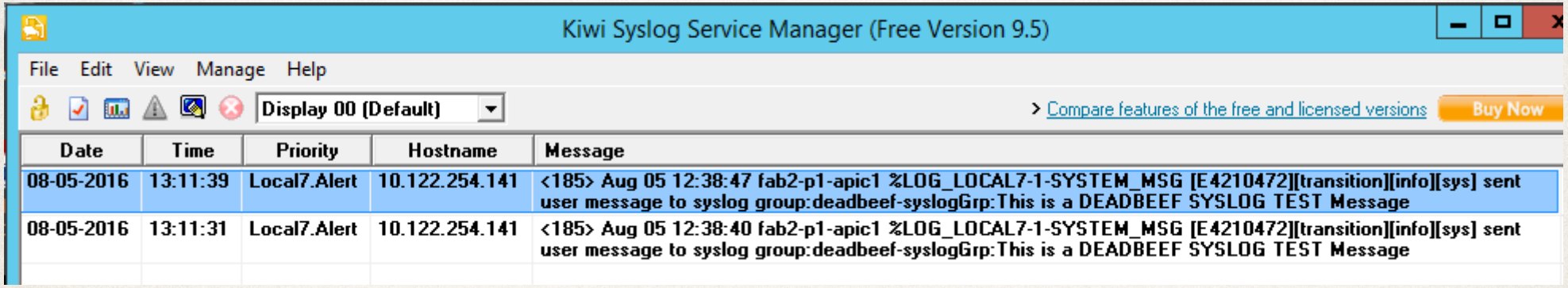


# TASK 4:
# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **OOB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.122.254.251 "This is a DEADBEEF SYSLOG TEST Message"** node **1**




**You can download a free version fo the Kiwi Syslog Server for Windows at:**
http://www.kiwisyslog.com

# ACI SYSLOG Configuration

## Configure using the APIC Admin GUI (Advanced mode)

In this section, I will show examples of configuring SYSLOG **utilizing the APIC Admin GUI**. In ACI version 1.2(xx) or later, there are two modes for the APIC Admin GUI. For this document, **I will use examples from the "ADVANCED" GUI Mode**. **[2] [3]**

# TASK 1:
# Configure MGMT Contracts to allow UDP Port 514 for SYSLOG Messages

- Management Contracts required for SYSLOG
  - SYSLOG on APIC using OOB management EPG **does not require** an explicit "**Out-Of-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**). That said, it is a good practice to go ahead and create a specific filter for SYSLOG and add it to the filter list in your OOB Contract Subject configuration. *Note: In earlier versions of ACI firmware, certain ports were always open and a contract was not needed for SYSLOG support on the Leaf and Spine nodes.*
  - SYSLOG on APIC using INB management EPG **requires** an explicit "**In-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**).
  - Unless the contract is created, The SYSLOG packets will be dropped by the Border Leaf with the L3 Out used by the fabric for MGMT Access. ***This is different from enabling/disabling the SYSLOG protocol in monitoring policies.***

# TASK 1:
# Configure MGMT Contracts to allow UDP Port 514 for SYSLOG Messages

- If Out-Of-Band or In-Band Contract(s) already exist, verify that UDP Port 514 is configured for SYSLOG messages. If Syslog port is not in filters, add UDP Port 514 to existing filters & contracts. Create the Required Contracts & filters with the appropriate Syslog Ports.
- Example of Syslog Filter and Contract Subject

Filter - vzEntry_syslog

Policy | Faults

ACT

Properties

Name: **vzEntry_syslog**

Description: syslog Filter

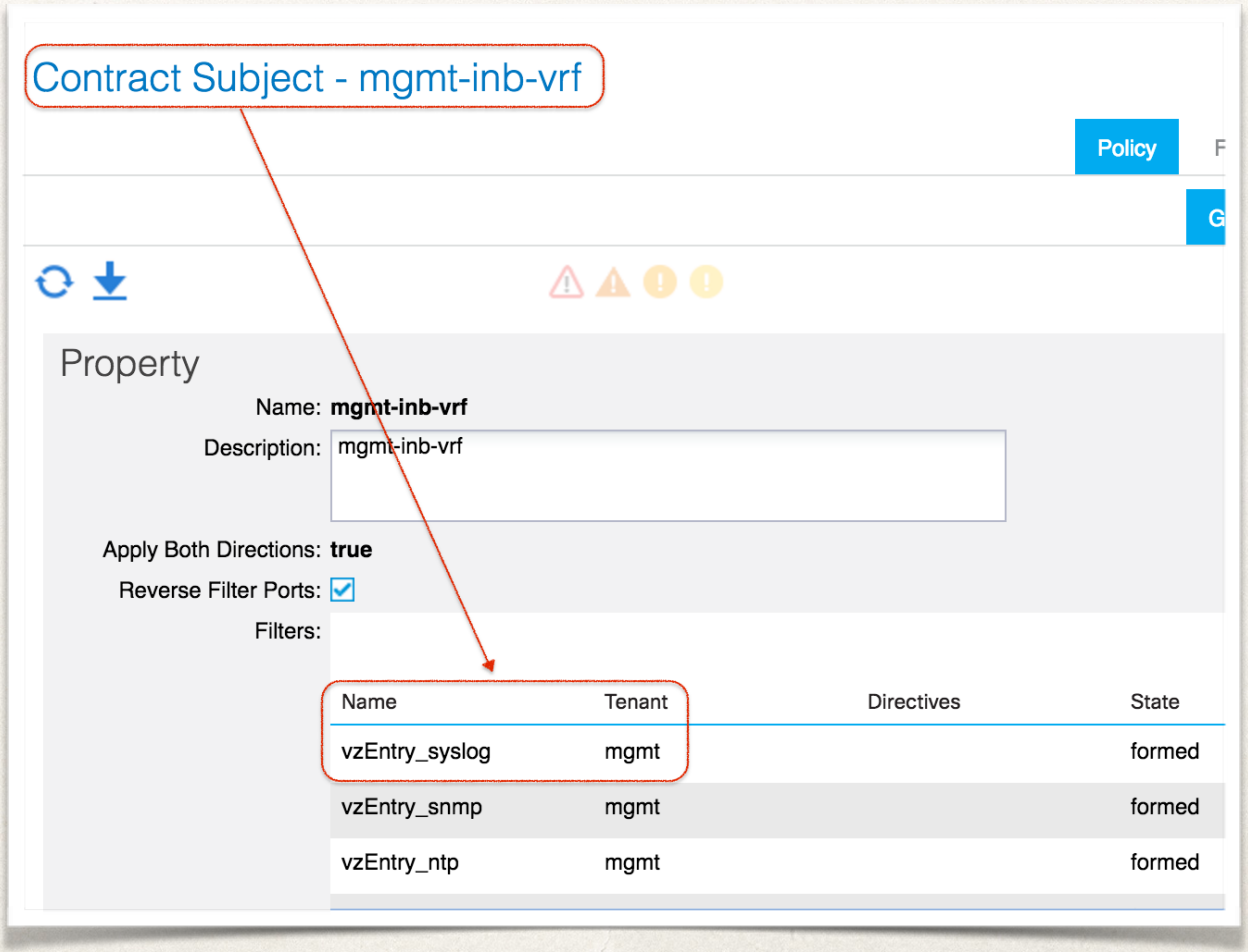
Alias:

Entries:

| Name | EtherType | ARP Flag | IP Protocol | Match Only Fragmen | Stateful | Source Port / Range From | Source Port / Range To | Destination Port / Range From | Destination Port / Range To |
|---|---|---|---|---|---|---|---|---|---|
| syslog-dest | IP | | udp | False | False | unspecified | unspecified | 514 | 514 |
| syslog-src | IP | | udp | False | False | 514 | 514 | unspecified | unspecified |

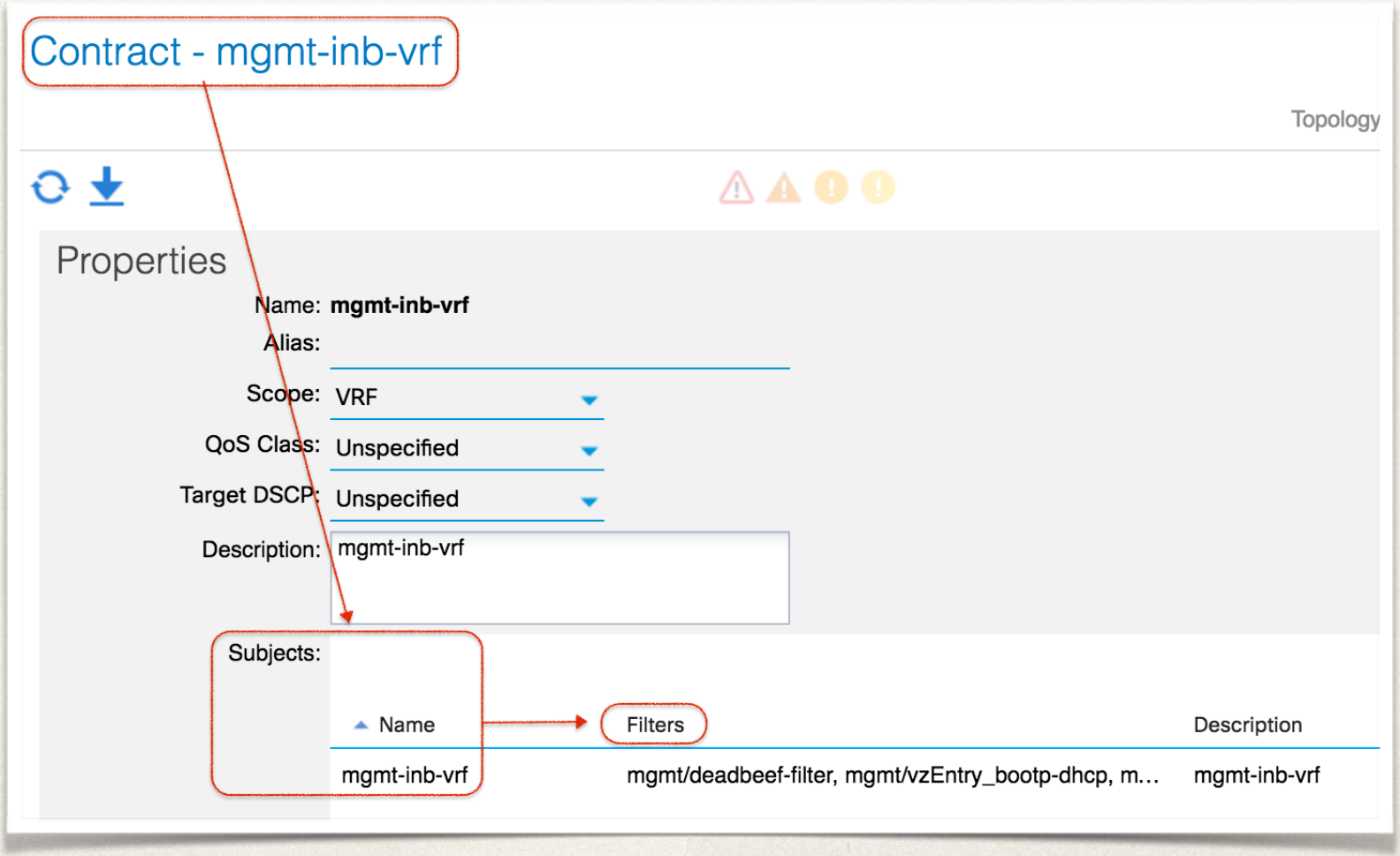
# TASK 1:
# Configure MGMT Contracts to allow UDP Port 514 for SYSLOG Messages

- Example of Syslog Filter and Contract Subject (cont.)



# TASK 1:
# Configure MGMT Contracts to allow UDP Port 514 for SYSLOG Messages

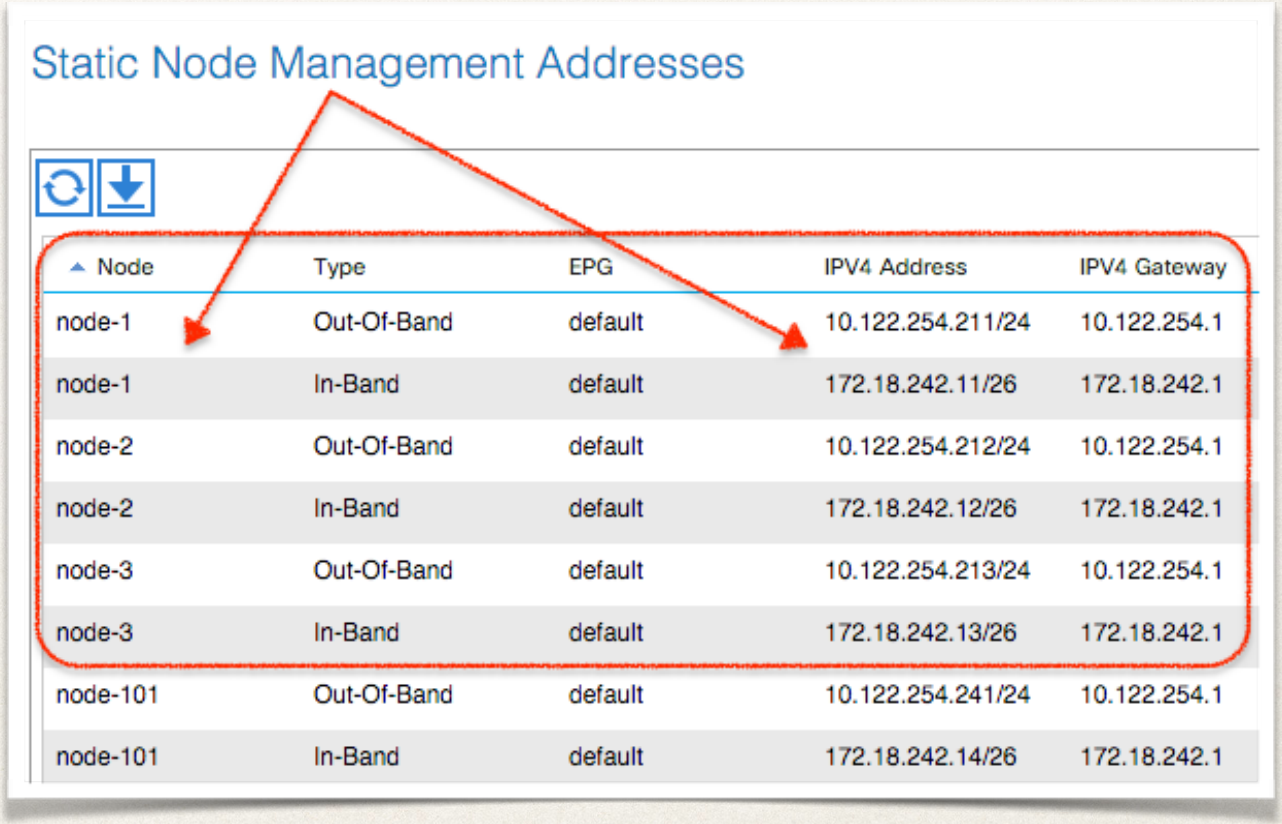
- Verify In-Band Contract:



# TASK 1:
# Configure MGMT Contracts to allow UDP Port 514 for SYSLOG Messages

- Verify Node Management Addresses Configured:

*Note: Node Management Address configurations for OOB and INB network interfaces on each ACI node will be required to be configured based on each management EPG that your fabric is using. For policies like Syslog & SNMP, node management address configuration policies are required for each node.*

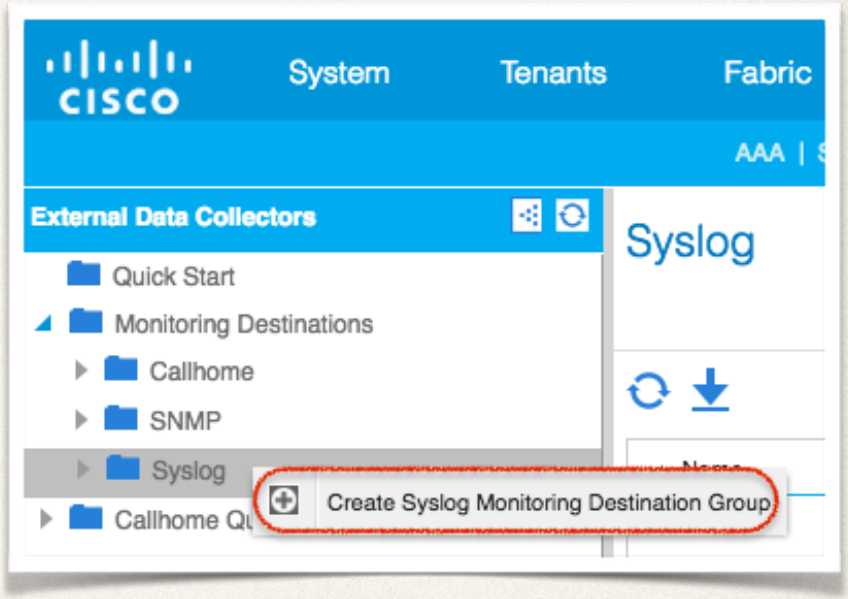



| Node | Type | EPG | IPV4 Address | IPV4 Gateway |
|---|---|---|---|---|
| node-1 | Out-Of-Band | default | 10.122.254.211/24 | 10.122.254.1 |
| node-1 | In-Band | default | 172.18.242.11/26 | 172.18.242.1 |
| node-2 | Out-Of-Band | default | 10.122.254.212/24 | 10.122.254.1 |
| node-2 | In-Band | default | 172.18.242.12/26 | 172.18.242.1 |
| node-3 | Out-Of-Band | default | 10.122.254.213/24 | 10.122.254.1 |
| node-3 | In-Band | default | 172.18.242.13/26 | 172.18.242.1 |
| node-101 | Out-Of-Band | default | 10.122.254.241/24 | 10.122.254.1 |
| node-101 | In-Band | default | 172.18.242.14/26 | 172.18.242.1 |

# TASK 2:
# Configure a SYSLOG Policy for the ACI Fabric

- For this configuration task, we will use the "**Advanced Admin GUI**" interface to configure the Syslog policy for the ACI Fabric.

- Configuration Steps:
  1. Access the APIC Admin GUI.
  2. Select **ADMIN -> EXTERNAL DATA COLLECTORS**.
  3. In the policies navigation panel on the left, select and expand the **MONITORING DESTINATIONS**.
  4. Select "**SYSLOG**" and Click on the " + " sign to **CREATE SYSLOG MONITORING DESTINATION GROUP**.

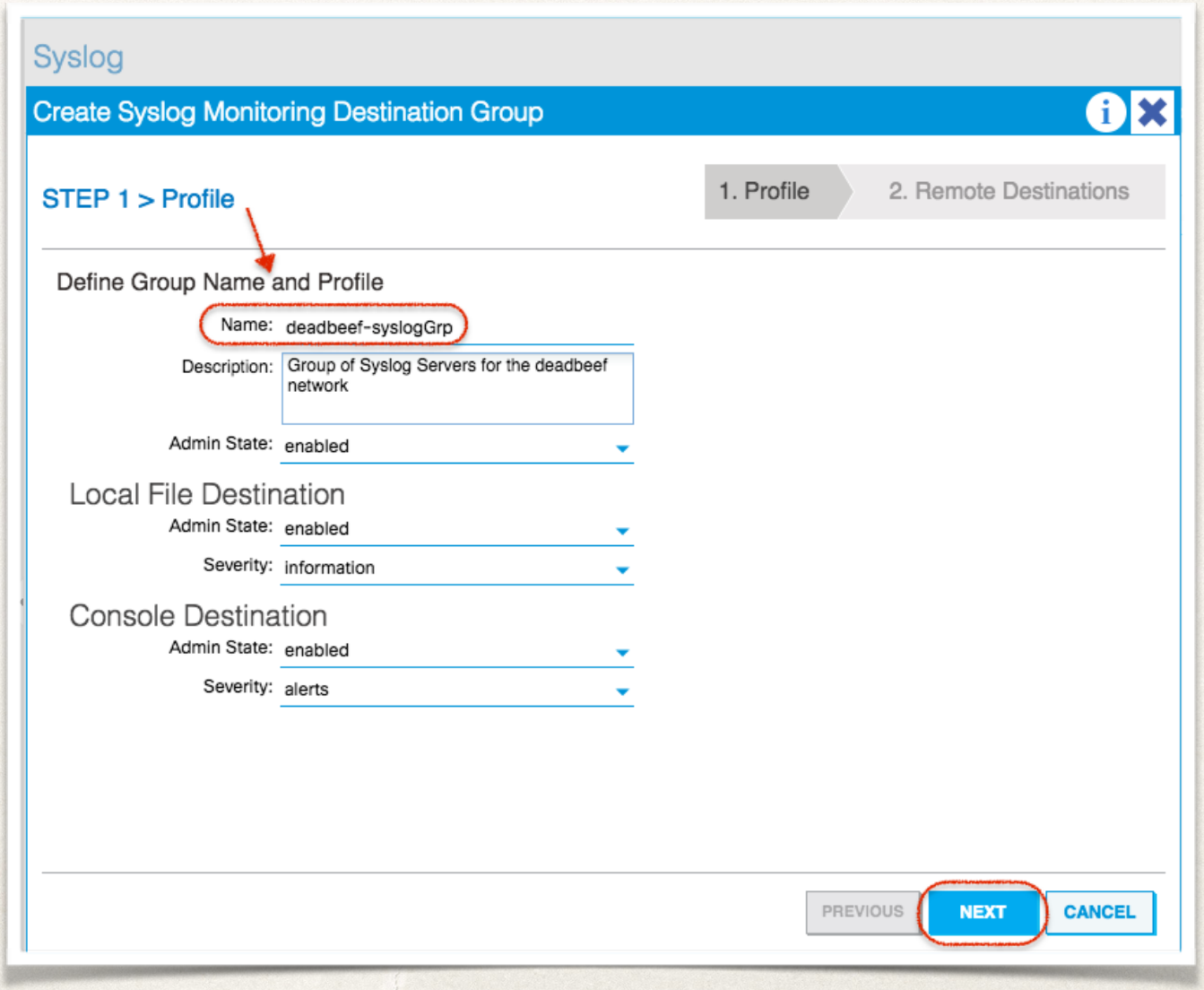


# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

5. In the **CREATE SYSLOG MONITORING DESTINATION GROUP - PROFILE** configuration panel, perform the following actions:
   - Add Name (deadbeef-syslogGrp)
   - Add a description (Group of Syslog Servers for the deadbeef network)
   - Select Admin State (enabled)
   - Select Admin State & Severity for Local File Destination (enabled & information)
   - Select Admin State & Severity for Console Destination (enabled & alerts)
   - Click **NEXT**

# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

5. For example:



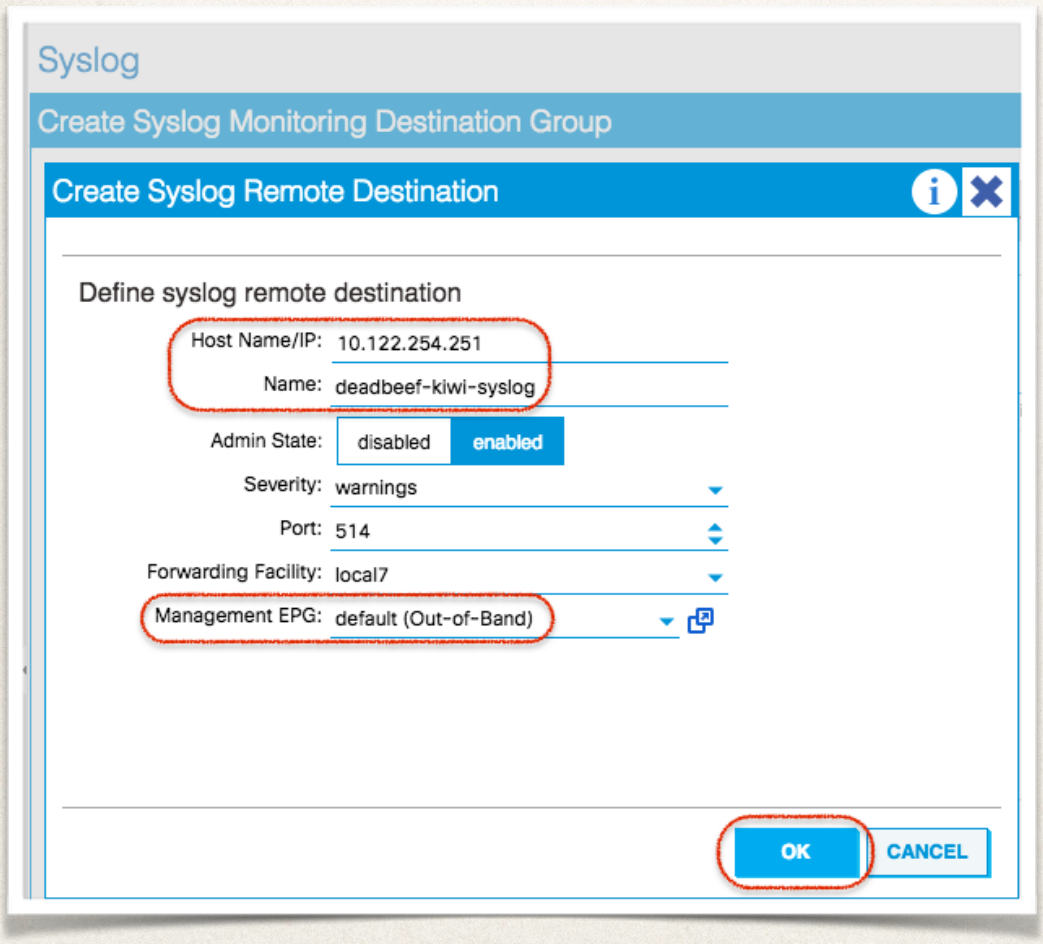
# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

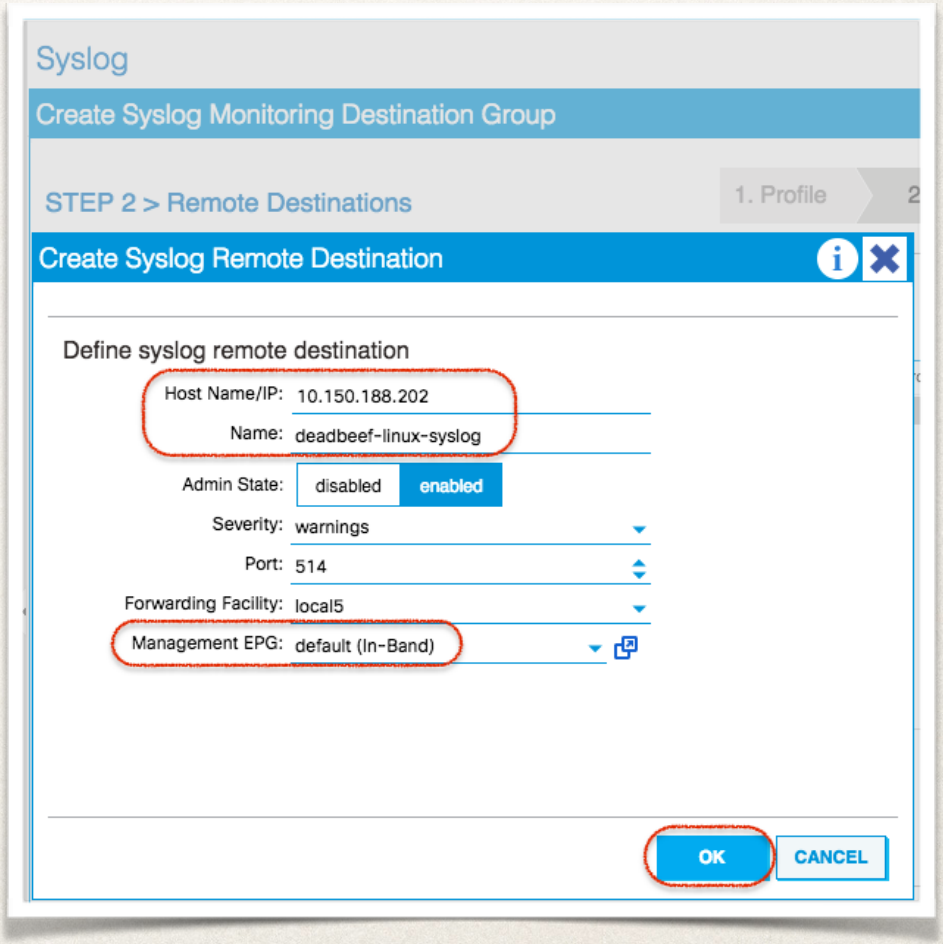
6. In the **CREATE SYSLOG REMOTE DESTINATION** configuration panel, perform the following actions:
   - Add Host Name/IP (ip address of Syslog Server)
   - Add Name (Name of Syslog Server)
   - Select Admin State (enabled)
   - Select Severity (warnings)
   - Select Management EPG (default - (Out-of-Band))
   - Click **OK**

   *Note: Repeat the "Create Syslog Remote Destination" tasks to add additional Syslog Servers for the ACI Fabric.*

# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

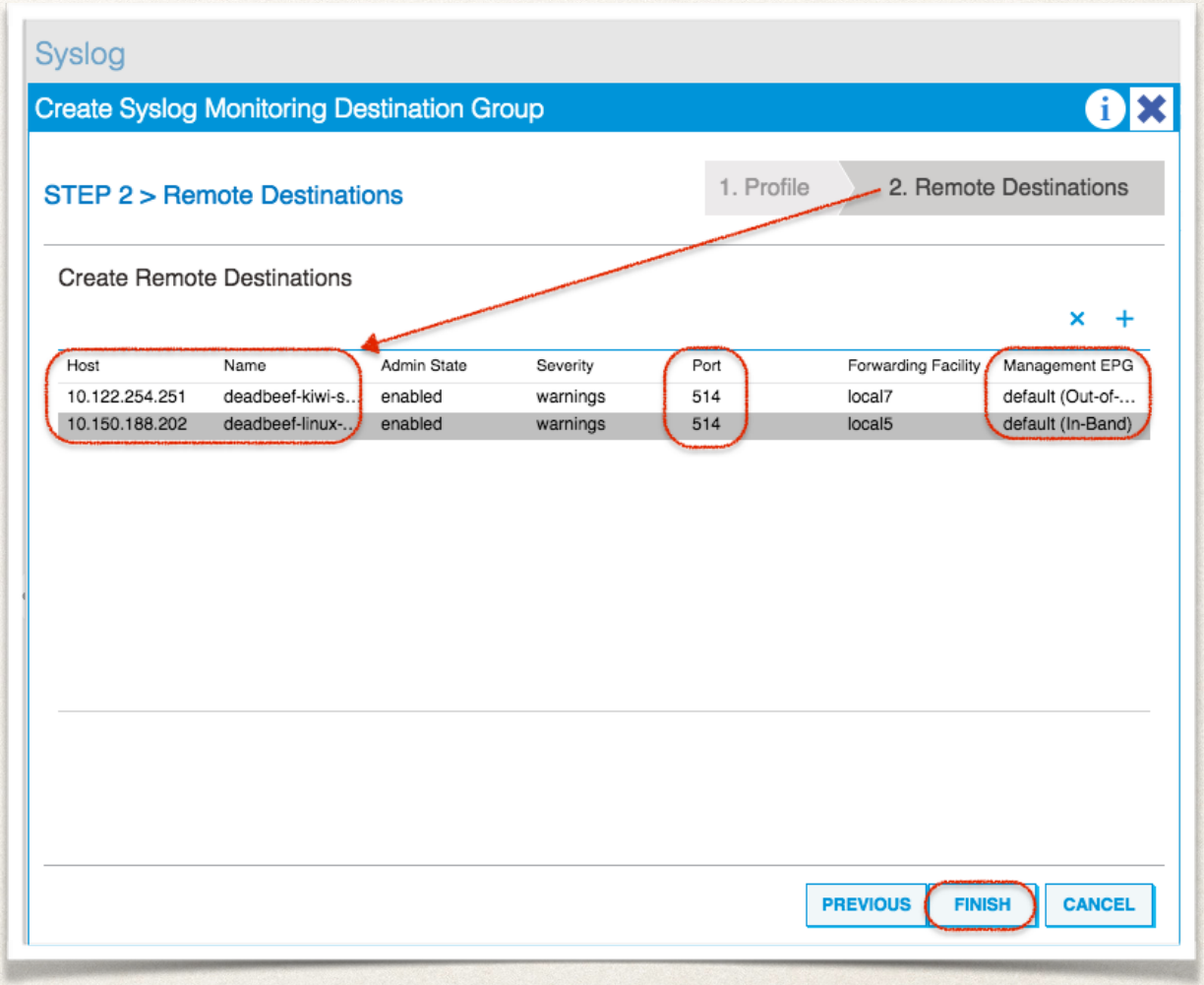
6. For example:



# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

7. The **SYSLOG REMOTE DESTINATION** configuration panel, lists all of the Syslog Servers that you have created. Once you have added all of your Remote Destinations, verify the information and Click on **FINISH** to complete creating the Syslog Monitoring Destination Group.



# TASK 2: (cont.)
# Configure a SYSLOG Policy for the ACI Fabric

8. Rest API example for TASK1:

   method: POST
   URL: https://a.b.c.d/api/node/mo/uni/fabric/slgroup-deadbeef-syslogGrp.json

   PAYLOAD BODY:

{"syslogGroup":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp","name":"deadbeef-syslogGrp","descr":"Group of Syslog Servers for the deadbeef network","rn":"slgroup-deadbeef-syslogGrp","status":"created"},"children":[{"syslogConsole":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp/console","rn":"console","status":"created"},"children":[]}},{"syslogFile":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp/file","rn":"file","status":"created"},"children":[]}},{"syslogProf":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp/prof","rn":"prof","status":"created"},"children":[]}},{"syslogRemoteDest":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.122.254.251","host":"10.122.254.251","name":"deadbeef-kiwi-syslog","rn":"rdst-10.122.254.251","status":"created"},"children":[{"fileRsARemoteHostToEpg":{"attributes":{"tDn":"uni/tn-mgmt/mgmtp-default/oob-default","status":"created"},"children":[]}}]}},{"syslogRemoteDest":{"attributes":{"dn":"uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.150.188.202","host":"10.150.188.202","name":"deadbeef-linux-syslog","forwardingFacility":"local5","rn":"rdst-10.150.188.202","status":"created"},"children":[{"fileRsARemoteHostToEpg":{"attributes":{"tDn":"uni/tn-mgmt/mgmtp-default/inb-default","status":"created"},"children":[]}}]}}]}}

   *Where "a.b.c.d" is an IP Address of one of the APICs in the ACI fabric cluster.*

# TASK 3:
## Configure the ACI Fabric nodes to send SYSLOG Messages

- After you have created the ACI Fabric's SYSLOG Monitoring Destination Group with SYSLOG Remote Destinations, you will need to configure Fabric "Monitoring Sources" to use this SYSLOG Monitoring Destination Group. There are 3 main Monitoring Sources that can be configured. I give examples of configuring each of the monitoring sources. In later ACI firmware releases, you can create a monitoring source in the Tenant scope.
- "Which monitoring sources do I need to configure?" "Do I need to configure all 3 monitoring sources?" are common questions that we get from customers. Take a look at the online documentation at: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals/b_ACI-Fundamentals_chapter_01110.html

# TASK 3: (cont.)
# Configure the ACI Fabric nodes to send SYSLOG Messages

## Configuring Monitoring Policies

Administrators can create monitoring policies with the following four broad scopes:

- **Fabric Wide:** includes both fabric and access objects
- **Access (also known as infrastructure):** access ports, FEX, VM controllers, and so on
- **Fabric:** fabric ports, cards, chassis, fans, and so on
- **Tenant:** EPGs , application profiles, services, and so on

**The APIC includes the following four classes of default monitoring policies:**

**monCommonPol (uni/fabric/moncommon):** ***applies to both fabric and access infrastructure hierarchies***

**monFabricPol (uni/fabric/monfab-default):** applies to **fabric** hierarchies

**monInfraPol (uni/infra/monifra-default):** applies to the **access infrastructure** hierarchy

**monEPGPol (uni/tn-common/monepg-default):** applies to **tenant** hierarchies

In each of the four classes of monitoring policies, *the default policy can be overridden by a specific policy*. For example, a monitoring policy applied to the deadbeef tenant (tn-deadbeef) would override the default one for the deadbeef tenant while other tenants would still be monitored by the default policy.

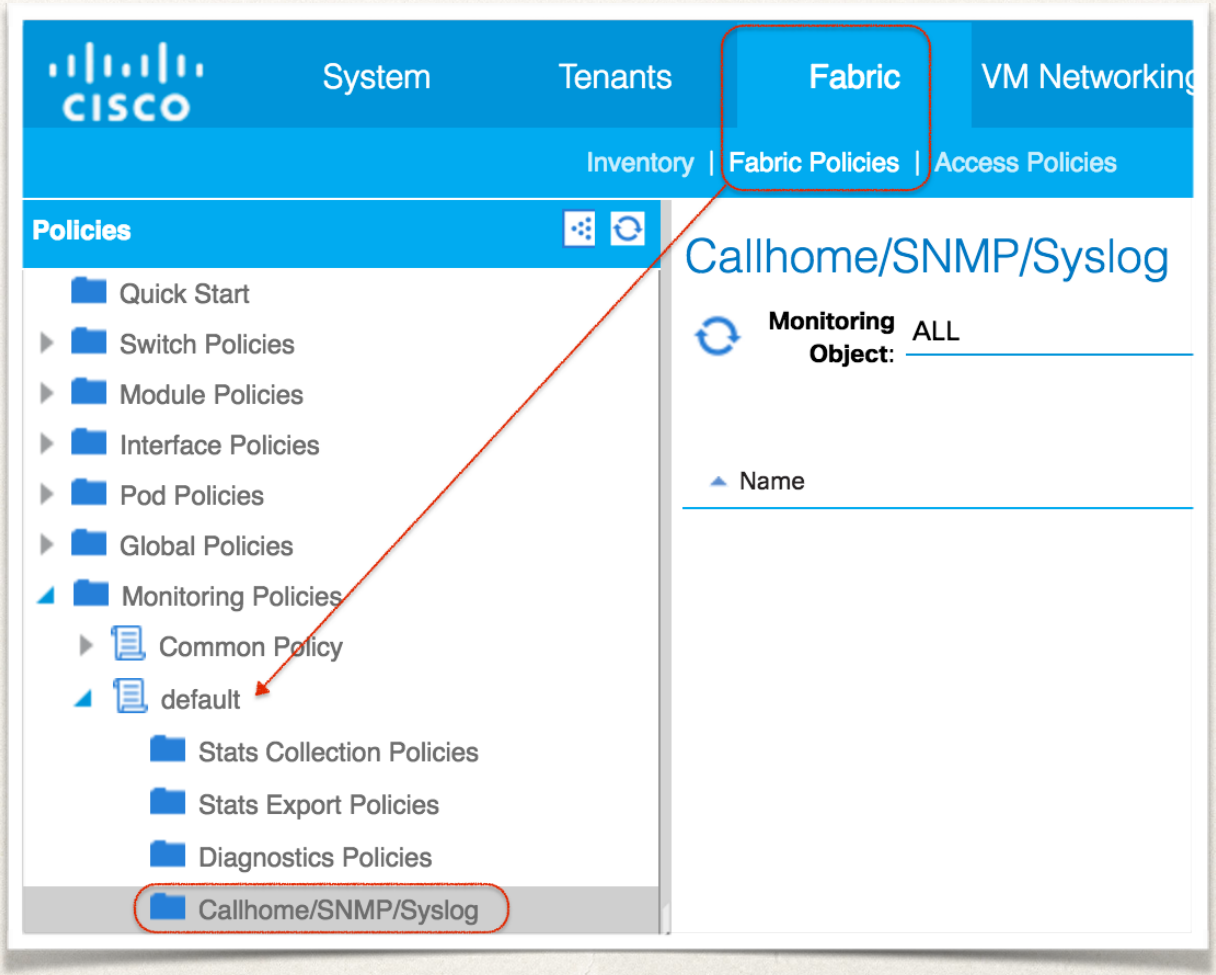
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- After you have created the ACI Fabric's SYSLOG Monitoring Destination Group with SYSLOG Remote Destinations, you will need to configure **Fabric "Monitoring Sources"** to use this SYSLOG Monitoring Destination Group.

- Configuration Steps:
  1. Access the APIC Admin GUI.
  2. Select **FABRIC -> FABRIC POLICIES.**
  3. In the Policies navigation panel on the left, select and expand the **MONITORING POLICIES**.
     - Expand **DEFAULT** and **Select** "**CALLHOME/SNMP/SYSLOG**".
     - In the "Callhome/SNMP/Syslog" configuration panel, **Select SYSLOG** as the "Source Type" and Click on the " + " sign to **CREATE SYSLOG SOURCE**.
     - In the "Create SYSLOG Source" configuration panel, perform the following actions:
       - **Enter Source Name** (deadbeef-syslogSrc)
       - Change **MIN SEVERITY** to INFORMATION
       - Select the "**CHECK ALL**" button to include: *Audit logs, Events, Faults, and Session logs.*
       - **Select** the **SYSLOG Monitoring Destination Group** that was created in a previous task (deadbeef-syslogGrp)
       - **Click Submit**

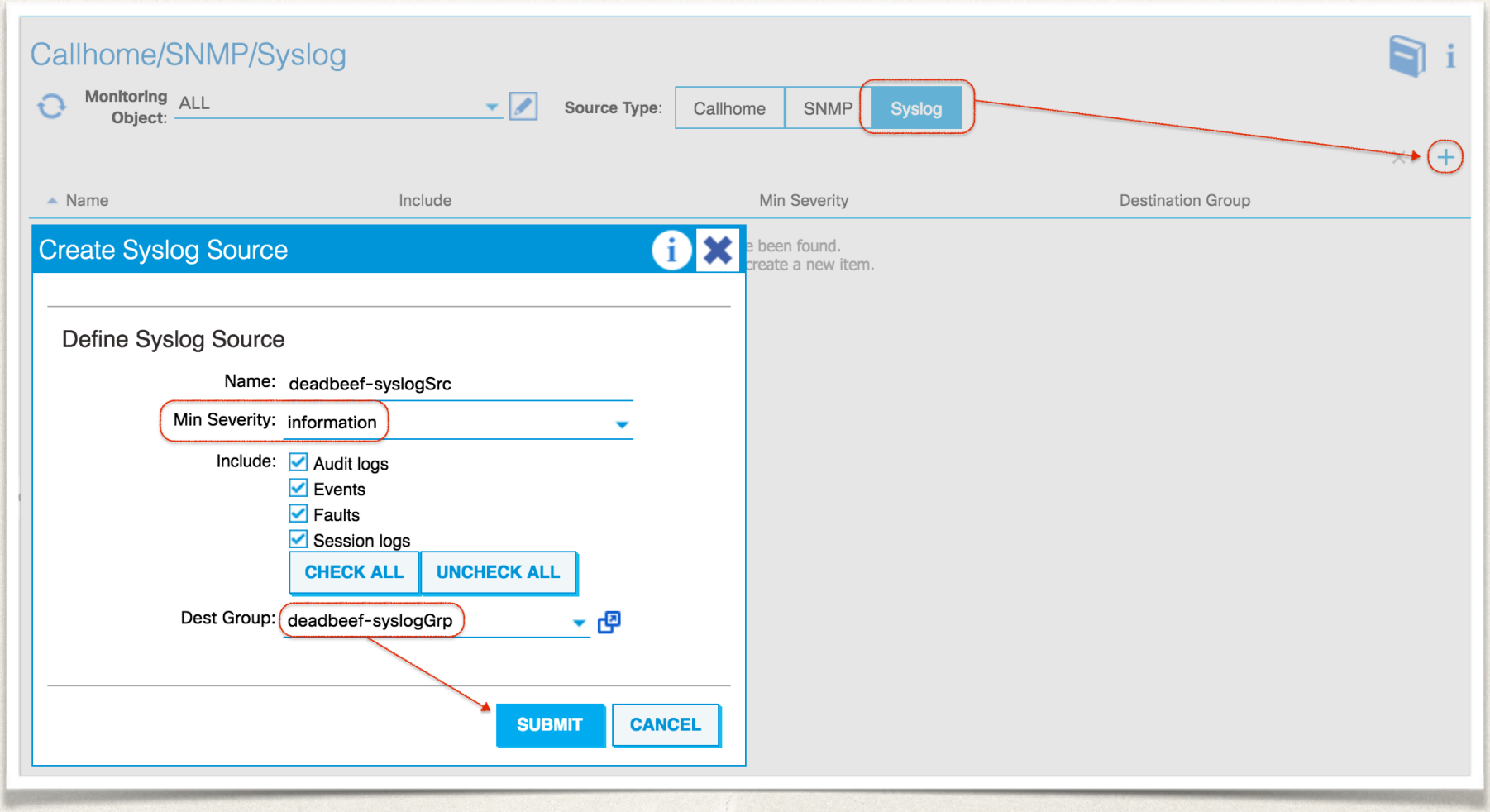
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - default (Callhome/SNMP/Syslog))



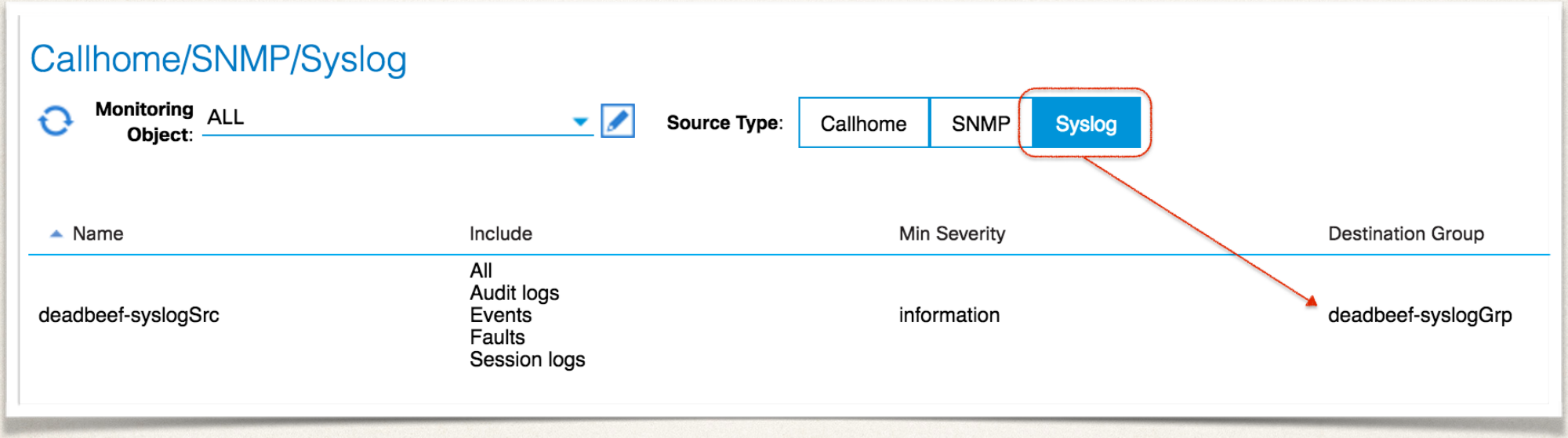
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - default (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - default (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Example of POST from the API Inspector:

**Fabric Policies - default (Callhome/SNMP/Syslog)**

method: POST
url:
https://10.122.254.211/api/node/mo/uni/fabric/monfab-default/slsrc-deadbeef-syslogSrc.json

payload
{"syslogSrc":{"attributes":{"dn":"uni/fabric/monfab-default/slsrc-deadbeef-syslogSrc","name":"deadbeef-syslogSrc","minSev":"information","incl":"audit,events,faults,session","rn":"slsrc-deadbeef-syslogSrc","status":"created"},"children":[{"syslogRsDestGroup":{"attributes":{"tDn":"uni/fabric/slgroup-deadbeef-syslogGrp","status":"created"},"children":[]}}]}}

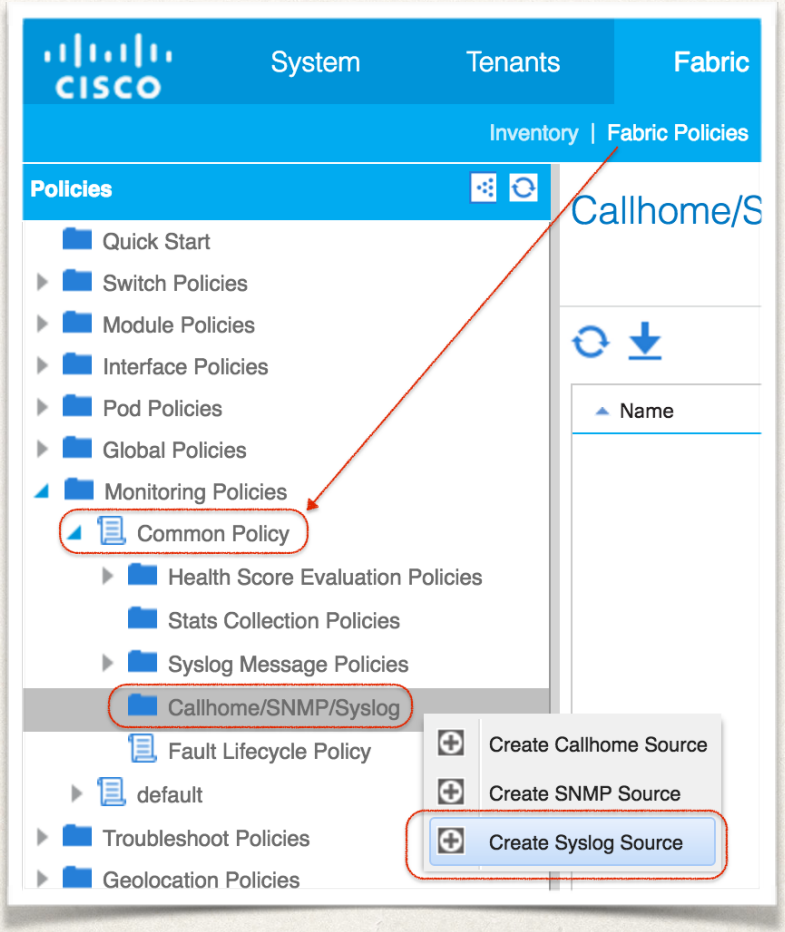
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- After you have created the ACI Fabric's SYSLOG Source in the Fabric Policies "Monitoring Sources" for Fabric Policies - DEFAULT, configure the SYSLOG Source in Fabric Policies - **COMMON POLICY.**

- Configuration Steps:
  1. Access the APIC Admin GUI.
  2. Select **FABRIC -> FABRIC POLICIES.**
  3. In the Policies navigation panel on the left, select and expand the **MONITORING POLICIES**.
     - Expand **COMMON** and **Select** "**CALLHOME/SNMP/SYSLOG**".
     - Right Click and select the " + " sign to **CREATE SYSLOG SOURCE**.
     - In the "Create SYSLOG Source" configuration panel, perform the following actions:
       - **Enter Source Name** (deadbeef-syslogSrc)
       - Change **MIN SEVERITY** to INFORMATION
       - Select the "**CHECK ALL**" button to include: *Audit logs, Events, Faults, and Session logs.*
       - **Select** the **SYSLOG Monitoring Destination Group** that was created in a previous task (deadbeef-syslogGrp)
       - **Click Submit**

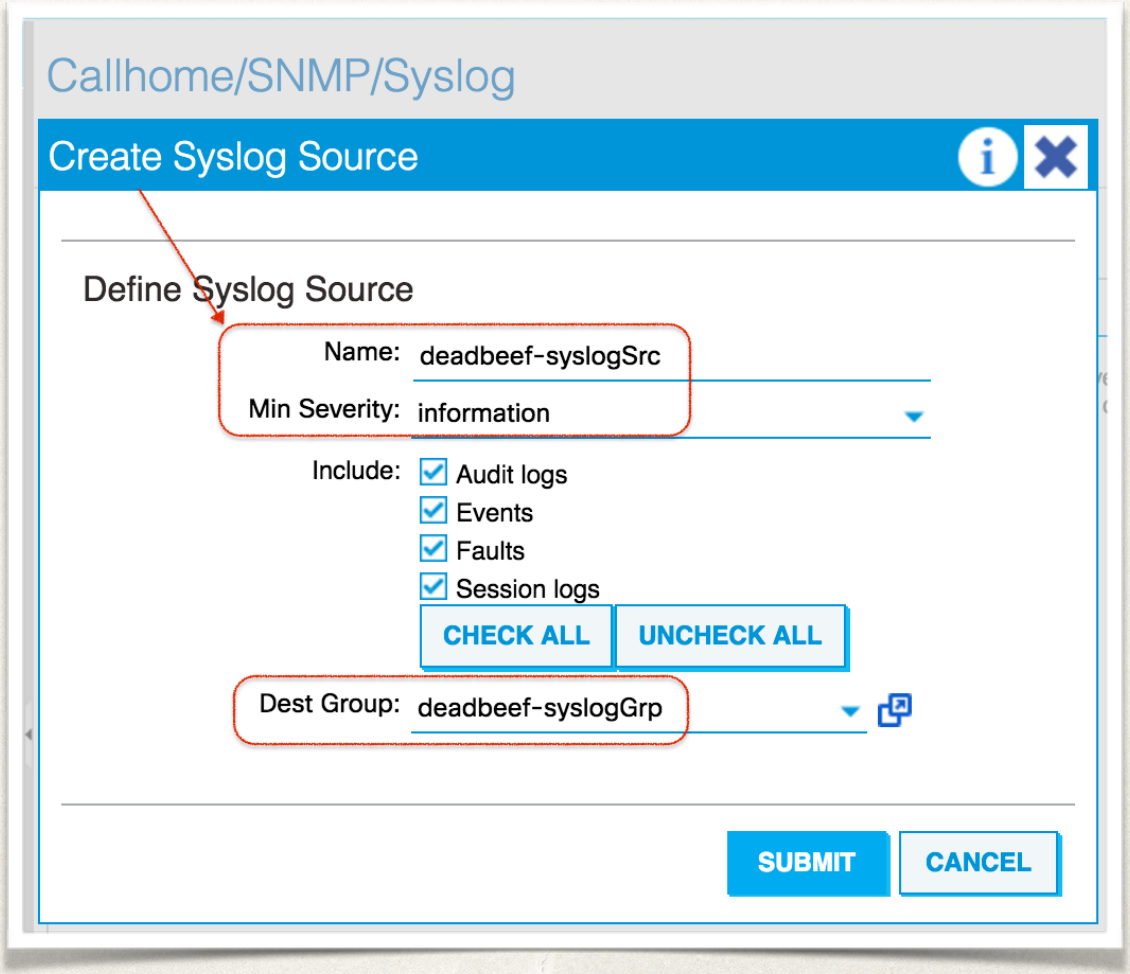
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - common (Callhome/SNMP/Syslog))



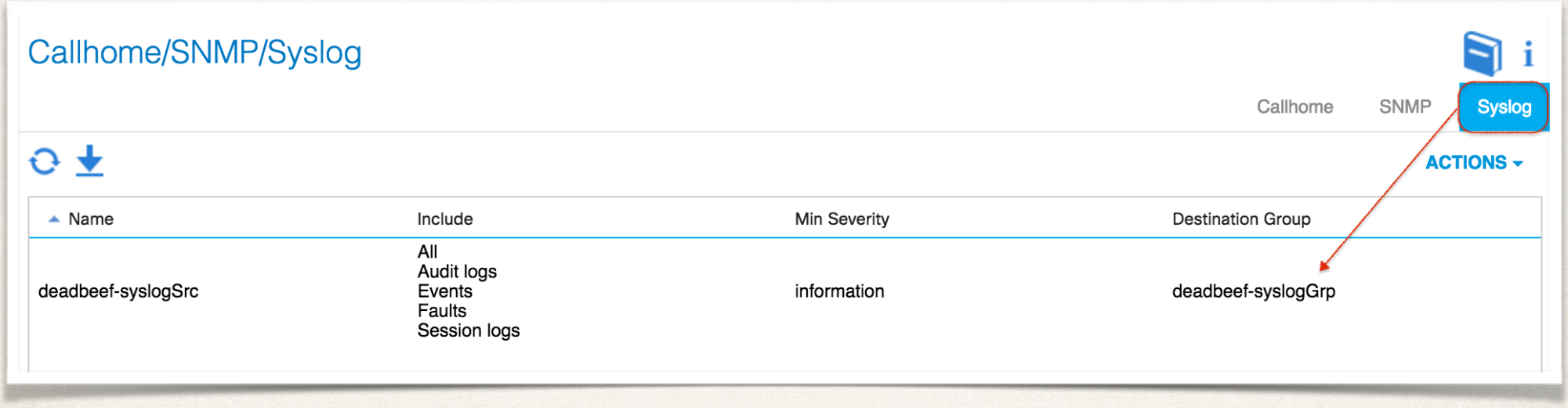
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - common (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - common (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Example of POST from the API Inspector:

**Fabric Policies - common (Callhome/SNMP/Syslog)**

method: POST
url:
https://10.122.254.211/api/node/mo/uni/fabric/moncommon/slsrc-deadbeef-syslogSrc.json

payload{"syslogSrc":{"attributes":{"dn":"uni/fabric/moncommon/slsrc-deadbeef-syslogSrc","name":"deadbeef-syslogSrc","minSev":"information","incl":"audit,events,faults,session","rn":"slsrc-deadbeef-syslogSrc","status":"created"},"children":[{"syslogRsDestGroup":{"attributes":{"tDn":"uni/fabric/slgroup-deadbeef-syslogGrp","status":"created"},"children":[]}}]}}

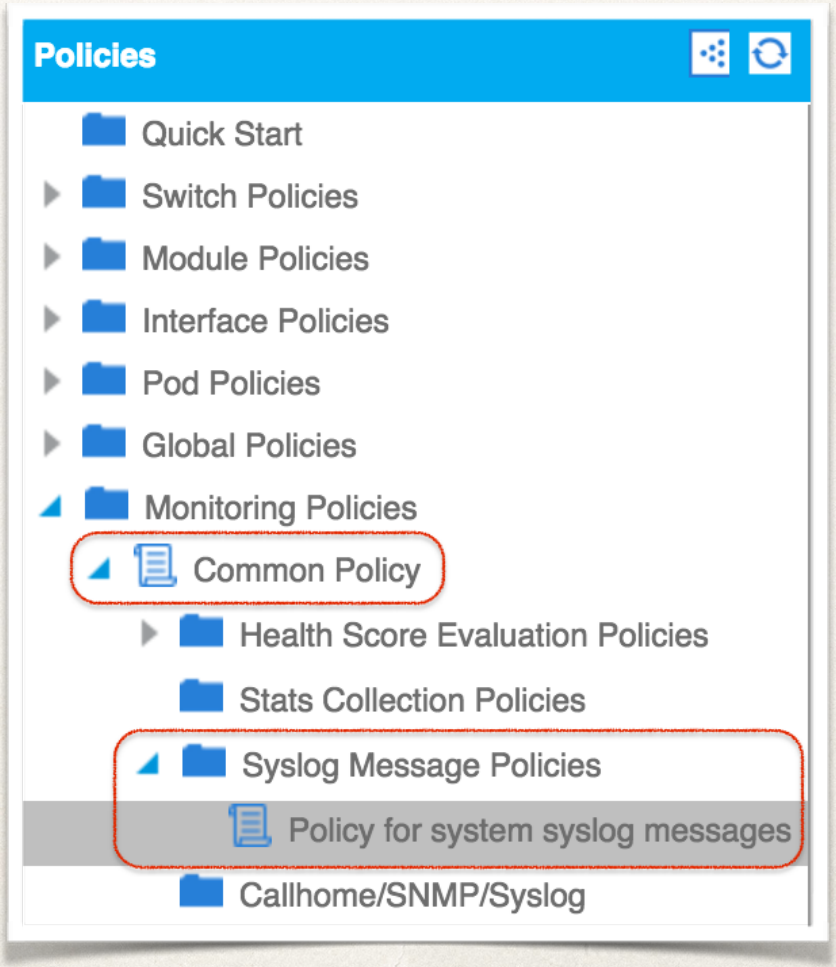
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- After you have created the ACI Fabric's SYSLOG Source in the Fabric Policies "Monitoring Sources" for Fabric Policies - COMMON, configure the **SYSLOG SYSTEM MESSAGES POLICY** in the **COMMON POLICY.** The task for this step is to configure th "**Facility Filter**" for the "**default**" facility. Changing the Severity to "**information**" will record *%ACLLOG-5-ACLLOG_PKTLOG* messages in Syslog.

- Configuration Steps:
  1. Access the APIC Admin GUI.
  2. Select **FABRIC -> FABRIC POLICIES.**
  3. In the Policies navigation panel on the left, select and expand the **MONITORING POLICIES**.
     - Expand **COMMON**
     - Expand **SYSTEM MESSAGE POLICIES**
     - **Select "POLICY FOR SYSTEM SYSLOG MESSAGES**".
     - In the "System Messages Policy" configuration panel, perform the following actions:
       - Select the "**default**" facility
       - Change **SEVERITY** to INFORMATION
       - **Click UPDATE**

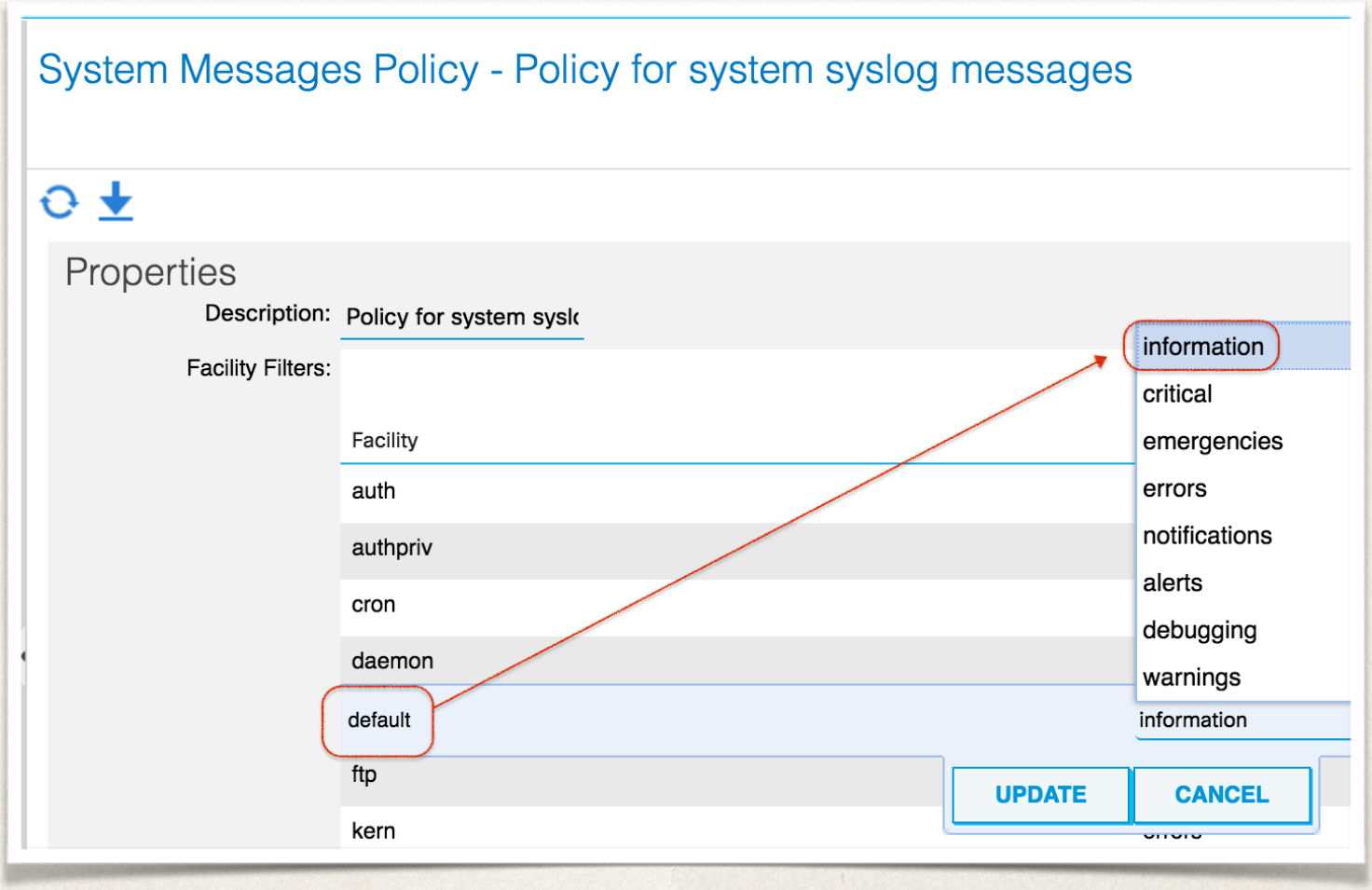
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - common **(SYSTEM MESSAGE POLICY)**)



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Fabric Policies - common **(SYSTEM MESSAGE POLICY)**)



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Example of POST from the API Inspector:

**Fabric Policies - common (SYSTEM MESSAGE POLICY)**

method: POST
url:
https://10.122.254.211/api/node/mo/uni/fabric/moncommon/sysmsgp/ff-default.json

payload{"syslogFacilityFilter":{"attributes":{"dn":"uni/fabric/moncommon/sysmsgp/ff-default","minSev":"information"},"children":[]}}

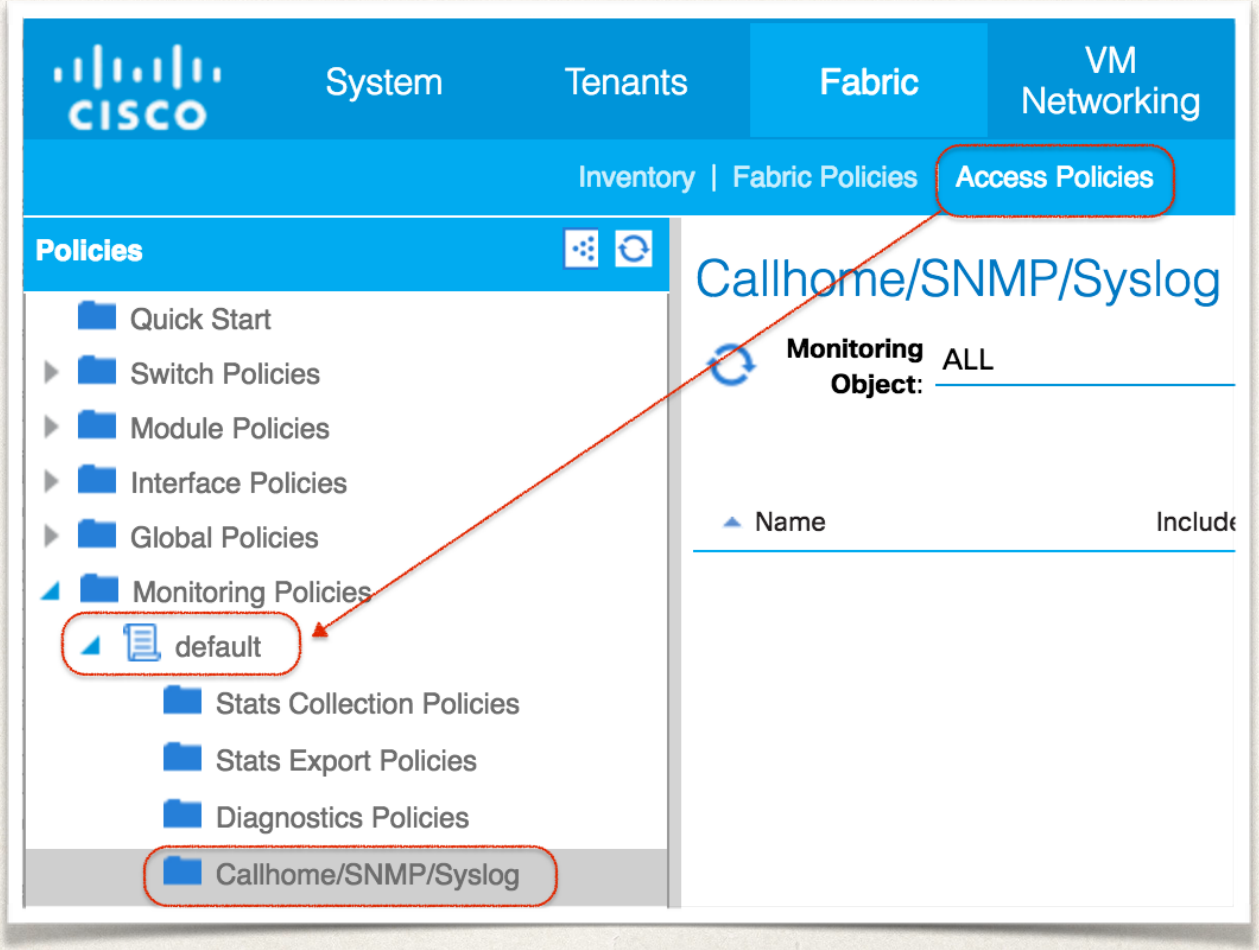
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- After you have created the ACI Fabric's SYSLOG Source in the Fabric Policies "Monitoring Sources" for *Fabric Policies - DEFAULT & COMMON*, configure the SYSLOG Source in **ACCESS Policies - DEFAULT POLICY.**

- <u>Configuration Steps:</u>
  1. Access the APIC Admin GUI.
  2. Select **FABRIC -> ACCESS POLICIES.**
  3. In the Policies navigation panel on the left, select and expand the **MONITORING POLICIES**.
     - Expand **DEFAULT** and **Select** "**CALLHOME/SNMP/SYSLOG**".
     - In the "Callhome/SNMP/Syslog" configuration panel, **Select SYSLOG** as the "Source Type" and Click on the " + " sign to **CREATE SYSLOG SOURCE**.
     - In the "Create SYSLOG Source" configuration panel, perform the following actions:
       - **Enter Source Name** (deadbeef-syslogSrc)
       - Change **MIN SEVERITY** to INFORMATION
       - Select the "**CHECK ALL**" button to include: *Audit logs, Events, Faults, and Session logs.*
       - **Select** the **SYSLOG Monitoring Destination Group** that was created in a previous task (deadbeef-syslogGrp)
       - **Click Submit**

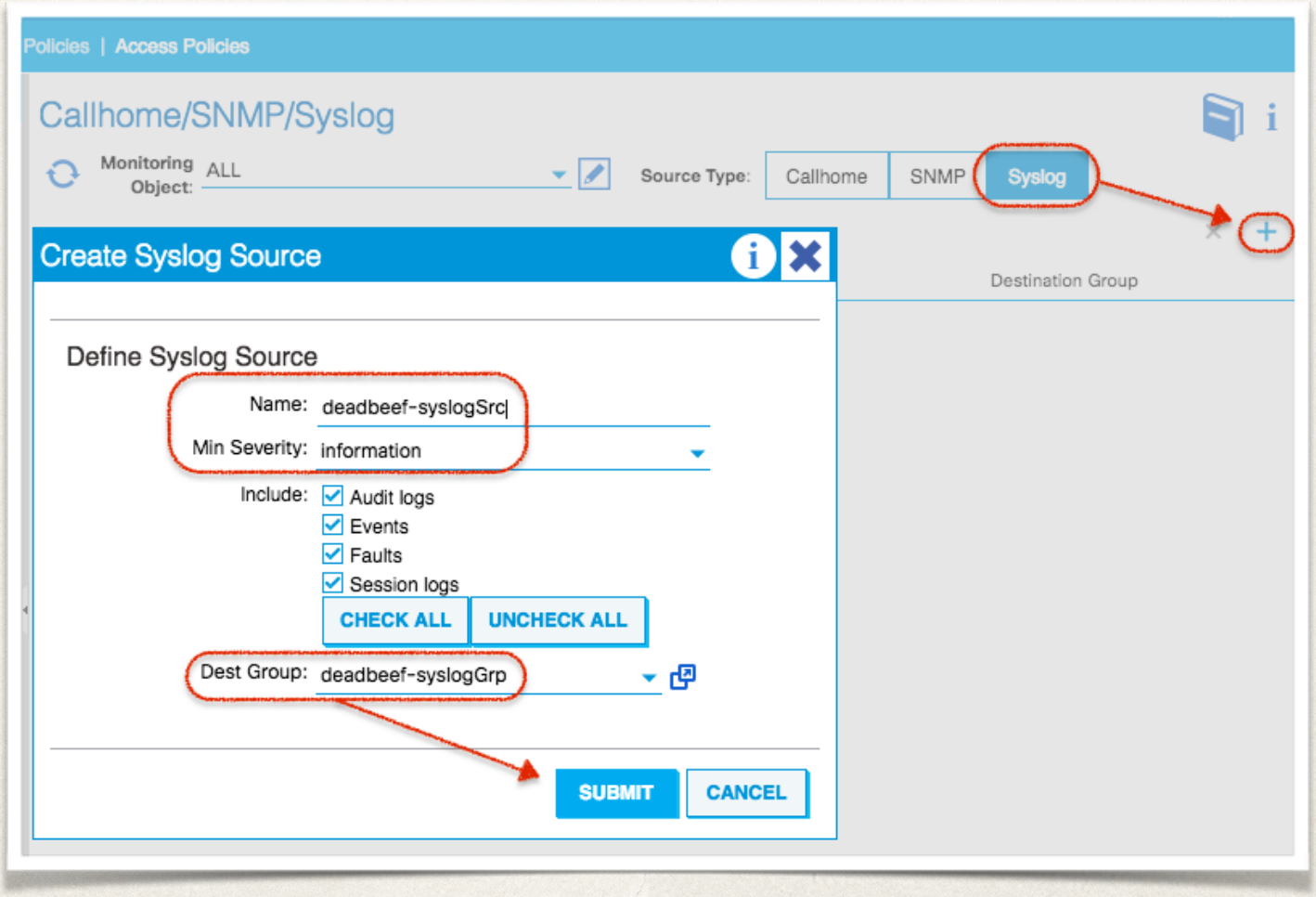
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Access Policies - default (Callhome/SNMP/Syslog))



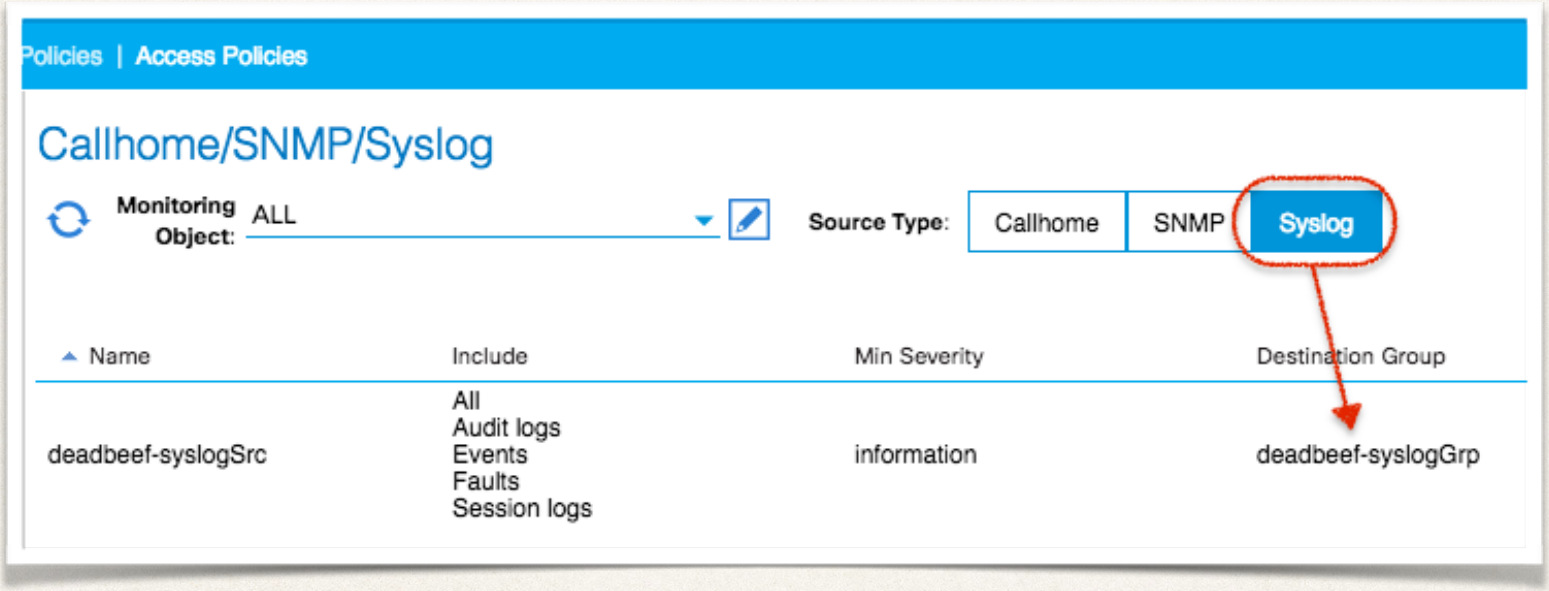
# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Access Policies - default (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Sample Screenshots: (Access Policies - default (Callhome/SNMP/Syslog))



# TASK 3: (cont.)
## Configure the ACI Fabric nodes to send SYSLOG Messages

- Example of POST from the API Inspector:

**Access Policies - default (Callhome/SNMP/Syslog)**

method: POST
url:
https://10.122.254.211/api/node/mo/uni/infra/moninfra-default/slsrc-deadbeef-syslogSrc.json

payload {"syslogSrc":{"attributes":{"dn":"uni/infra/moninfra-default/slsrc-deadbeef-syslogSrc","incl":"audit,events,faults,session","minSev":"information","name":"deadbeef-syslogSrc","rn":"slsrc-deadbeef-syslogSrc","status":"created"},"children":[{"syslogRsDestGroup":{"attributes":{"tDn":"uni/fabric/slgroup-deadbeef-syslogGrp","status":"created"},"children":[]}}]}}

# TASK 4:
# Verify the Syslog configuration (APIC)

- Steps to Verify Configuration (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1
  2. Use the "**show running-config syslog**" & "**show running-config logging server-group**" commands to verify Syslog.

CLI configuration Example:

```
apic1# show running-config syslog
# Command: show running-config syslog
 syslog deadbeef-moninfra-syslog
   exit
 syslog deadbeef-moncommon-syslog
   logging audit
   logging event
   logging fault
   logging session
   logging severity information
   logging server-group deadbeef-syslogGrp
   exit
 syslog deadbeef-monfab-syslog
   exit

apic1# show running-config logging server-group deadbeef-syslogGrp
# Command: show running-config logging server-group deadbeef-syslogGrp
 logging server-group deadbeef-syslogGrp
   description "Group of Syslog Servers for the deadbeef network"
   logfile
   console
   server 10.117.67.30 severity information facility local5 mgmtepg inb port 514
   server 10.122.254.251 severity information facility local7 mgmtepg oob port 514
   exit
```

# TASK 4: (cont.)
# Verify the Syslog configuration (Leaf\Spine)

- Steps to Verify Configuration (using the LEAF or SPINE CLI):
  1. SSH or use CONSOLE to access LEAF or SPINE Nodes
  2. Use the following commands to verify Syslog configuration:
     - cat /mit/uni/fabric/slgroup-**deadbeef-syslogGrp**/summary
     - ls /mit/uni/fabric/slgroup-**deadbeef-syslogGrp**/rdst* | grep "rdst"
     - cat /mit/uni/fabric/monfab-default/slsrc-**deadbeef-monfab-syslog**/summary
     - cat /mit/uni/fabric/moncommon/slsrc-**deadbeef-moncommon-syslog**/summary
     - cat /mit/uni/infra/moninfra-default/slsrc-**deadbeef-moninfra-syslog**/summary

*Note: Use the above commands as examples for your configuration. You will need to replace the "Names" of the Syslog Group and Syslog Sources (**in BLUE**) with your policy names.*

# TASK 4: (cont.)
# Verify the Syslog configuration (Leaf\Spine)

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/fabric/slgroup-deadbeef-syslogGrp/summary
# Syslog Monitoring Destination Group
name              : deadbeef-syslogGrp
childAction       :
descr             : Group of Syslog Servers for the deadbeef network
dn                : uni/fabric/slgroup-deadbeef-syslogGrp
format            : aci
lcOwn             : policy
modTs             : 2016-08-17T03:01:14.377+00:00
monPolDn          : uni/fabric/monfab-default
remoteDestCount   : 2
rn                : slgroup-deadbeef-syslogGrp
status            :
uid               : 15374

fab2-p1-leaf1# ls /mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst* | grep "rdst"
/mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.117.67.30:
/mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.122.254.251:
```

# TASK 4: (cont.)
# Verify the Syslog configuration (Leaf\Spine)

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog/summary
# Syslog Source
name         : deadbeef-monfab-syslog
childAction  :
descr        :
dn           : uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog
incl         : all,audit,events,faults,session
lcOwn        : policy
minSev       : information
modTs        : 2016-08-17T02:48:29.598+00:00
monPolDn     : uni/fabric/monfab-default
rn           : slsrc-deadbeef-monfab-syslog
status       :
uid          : 15374

fab2-p1-leaf1# cat /mit/uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog/summary
# Syslog Source
name         : deadbeef-moncommon-syslog
childAction  :
descr        :
dn           : uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog
incl         : all,audit,events,faults,session
lcOwn        : policy
minSev       : information
modTs        : 2016-08-17T02:48:18.629+00:00
monPolDn     : uni/fabric/moncommon
rn           : slsrc-deadbeef-moncommon-syslog
status       :
uid          : 15374
```

# TASK 4: (cont.)
# Verify the Syslog configuration (Leaf\Spine)

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/infra/moninfra-default/slsrc-deadbeef-moninfra-
syslog/summary
# Syslog Source
name          : deadbeef-moninfra-syslog
childAction   :
descr         :
dn            : uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-08-17T02:48:42.692+00:00
monPolDn      : uni/infra/moninfra-default
rn            : slsrc-deadbeef-moninfra-syslog
status        :
uid           : 15374
```

*Note: Repeat the same CLI commands to verify Syslog configuration on all Leaf & Spine Nodes in the ACI Fabric.*

# TASK 5:
# Test the Syslog configuration using the CLI Syslog Test Feature

- Test Syslog Configuration Steps (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1.
  2. Use the "**logit**" CLI command to verify Syslog configuration.
     *Note: The Syslog "**logit**" test comand will be available in ACI versions from Congo Maintance Releases and later.*
  3. Perform a "**logit**" test for each configured remote destination. Use Node 1 (APIC1) for each test.

**Command Syntax:**

# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message>
# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message> node <id>

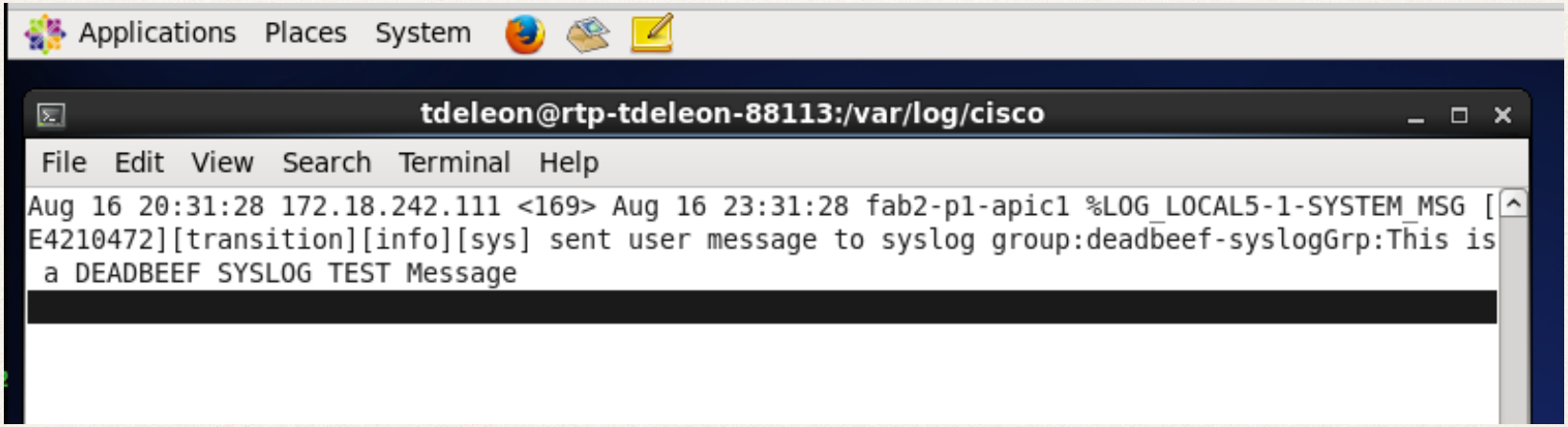
*Remember to run a test for each configured Syslog remote destination.*

# TASK 5:
# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **INB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.117.67.30 "This is a DEADBEEF SYSLOG TEST Message"** node **1**

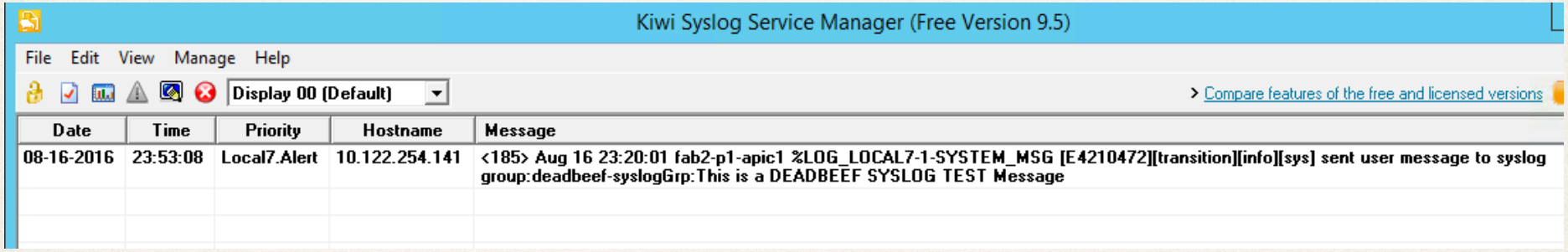


# TASK 5:
# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **OOB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.122.254.251 "This is a DEADBEEF SYSLOG TEST Message"** node **1**




**You can download a free version fo the Kiwi Syslog Server for Windows at:**
http://www.kiwisyslog.com

# Troubleshooting ACI SYSLOG Configuration

This section will provide an overview on generic troubleshooting SYSLOG policies in the ACI Fabric. Once SYSLOG policies are configured for sending SYSLOG messages, verify that the configuration is pushed to the LEAF\SPINE\APIC nodes. Use the available CLI commands to verify configuration is enabled and applied. If needed, use of external tools and apps may be necessary.

# Verify ACI SYSLOG Configuration - “CLI Commands”

After completing the configuration of SYSLOG policies, verify configuration on Leaf\Spine\APIC Nodes. *Note: iNXOS CLI support for the APIC controllers was added in ACI version 1.2(xx) or later so the APIC iNXOS CLI related commands only pertains to fabrics running ACI version 1.2(xx) or later.*

# Verify ACI SYSLOG Configuration

## "show commands"

- **After completing the configuration of SYSLOG policies, verify configuration on Leaf\Spine\APIC Nodes.** *Note: iNXOS CLI support for the APIC controllers was added in ACI version 1.2(xx) or later so the APIC iNXOS CLI related commands only pertains to fabrics running ACI version 1.2(xx) or later.*

1. SSH to a Fabric APIC. Use the "*attach node-name*" command to connect to the desired Leaf\Spine Nodes.
2. Use the following ACI CLI **SHOW** commands to verify the configuration on the Leaf\Spine\APIC nodes:

| APIC CLI COMMANDS |
| --- |
| show running-config logging |
| show running-config logging server-group <syslog destination group> |
| show running-config syslog |
| show running-config syslog common |

# Verify the Syslog configuration (APIC)

## "show commands"

- Steps to Verify Configuration (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1
  2. Use the "**show running-config syslog**" & "**show running-config logging server-group**" commands to verify Syslog.

CLI configuration Example:

```
apic1# show running-config syslog
# Command: show running-config syslog
 syslog deadbeef-moninfra-syslog
   exit
 syslog deadbeef-moncommon-syslog
   logging audit
   logging event
   logging fault
   logging session
   logging severity information
   logging server-group deadbeef-syslogGrp
   exit
 syslog deadbeef-monfab-syslog
   exit

apic1# show running-config logging server-group deadbeef-syslogGrp
# Command: show running-config logging server-group deadbeef-syslogGrp
 logging server-group deadbeef-syslogGrp
   description "Group of Syslog Servers for the deadbeef network"
   logfile
   console
   server 10.117.67.30 severity information facility local5 mgmtepg inb port 514
   server 10.122.254.251 severity information facility local7 mgmtepg oob port 514
   exit
```

# Verify ACI SYSLOG Configuration (cont.)

## "show commands"

- **After completing the configuration of SYSLOG policies, verify configuration on Leaf\Spine\APIC Nodes.** *Note: iNXOS CLI support for the APIC controllers was added in ACI version 1.2(xx) or later so the APIC iNXOS CLI related commands only pertains to fabrics running ACI version 1.2(xx) or later.*

1. SSH to a Fabric APIC. Use the "*attach node-name*" command to connect to the desired Leaf\Spine Nodes.
2. Use the following ACI CLI commands to verify the configuration on the Leaf\Spine Nodes:

| LEAF\SPINE CLI COMMANDS |
| --- |
| cat /mit/uni/fabric/slgroup-**syslogGroup-NAME**/summary |
| ls /mit/uni/fabric/slgroup-**syslogGroup-NAME**/rdst* \| grep "rdst" |
| cat /mit/uni/fabric/monfab-default/slsrc-**syslogSource-NAME**/summary |
| cat /mit/uni/fabric/moncommon/slsrc-**syslogSource-NAME**/summary |
| cat /mit/uni/infra/moninfra-default/slsrc-**syslogSource-NAME**/summary |

# Verify the Syslog configuration (Leaf\Spine)

## "CLI commands"

- Steps to Verify Configuration (using the LEAF or SPINE CLI):
  1. SSH or use CONSOLE to access LEAF or SPINE Nodes
  2. Use the following commands to verify Syslog configuration:
     - cat /mit/uni/fabric/slgroup-**deadbeef-syslogGrp**/summary
     - ls /mit/uni/fabric/slgroup-**deadbeef-syslogGrp**/rdst* | grep "rdst"
     - cat /mit/uni/fabric/monfab-default/slsrc-**deadbeef-monfab-syslog**/summary
     - cat /mit/uni/fabric/moncommon/slsrc-**deadbeef-moncommon-syslog**/summary
     - cat /mit/uni/infra/moninfra-default/slsrc-**deadbeef-moninfra-syslog**/summary

*Note: Use the above commands as examples for your configuration. You will need to replace the "Names" of the Syslog Group and Syslog Sources (**in BLUE**) with your policy names.*

# Verify the Syslog configuration (Leaf\Spine)

## "CLI commands"

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/fabric/slgroup-deadbeef-syslogGrp/summary
# Syslog Monitoring Destination Group
name              : deadbeef-syslogGrp
childAction       :
descr             : Group of Syslog Servers for the deadbeef network
dn                : uni/fabric/slgroup-deadbeef-syslogGrp
format            : aci
lcOwn             : policy
modTs             : 2016-08-17T03:01:14.377+00:00
monPolDn          : uni/fabric/monfab-default
remoteDestCount   : 2
rn                : slgroup-deadbeef-syslogGrp
status            :
uid               : 15374

fab2-p1-leaf1# ls /mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst* | grep "rdst"
/mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.117.67.30:
/mit/uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.122.254.251:
```

# Verify the Syslog configuration (Leaf\Spine)

## "CLI commands"

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog/summary
# Syslog Source
name          : deadbeef-monfab-syslog
childAction   :
descr         :
dn            : uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-08-17T02:48:29.598+00:00
monPolDn      : uni/fabric/monfab-default
rn            : slsrc-deadbeef-monfab-syslog
status        :
uid           : 15374

fab2-p1-leaf1# cat /mit/uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog/summary
# Syslog Source
name          : deadbeef-moncommon-syslog
childAction   :
descr         :
dn            : uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-08-17T02:48:18.629+00:00
monPolDn      : uni/fabric/moncommon
rn            : slsrc-deadbeef-moncommon-syslog
status        :
uid           : 15374
```

# Verify the Syslog configuration (Leaf\Spine)

## "CLI commands"

CLI configuration Example:

```
fab2-p1-leaf1# cat /mit/uni/infra/moninfra-default/slsrc-deadbeef-moninfra-
syslog/summary
# Syslog Source
name          : deadbeef-moninfra-syslog
childAction   :
descr         :
dn            : uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-08-17T02:48:42.692+00:00
monPolDn      : uni/infra/moninfra-default
rn            : slsrc-deadbeef-moninfra-syslog
status        :
uid           : 15374
```

*Note: Repeat the same CLI commands to verify Syslog configuration on all Leaf & Spine Nodes in the ACI Fabric.*

# Verify ACI SYSLOG Configuration - “moquery”

Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies.
On each APIC\Leaf\Spine with SYSLOG configured, run *“moquery -c [object class]”*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

syslogGroup

```
apic1# moquery -c syslogGroup
Total Objects shown: 1

# syslog.Group
name             : deadbeef-syslogGrp
childAction      :
descr            : Group of Syslog Servers for the deadbeef network
dn               : uni/fabric/slgroup-deadbeef-syslogGrp
format           : aci
lcOwn            : local
modTs            : 2016-11-30T18:22:38.309-05:00
monPolDn         : uni/fabric/monfab-default
nameAlias        :
remoteDestCount  : 2
rn               : slgroup-deadbeef-syslogGrp
status           :
uid              : 15374
```

*Note: Repeat the "moquery -c syslogGroup" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

syslogRemoteDest

```
apic1# moquery -c syslogRemoteDest
Total Objects shown: 2

# syslog.RemoteDest
host               : 10.122.254.251
adminState         : enabled
childAction        :
descr              :
dn                 : uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.122.254.251
epgDn              : uni/tn-mgmt/mgmtp-default/oob-default
format             : aci
forwardingFacility : local7
ip                 :
lcOwn              : local
modTs              : 2016-11-14T10:37:52.819-05:00
monPolDn           : uni/fabric/monfab-default
name               : nangaparbat.cisco.com
nameAlias          :
operState          : unknown
port               : 514
rn                 : rdst-10.122.254.251
severity           : information
status             :
uid                : 15374
vrfId              : 0
vrfName            :
```

*Note: Repeat the "moquery -c syslogRemoteDest" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

**syslogRemoteDest (cont.)**

```
# syslog.RemoteDest
host               : 10.117.67.30
adminState         : enabled
childAction        :
descr              :
dn                 : uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.117.67.30
epgDn              : uni/tn-mgmt/mgmtp-default/inb-default
format             : aci
forwardingFacility : local7
ip                 :
lcOwn              : local
modTs              : 2016-11-30T18:22:38.309-05:00
monPolDn           : uni/fabric/monfab-default
name               : deadbeef-macosx-vpn30
nameAlias          :
operState          : unknown
port               : 514
rn                 : rdst-10.117.67.30
severity           : warnings
status             :
uid                : 15374
vrfId              : 0
vrfName            :
```

*Note: Repeat the "moquery -c syslogRemoteDest" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

**syslogProf**

```
apic1# moquery -c syslogProf
Total Objects shown: 1

# syslog.Prof
adminState   : enabled
childAction  :
descr        :
dn           : uni/fabric/slgroup-deadbeef-syslogGrp/prof
lcOwn        : local
modTs        : 2016-10-18T23:51:46.876-05:00
name         : syslog
nameAlias    :
port         : 514
rn           : prof
status       :
transport    : udp
uid          : 15374
```

*Note: Repeat the "moquery -c syslogProf" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

syslogRtDestGroup

```
apic1# moquery -c syslogRtDestGroup
Total Objects shown: 4

# syslog.RtDestGroup
tDn           : uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog
childAction   :
dn            : uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog]
lcOwn         : local
modTs         : 2016-10-19T00:06:32.562-05:00
rn            : rtdestGroup-[uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog]
status        :
tCl           : syslogSrc

# syslog.RtDestGroup
tDn           : uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog
childAction   :
dn            : uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog]
lcOwn         : local
modTs         : 2016-10-19T00:06:11.329-05:00
rn            : rtdestGroup-[uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog]
status        :
tCl           : syslogSrc
```

*Note: Repeat the "moquery -c syslogRtDestGroup" command on each APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

**syslogRtDestGroup (cont.)**

```
# syslog.RtDestGroup
tDn           : uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog
childAction   :
dn            : uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/fabric/moncommon/slsrc-deadbeef-moncommon-
syslog]
lcOwn         : local
modTs         : 2016-10-19T00:05:36.276-05:00
rn            : rtdestGroup-[uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog]
status        :
tCl           : syslogSrc

# syslog.RtDestGroup
tDn           : uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog
childAction   :
dn            : uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-
deadbeef-hsrp-syslog]
lcOwn         : local
modTs         : 2016-10-25T21:31:11.632-05:00
rn            : rtdestGroup-[uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog]
status        :
tCl           : syslogSrc
```

*Note: Repeat the "moquery -c syslogRtDestGroup" command on each APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*" ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

syslogSrc

```
apic1# moquery -c syslogSrc
Total Objects shown: 4

# syslog.Src
name          : deadbeef-hsrp-syslog
childAction   :
descr         :
dn            : uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog
incl          : all,audit,events,faults,session
lcOwn         : local
minSev        : information
modTs         : 2016-10-25T21:31:11.597-05:00
monPolDn      : uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp
nameAlias     :
rn            : slsrc-deadbeef-hsrp-syslog
status        :
uid           : 15374

# syslog.Src
name          : deadbeef-moninfra-syslog
childAction   :
descr         :
dn            : uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog
incl          : all,audit,events,faults,session
lcOwn         : local
minSev        : information
modTs         : 2016-11-30T17:32:35.132-05:00
monPolDn      : uni/infra/moninfra-default
nameAlias     :
rn            : slsrc-deadbeef-moninfra-syslog
status        :
uid           : 15374
```

*Note: Repeat the "moquery -c syslogSrc" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*" ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

syslogSrc (cont.)

```
# syslog.Src
name          : deadbeef-monfab-syslog
childAction   :
descr         :
dn            : uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog
incl          : all,audit,events,faults,session
lcOwn         : local
minSev        : information
modTs         : 2016-11-14T10:54:50.951-05:00
monPolDn      : uni/fabric/monfab-default
nameAlias     :
rn            : slsrc-deadbeef-monfab-syslog
status        :
uid           : 15374

# syslog.Src
name          : deadbeef-moncommon-syslog
childAction   :
descr         :
dn            : uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog
incl          : all,audit,events,faults,session
lcOwn         : local
minSev        : information
modTs         : 2016-11-14T10:54:18.641-05:00
monPolDn      : uni/fabric/moncommon
nameAlias     :
rn            : slsrc-deadbeef-moncommon-syslog
status        :
uid           : 15374uid          : 15374
```

*Note: Repeat the "moquery -c syslogSrc" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (APIC)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each APIC with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter).**

**syslogFacilityFilter**

```
apic1# moquery -c syslogFacilityFilter | grep -E "facility|minSev|monPolDn" | grep -A
2 default

facility     : default
minSev       : information
facility     : kern
```

*** After you have created the ACI Fabric's SYSLOG Source in the Fabric Policies "Monitoring Sources" for Fabric Policies - COMMON, configure the SYSLOG SYSTEM MESSAGES POLICY in the COMMON POLICY. The task for this step is to configure th **"Facility Filter"** for the **"default"** facility. Changing the Severity to **"information"** will record %**ACLLOG-5-ACLLOG_PKTLOG** messages in Syslog.*

***Note: Repeat the "moquery -c syslogFacilityFilter" command on each Leaf \Spine\APIC node configured for SYSLOG.***

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

**syslogGroup**

```
leaf1# moquery -c syslogGroup
Total Objects shown: 1

# syslog.Group
name             : deadbeef-syslogGrp
childAction      :
descr            : Group of Syslog Servers for the deadbeef network
dn               : uni/fabric/slgroup-deadbeef-syslogGrp
format           : aci
lcOwn            : policy
modTs            : 2016-11-30T18:22:38.347-05:00
monPolDn         : uni/fabric/monfab-default
nameAlias        :
remoteDestCount  : 2
rn               : slgroup-deadbeef-syslogGrp
status           :
uid              : 15374
```

*Note: Repeat the "moquery -c syslogGroup" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

syslogRemoteDest

```
leaf1# moquery -c syslogRemoteDest
Total Objects shown: 2

# syslog.RemoteDest
host               : 10.122.254.251
adminState         : enabled
childAction        :
descr              :
dn                 : uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.122.254.251
epgDn              : uni/tn-mgmt/mgmtp-default/oob-default
format             : aci
forwardingFacility : local7
ip                 :
lcOwn              : policy
modTs              : 2016-11-30T18:22:38.347-05:00
monPolDn           : uni/fabric/monfab-default
name               : nangaparbat.cisco.com
nameAlias          :
operState          : unknown
port               : 514
rn                 : rdst-10.122.254.251
severity           : information
status             :
uid                : 15374
vrfId              : 0
vrfName            : management
```

*Note: Repeat the "moquery -c syslogRemoteDest" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

syslogRemoteDest (cont.)

```
# syslog.RemoteDest
host               : 10.117.67.30
adminState         : enabled
childAction        :
descr              :
dn                 : uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.117.67.30
epgDn              : uni/tn-mgmt/mgmtp-default/inb-default
format             : aci
forwardingFacility : local7
ip                 :
lcOwn              : policy
modTs              : 2016-11-30T18:22:38.347-05:00
monPolDn           : uni/fabric/monfab-default
name               : deadbeef-macosx-vpn30
nameAlias          :
operState          : unknown
port               : 514
rn                 : rdst-10.117.67.30
severity           : warnings
status             :
uid                : 15374
vrfId              : 0
vrfName            : mgmt:inb
```

*Note: Repeat the "moquery -c syslogRemoteDest" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

**syslogProf**

```
leaf1# moquery -c syslogProf
Total Objects shown: 1

# syslog.Prof
adminState   : enabled
childAction  :
descr        :
dn           : uni/fabric/slgroup-deadbeef-syslogGrp/prof
lcOwn        : policy
modTs        : 2016-11-29T15:36:26.940-05:00
name         : syslog
nameAlias    :
port         : 514
rn           : prof
status       :
transport    : udp
uid          : 15374
```

*Note: Repeat the "moquery -c syslogProf" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

**syslogSrc**

```
leaf1# moquery -c syslogSrc
Total Objects shown: 3

# syslog.Src
name          : deadbeef-moninfra-syslog
childAction   :
descr         :
dn            : uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-11-30T17:32:35.454-05:00
monPolDn      : uni/infra/moninfra-default
nameAlias     :
rn            : slsrc-deadbeef-moninfra-syslog
status        :
uid           : 15374
```

*Note: Repeat the "moquery -c syslogSrc" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*" ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

syslogSrc (cont.)

```
# syslog.Src
name          : deadbeef-moncommon-syslog
childAction   :
descr         :
dn            : uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-11-29T15:36:25.408-05:00
monPolDn      : uni/fabric/moncommon
nameAlias     :
rn            : slsrc-deadbeef-moncommon-syslog
status        :
uid           : 15374

# syslog.Src
name          : deadbeef-monfab-syslog
childAction   :
descr         :
dn            : uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog
incl          : all,audit,events,faults,session
lcOwn         : policy
minSev        : information
modTs         : 2016-11-29T15:36:25.408-05:00
monPolDn      : uni/fabric/monfab-default
nameAlias     :
rn            : slsrc-deadbeef-monfab-syslog
status        :
uid           : 15374
```

*Note: Repeat the "moquery -c syslogSrc" command on each Leaf \Spine\APIC node configured for SYSLOG.*

# Verify the Syslog configuration (Leaf\Spine)

## "moquery"

- **Managed Object(MO) Queries is another way to verify configuration of SYSLOG Policies. On each Leaf\Spine with SYSLOG configured, run "*moquery -c [object class]*"**
  **ie. (syslogGroup, syslogRemoteDest, syslogProf, syslogSrc, syslogFacilityFilter).**

**syslogFacilityFilter**

```
leaf1# moquery -c syslogFacilityFilter | grep -E "facility|minSev|monPolDn" | grep -A 2 default

facility     : default
minSev       : information
facility     : local2
```

*** After you have created the ACI Fabric's SYSLOG Source in the Fabric Policies "Monitoring Sources" for Fabric Policies - COMMON, configure the SYSLOG SYSTEM MESSAGES POLICY in the COMMON POLICY. The task for this step is to configure th **"Facility Filter"** for the **"default"** facility. Changing the Severity to **"information"** will record %**ACLLOG-5-ACLLOG_PKTLOG** messages in Syslog.*

***Note: Repeat the "moquery -c syslogFacilityFilter" command on each Leaf \Spine\APIC node configured for SYSLOG.***

# Verify ACI SYSLOG Configuration - “VISORE”

Another tool to verify SYSLOG configuration is VISORE. Enclosed are some samples of the VISORE information related to the SYSLOG configuration.

# Verify ACI SYSLOG Configuration

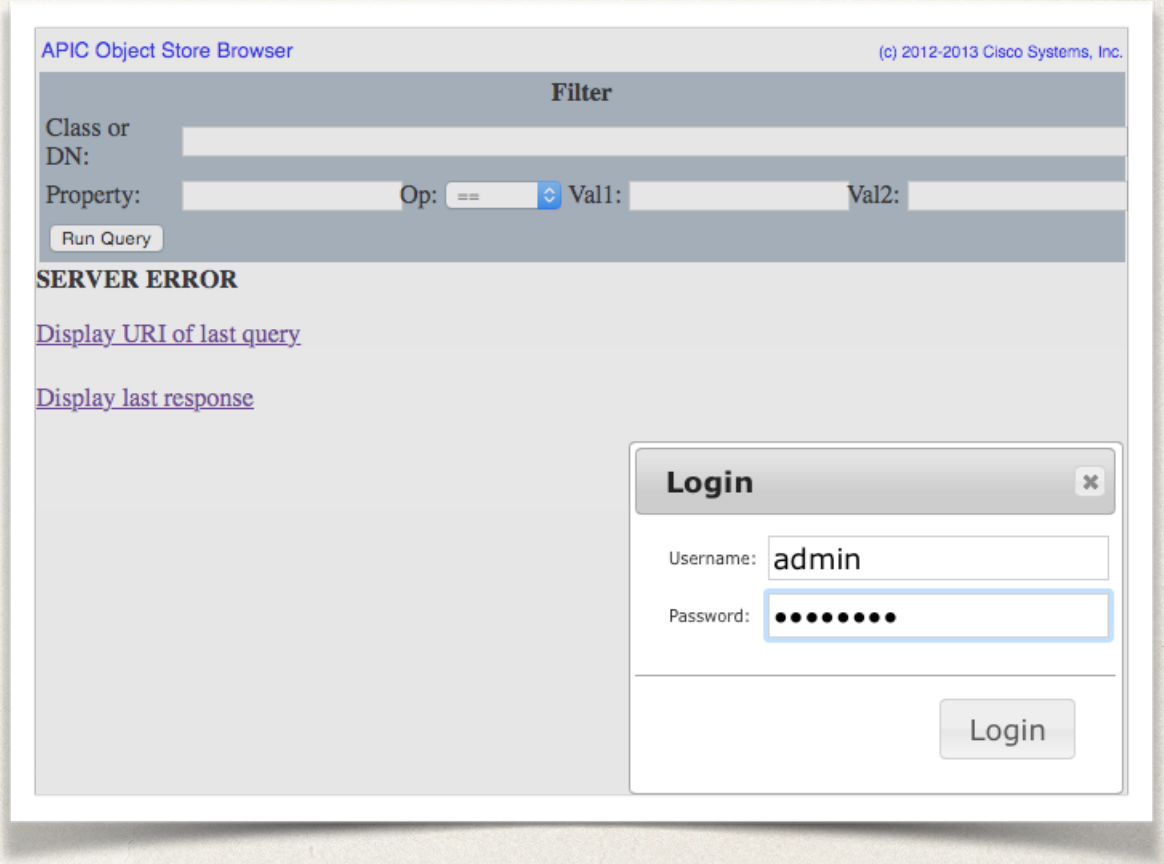
## "VISORE"

- **Another tool to verify SYSLOG configuration is VISORE. Enclosed are some samples of the VISORE information related to the SYSLOG configuration.**
  **(syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter)**

- **To access VISORE, use a browser using the following address:**

  ***https://<APIC_IP_address>/visore.html***

  *note: use your APIC Admin Credentials to login to VISORE*



# Verify ACI SYSLOG Configuration

## "VISORE"

- **Managed Object(MO) Classes for SYSLOG Policy configuration in ACI:**

**syslogGroup** - The syslog destination group contains all information required to send syslog messages to a group of destinations.

**syslogRemoteDest** - The syslog remote destination host enables you to specify syslog servers to which messages from the APIC and fabric nodes should be forwarded.

**syslogProf** - Represents the configuration parameters used for this protocol.

**syslogRtDestGroup** - A target relation to the syslog destination group.

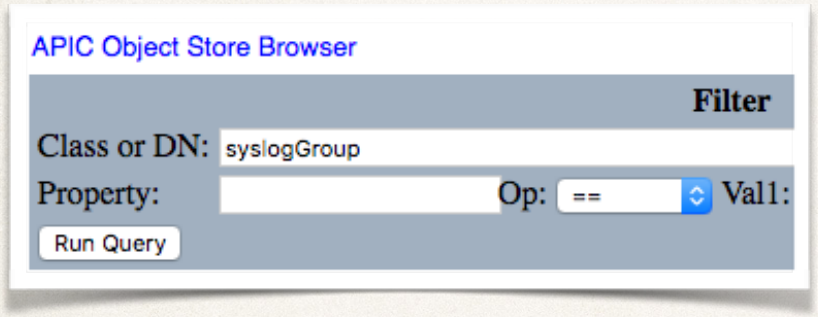
**syslogSrc** - The syslog source configures a syslog source that specifies the minimum severity of items to be sent as syslog messages to the syslog servers in the destination group.

**syslogFacilityFilter** - Facility and Severity levels for filters used for monitoring Syslog messages.

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogGroup**




| syslogGroup | |
|---|---|
| childAction | |
| descr | Group of Syslog Servers for the deadbeef network |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp |
| format | aci |
| lcOwn | local |
| modTs | 2016-11-30T18:22:38.309-05:00 |
| monPolDn | uni/fabric/monfab-default |
| name | deadbeef-syslogGrp |
| nameAlias | |
| remoteDestCount | 2 |
| status | |
| uid | 15374 |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogRemoteDest**

APIC Object Store Browser

Class or DN: syslogRemoteDest

Property: Op:

Run Query

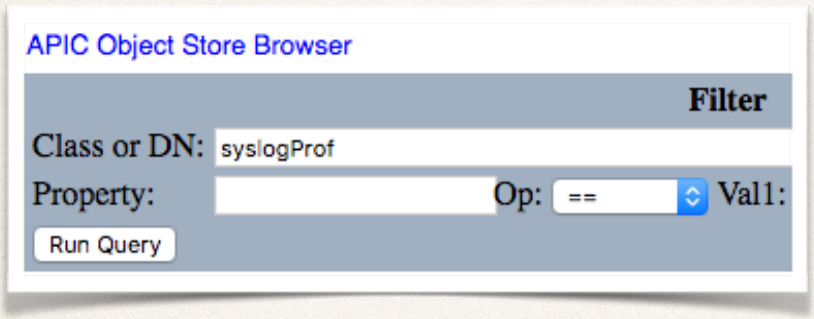
| syslogRemoteDest | |
|---|---|
| adminState | enabled |
| childAction | |
| descr | |
| dn | uni/fabric/slgroup-deadb |
| epgDn | uni/tn-mgmt/mgmtp-defa |
| format | aci |
| forwardingFacility | local7 |
| host | 10.122.254.251 |
| ip | |
| lcOwn | local |
| modTs | 2016-11-14T10:37:52.81 |
| monPolDn | uni/fabric/monfab-defaul |
| name | nangaparbat.cisco.com |
| nameAlias | |
| operState | unknown |
| port | 514 |
| severity | information |
| status | |
| uid | 15374 |
| vrfId | 0 |
| vrfName | |

| syslogRemoteDest | |
|---|---|
| adminState | enabled |
| childAction | |
| descr | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/rdst-10.117.67.30 |
| epgDn | uni/tn-mgmt/mgmtp-default/inb-default |
| format | aci |
| forwardingFacility | local7 |
| host | 10.117.67.30 |
| ip | |
| lcOwn | local |
| modTs | 2016-11-30T18:22:38.309-05:00 |
| monPolDn | uni/fabric/monfab-default |
| name | deadbeef-macosx-vpn30 |
| nameAlias | |
| operState | unknown |
| port | 514 |
| severity | warnings |
| status | |
| uid | 15374 |
| vrfId | 0 |
| vrfName | |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogProf**




| syslogProf | |
|---|---|
| adminState | enabled |
| childAction | |
| descr | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/prof |
| lcOwn | local |
| modTs | 2016-10-18T23:51:46.876-05:00 |
| name | syslog |
| nameAlias | |
| port | 514 |
| status | |
| transport | udp |
| uid | 15374 |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogRtDestGroup**

APIC Object Store Browser

Class or DN: syslogRtDestGroup

Property:

Run Query

| syslogRtDestGroup | |
|---|---|
| childAction | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog] |
| lcOwn | local |
| modTs | 2016-10-19T00:06:32.562-05:00 |
| status | |
| tCl | syslogSrc |
| tDn | uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog |

| syslogRtDestGroup | |
|---|---|
| childAction | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog] |
| lcOwn | local |
| modTs | 2016-10-19T00:06:11.329-05:00 |
| status | |
| tCl | syslogSrc |
| tDn | uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogRtDestGroup (cont.)**

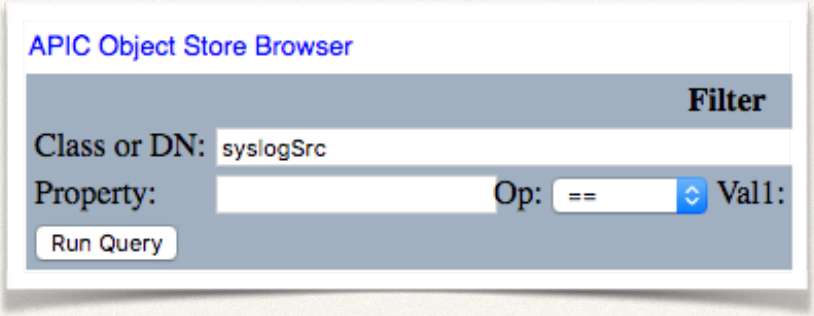
| syslogRtDestGroup | |
|---|---|
| childAction | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog] |
| lcOwn | local |
| modTs | 2016-10-19T00:05:36.276-05:00 |
| status | |
| tCl | syslogSrc |
| tDn | uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog |

| syslogRtDestGroup | |
|---|---|
| childAction | |
| dn | uni/fabric/slgroup-deadbeef-syslogGrp/rtdestGroup-[uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog] |
| lcOwn | local |
| modTs | 2016-10-25T21:31:11.632-05:00 |
| status | |
| tCl | syslogSrc |
| tDn | uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogSrc**




| | syslogSrc | syslogSrc |
|---|---|---|
| childAction | | |
| descr | | |
| dn | uni/infra/moninfra-default/slsrc-deadbeef-moninfra-syslog | uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp/slsrc-deadbeef-hsrp-syslog |
| incl | all,audit,events,faults,session | all,audit,events,faults,session |
| lcOwn | local | local |
| minSev | information | information |
| modTs | 2016-11-30T17:32:35.132-05:00 | 2016-10-25T21:31:11.597-05:00 |
| monPolDn | uni/infra/moninfra-default | uni/tn-deadbeef-hsrp/monepg-deadbeef-hsrp |
| name | deadbeef-moninfra-syslog | deadbeef-hsrp-syslog |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogSrc (cont.)**

| syslogSrc | |
|---|---|
| childAction | |
| descr | |
| dn | uni/fabric/monfab-default/slsrc-deadbeef-monfab-syslog |
| incl | all,audit,events,faults,session |
| lcOwn | local |
| minSev | information |
| modTs | 2016-11-14T10:54:50.951-05:00 |
| monPolDn | uni/fabric/monfab-default |
| name | deadbeef-monfab-syslog |

| syslogSrc | |
|---|---|
| childAction | |
| descr | |
| dn | uni/fabric/moncommon/slsrc-deadbeef-moncommon-syslog |
| incl | all,audit,events,faults,session |
| lcOwn | local |
| minSev | information |
| modTs | 2016-11-14T10:54:18.641-05:00 |
| monPolDn | uni/fabric/moncommon |
| name | deadbeef-moncommon-syslog |

# Verify ACI SYSLOG Configuration (cont.)

## "VISORE"

**syslogFacilityFilter ("default")**

APIC Object Store Browser

Filter

Class or DN: syslogFacilityFilter

Property: facility Op: == Val1: default

Run Query

| syslogFacilityFilter | |
|---|---|
| childAction | |
| descr | |
| dn | uni/fabric/moncommon/sysmsgp/ff-default |
| facility | default |
| lcOwn | local |
| minSev | information |
| modTs | 2016-11-14T10:36:09.650-05:00 |

# Verify ACI SYSLOG Configuration - “REST API”

Another tool to verify SYSLOG configuration is REST API. Enclosed are some samples of the REST API using POSTMAN application to gather information related to the SYSLOG configuration.

# Verify ACI SYSLOG Configuration

## "REST API"

- **Managed Object(MO) Classes for SYSLOG Policy configuration in ACI:**

**syslogGroup** - The syslog destination group contains all information required to send syslog messages to a group of destinations.

**syslogRemoteDest** - The syslog remote destination host enables you to specify syslog servers to which messages from the APIC and fabric nodes should be forwarded.

**syslogProf** - Represents the configuration parameters used for this protocol.

**syslogRtDestGroup** - A target relation to the syslog destination group.

**syslogSrc** - The syslog source configures a syslog source that specifies the minimum severity of items to be sent as syslog messages to the syslog servers in the destination group.

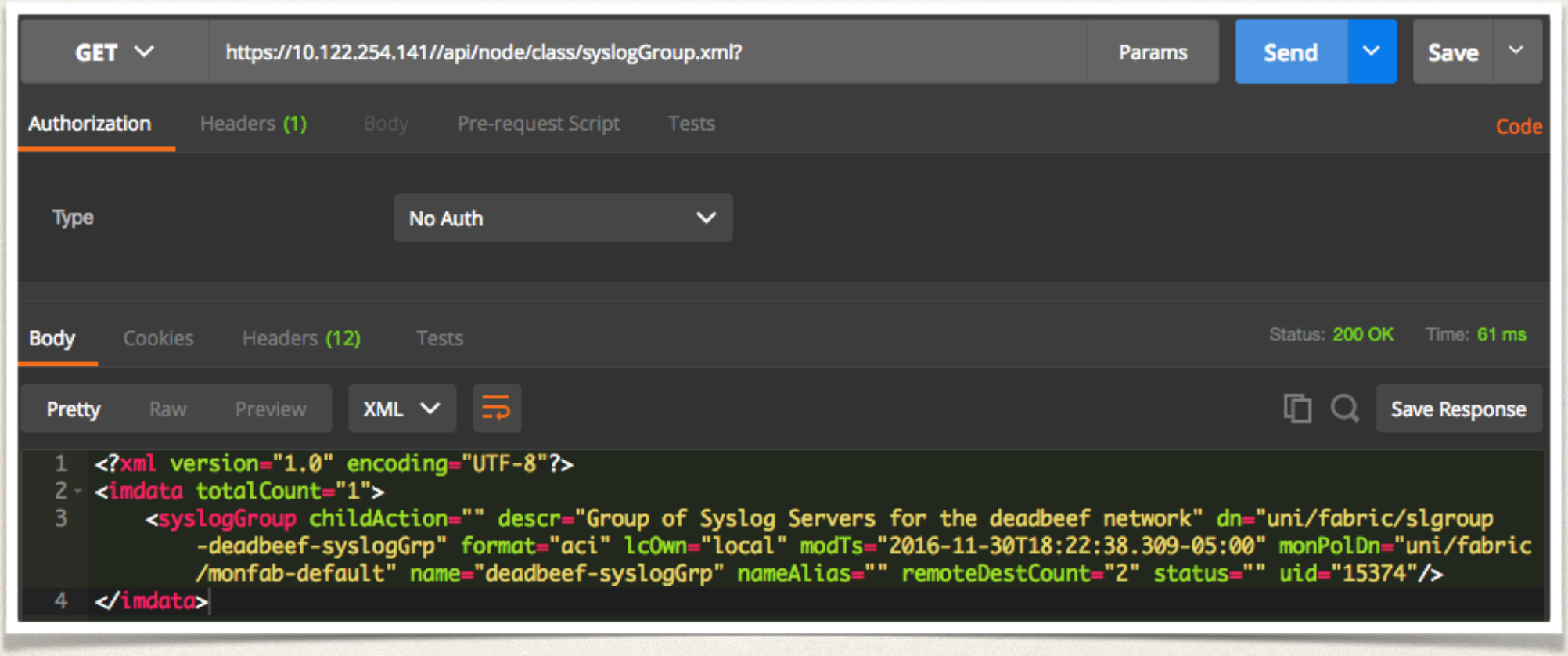
**syslogFacilityFilter** - Facility and Severity levels for filters used for monitoring Syslog messages.

# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogGroup**

**/api/node/class/syslogGroup.xml?**

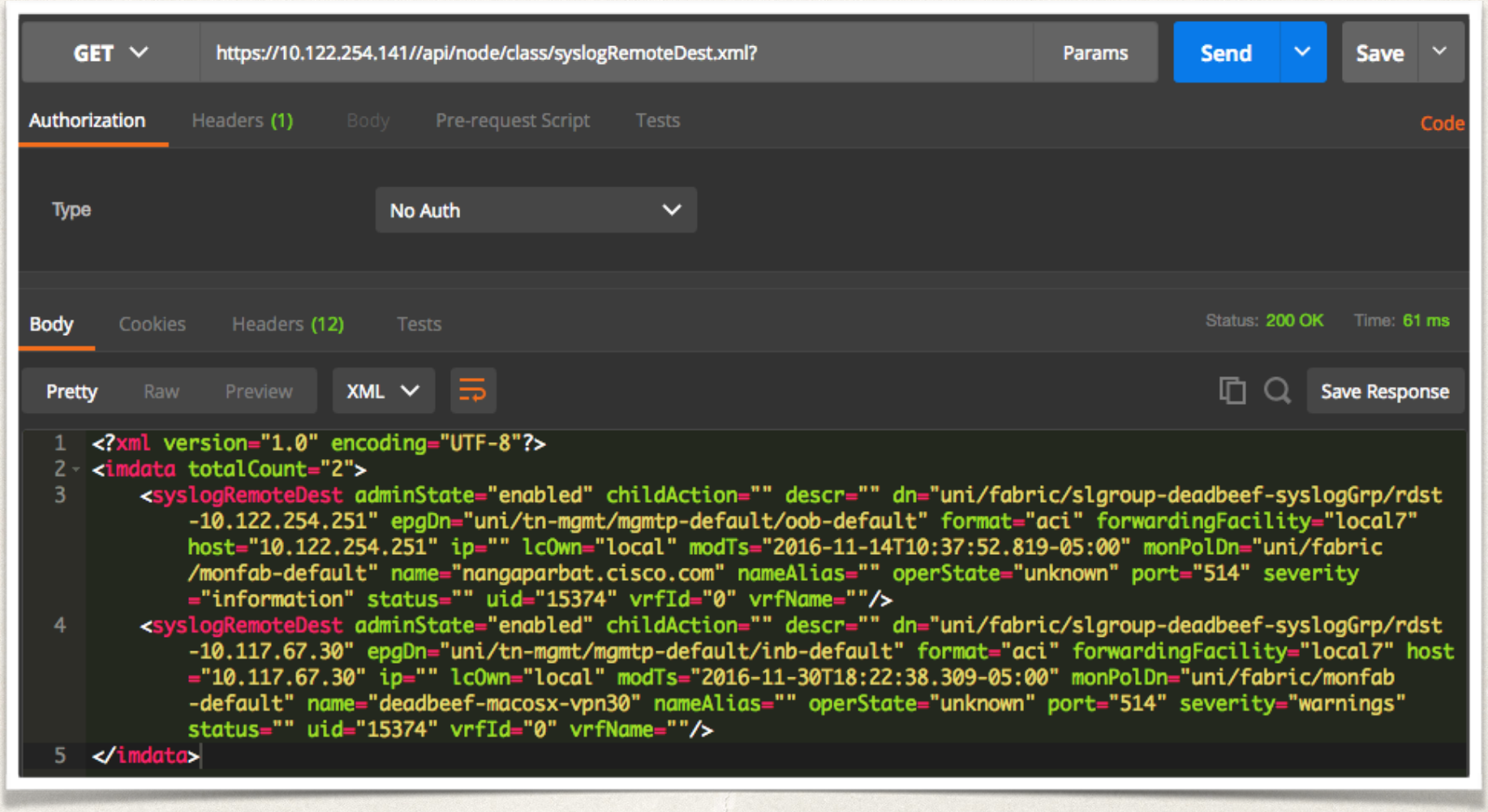


# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogRemoteDest**

**/api/node/class/syslogRemoteDest.xml?**

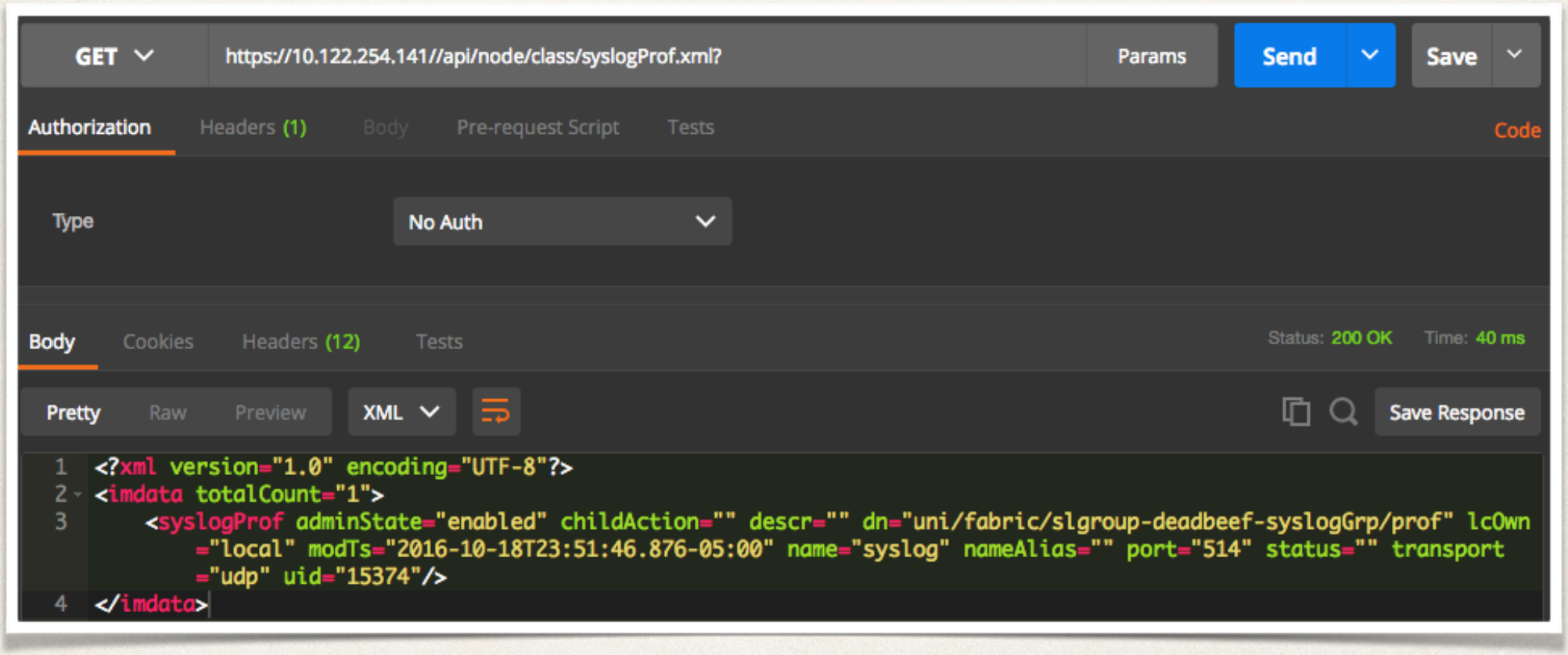


# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogProf**

**/api/node/class/syslogProf.xml?**

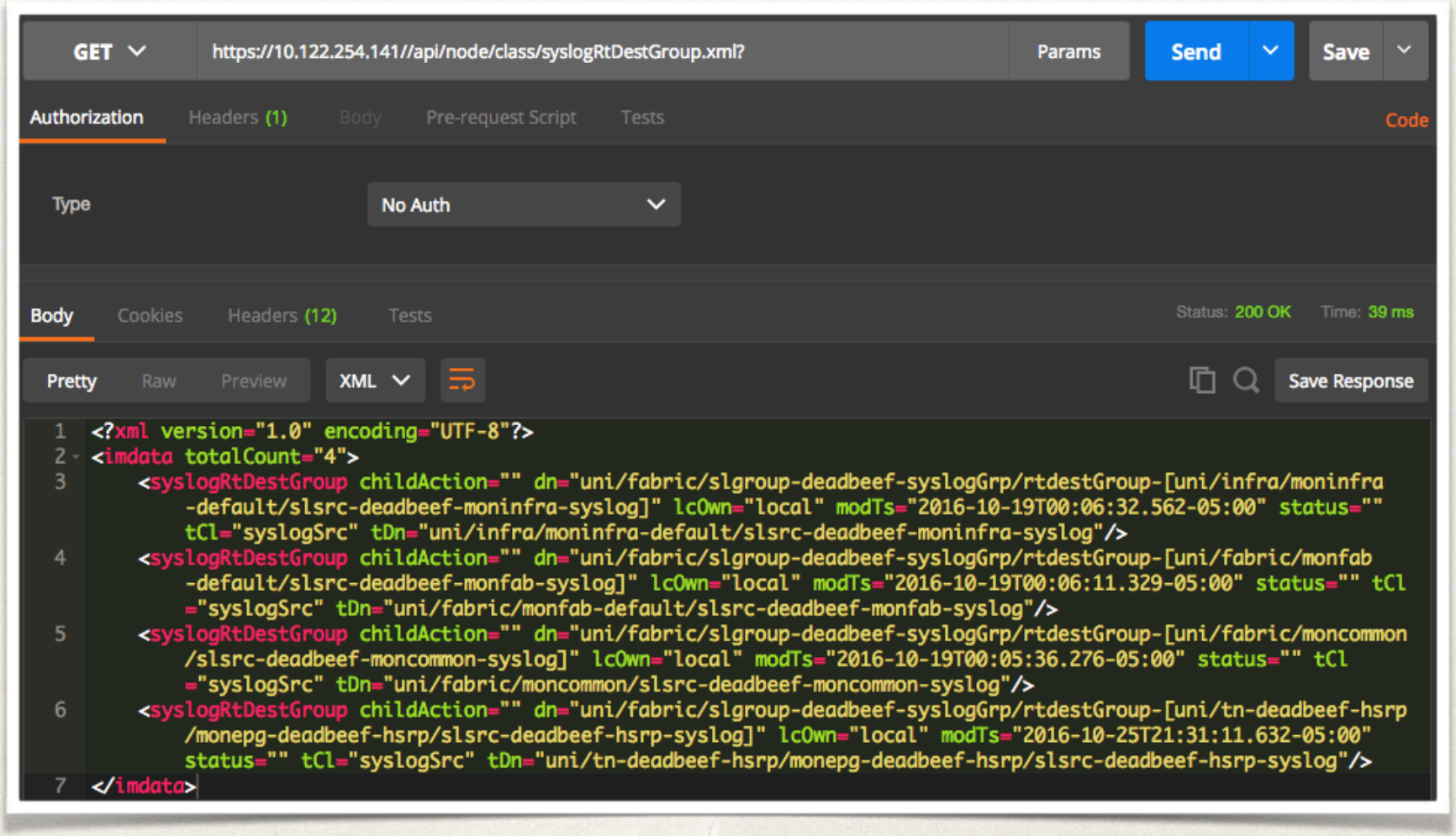


# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogRtDestGroup**

**/api/node/class/syslogRtDestGroup.xml?**

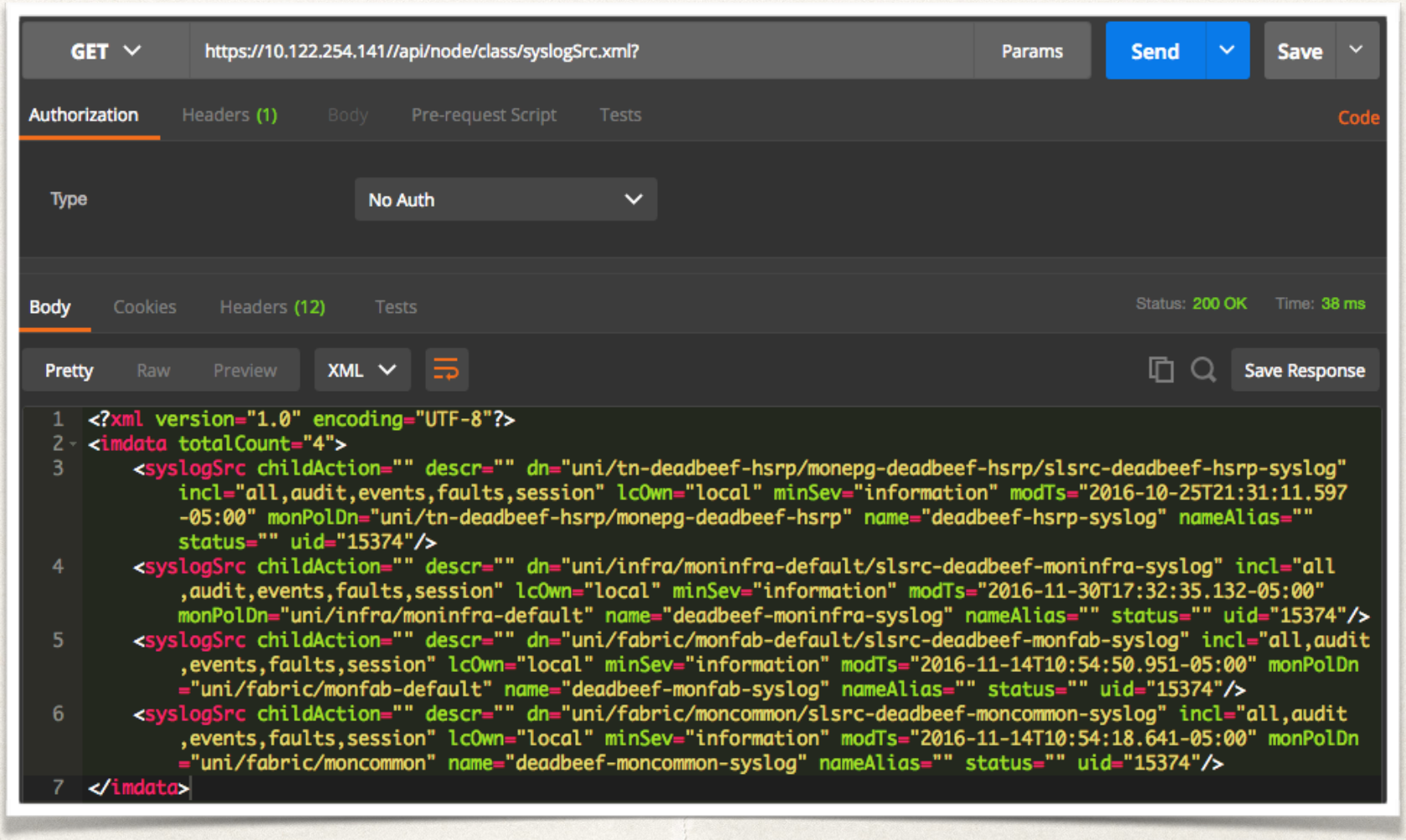


# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogSrc**

**/api/node/class/syslogSrc.xml?**

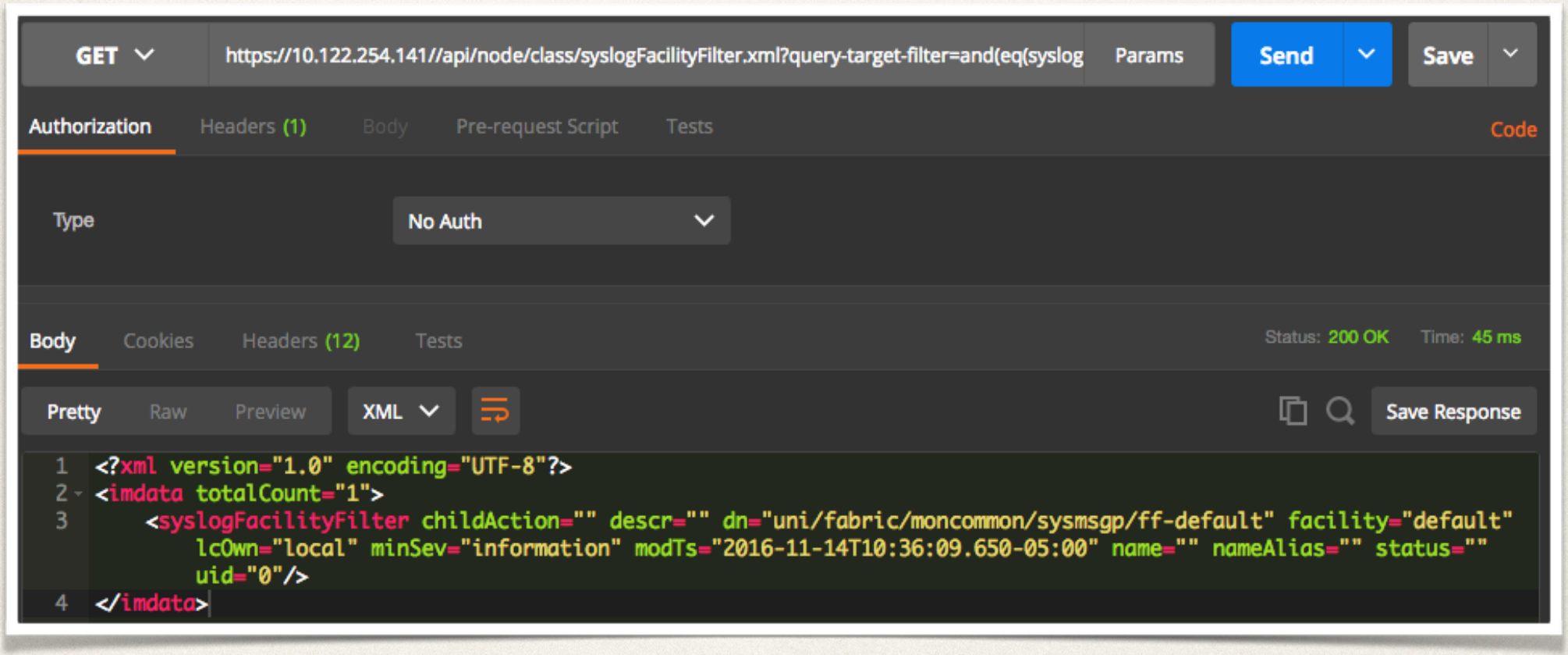


# Verify ACI SYSLOG Configuration (cont.)

## "REST API"

**syslogFacilityFilter ("default")**

**/api/node/class/syslogFacilityFilter.xml?query-target-filter=and(eq(syslogFacilityFilter.facility,"default"))**



# Verify ACI SYSLOG Configuration - “Logical Model”

Checking the Logical Model on the APIC is another way to verify configuration of SYSLOG Policies. On an APIC , run “ *Cat ….*/*summary* ” on the key components of the SYSLOG configuration for the ACI Fabric. The following is a list of SUMMARY files to use to verify the SYSLOG configuration.

# Verify ACI SYSLOG Configuration (cont.)

## "Logical Model"

- **Checking the Logical Model on the APIC is another way to verify configuration of SYSLOG Policies. On an APIC , run " *Cat ..../summary* " on the key components of the SYSLOG configuration for the ACI Fabric. The following is a list of SUMMARY files to use to verify the SYSLOG configuration.**
  - cat /aci/tenants/mgmt/security-policies/out-of-band-contracts/summary
  - cat /aci/tenants/mgmt/security-policies/filters/summary
  - cat /aci/tenants/mgmt/node-management-epgs/default/out-of-band/default/summary
  - cat /aci/admin/external-data-collectors/monitoring-destinations/syslog/*/operational/summary
  - cat /aci/fabric/fabric-policies/monitoring-policies/monitoring-policy-default/callhome-snmp-syslog/all/syslog*/summary
  - cat /aci/fabric/fabric-policies/monitoring-policies/common-policy/callhome-snmp-syslog/syslog/*/summary
  - cat /aci/fabric/access-policies/monitoring-policies/default/callhome-snmp-syslog/all/syslog*/summary

# Verify SYSLOG Messages are being sent by LEAF\SPINE\APIC

The CDET “CSCuy61215 ACI: Enhancement to Send Test to Syslog Destinations“ added a tool to test the SYSLOG configuration for the LEAF\SPINE\APIC nodes. The following section will examples on how to test the Syslog configuration using the CLI Syslog Test Feature.

# Test the Syslog configuration using the CLI Syslog Test Feature

- Test Syslog Configuration Steps (using the APIC iNXOS CLI):
  1. SSH or use CIMC\SOL to access APIC1.
  2. Use the "**logit**" CLI command to verify Syslog configuration.
     *Note: The Syslog "**logit**" test comand will be available in ACI versions from Congo Maintance Releases and later.*
  3. Perform a "**logit**" test for each configured remote destination. Use Node 1 (APIC1) for each test. *(To send messages from Leaf & Spines, you must use the NodeId for each switch)*

**Command Syntax:**

# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message>
# logit severity <severity> dest-grp <destGroup> server <remoteDest> <message> node <id>

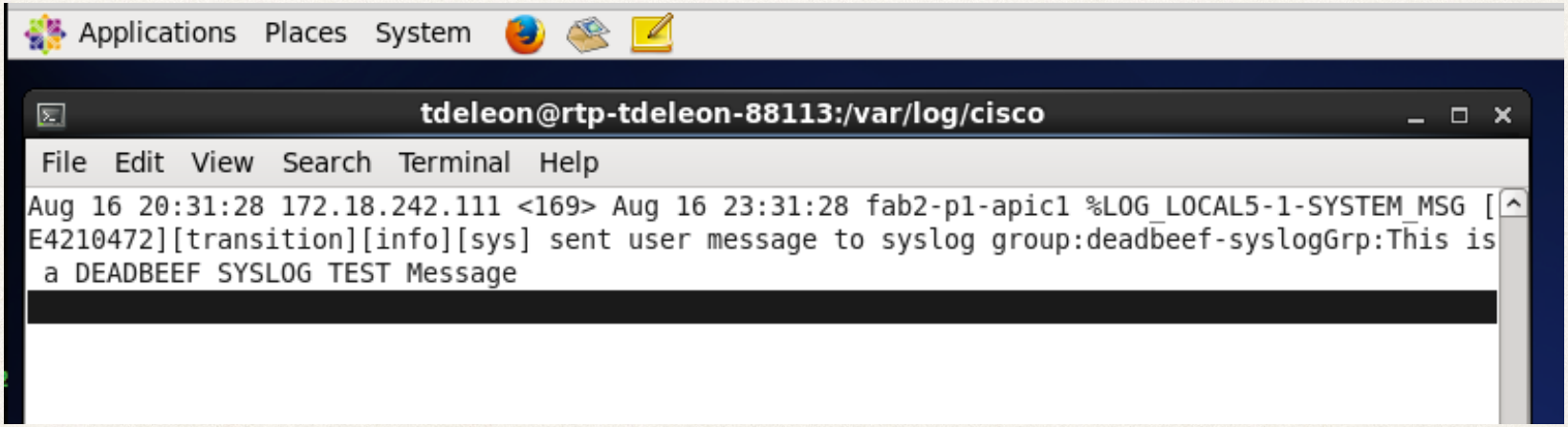
*Remember to run a test for each configured Syslog remote destination. The test command for each node is run on the APIC for all nodes. The APIC will signal the remote nodes to send a Syslog Test Message.*

# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **INB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.117.67.30 "This is a DEADBEEF SYSLOG TEST Message"** node **1**

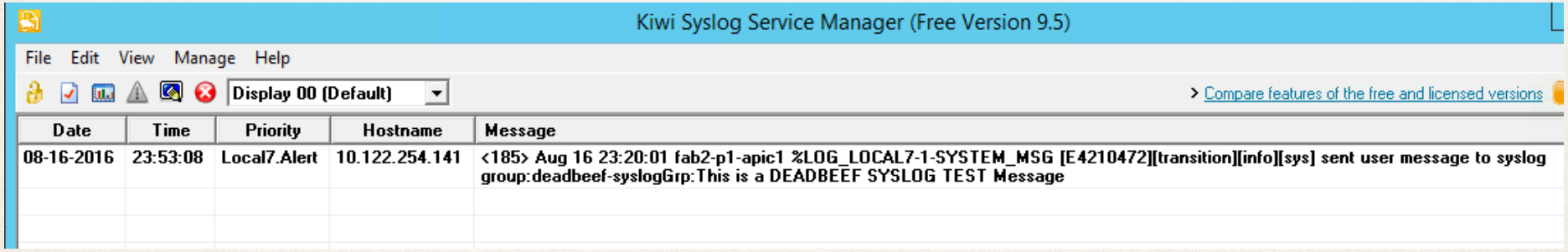


# Test the Syslog configuration using the CLI Syslog Test Feature (Cont.)

Test Syslog Example:

Server Test using the **OOB MGMT EPG**:

apic1# logit severity **1** dest-grp **deadbeef-syslogGrp** server **10.122.254.251 "This is a DEADBEEF SYSLOG TEST Message"** node **1**




**You can download a free version fo the Kiwi Syslog Server for Windows at:**
http://www.kiwisyslog.com

# Troubleshooting the ACI SYSLOG Configuration on the APIC

The following section will give examples on how to troubleshoot the Syslog configuration on the APIC. *Note: Some of the following commands may require ROOT access. Temporary "Root" access requires assistance from a Cisco ACI TAC Engineer.*

# Debugging SYSLOG on the APIC

**In addition to the "Show" commands that listed earlier to verify the SYSLOG configuration on APIC Controllers, you can use some additional commands to gather more information in regards to SYSLOG. Some of the following commands may require ROOT access. Temporary "Root" access requires assistance from a Cisco ACI TAC Engineer.**

As mentioned earlier, SYSLOG on APIC using OOB management EPG **does not require** an explicit "**Out-Of-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**). That said, it is a good practice to go ahead and create a specific filter for SYSLOG and add it to the filter list in your OOB Contract Subject configuration. *Note: In earlier versions of ACI firmware, certain ports were always open and a contract was not needed for SYSLOG support on the Leaf and Spine nodes.*

SYSLOG on APIC using INB management EPG **requires** an explicit "**In-Band Contract**" on the APIC for enabling the SYSLOG port (**UDP:514**). Unless the contract is created, The SYSLOG packets will be dropped by the Border Leaf with the L3 Out used by the fabric for MGMT Access. ***This is different from enabling/disabling the SYSLOG protocol in monitoring policies.***

Also in addition to contracts being needed, Node Management Address(s) in the Tenant mgmt need to be configured for the APIC(s). Verify that the APIC Node management address(s) are configured also.

# Debugging SYSLOG on APIC (cont.)

- On each APIC, verify the "rsyslogd" process is running. Record the process ID (pid) for "rsyslogd". You can use one or both of the following commands:
  - netstat -p | grep syslog
  - ps -A | grep rsyslog
  - pidof rsyslogd

For Example:
(note: some output has been abbreviated for display purposes)

```
root@rtp-f2-p1-apic1:~# netstat -p | grep syslog
unix  19     [ ]         DGRAM                    112323   5908/svc_ifc_eventm  /var/run/mgmt/syslog_socket
unix  10     [ ]         DGRAM                    11512    2081/rsyslogd        /dev/log

root@rtp-f2-p1-apic1:~# ps -A | grep rsyslog
 2081 ?        00:00:01 rsyslogd

root@rtp-f2-p1-apic1:~# pidof rsyslogd
2081
```

* rsyslogd PID = **2081**

*Note: Repeat on each APIC node having issues with the SYSLOG feature.*

# Debugging SYSLOG on APIC (cont.)

- On each APIC, verify the "rsyslogd" process is running. Gather the rsylogd version and see if there were any rsyslog errors in the APIC kernel log. You can use one or both of the following commands:
  - rsyslogd -version
  - cat /var/log/messages | grep rsyslog
    *Note: the "/var/log/messages" is the kernel log file not the SYSLOG message file.*

For Example:
(note: some output has been abbreviated for display purposes)

```
root@rtp-f2-p1-apic1:~# rsyslogd -version
rsyslogd 7.4.7, compiled with:
    FEATURE_REGEXP:                         Yes
    FEATURE_LARGEFILE:                      No
    GSSAPI Kerberos 5 support:              Yes
    FEATURE_DEBUG (debug build, slow code): No
    32bit Atomic operations supported:      Yes
    64bit Atomic operations supported:      Yes
    Runtime Instrumentation (slow code):    No
    uuid support:                           Yes
```

```
root@rtp-f2-p1-apic1:~# cat /var/log/messages | grep rsyslog
Dec 10 20:20:24 localhost rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-pid="2081" x-info="http://www.rsyslog.com"] start
Dec 10 20:20:24 localhost rsyslogd-2184: action '1' treated as ':omusrmsg:1' - please change syntax, '1' will not be supported in the
future [try http://www.rsyslog.com/e/2184 ]
Dec 10 20:20:24 localhost rsyslogd-3000: invalid character in selector line - ';template' expected
Dec 10 20:20:24 localhost rsyslogd-2207: error during parsing file /etc/rsyslog.conf, on or before line 60: errors occured in file '/
etc/rsyslog.conf' around line 60 [try http://www.rsyslog.com/e/2207 ]
```

***Note: Repeat on each APIC node having issues with the SYSLOG feature.***

# Debugging SYSLOG on APIC

## "netstat"

- On each APIC, gather some network statistics in relation to "syslog" and "syslog ports". You use the output to verify the management interfaces are transmitting & recieving packets. You can use the following commands to gather network status:
  - netstat -ai | egrep "Iface|bond0.1100"
  - netstat -ai | egrep "Iface|bond0.1100|oobmgmt"
  - netstat -nr

*Note: "bond0.1100" is the vlan encap configured on the INB mgmt EPG for APIC. Replace "1100" for your configured vlan encap.*

For Example:
(note: some output has been abbreviated for display purposes)

```
root@rtp-f2-p1-apic1:~# netstat -ai | egrep "Iface|bond0.1100"
Iface        MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
bond0.1100  1496     7748      0     77 0          10067      0      0      0 BMRU

root@rtp-f2-p1-apic1:~# netstat -ai | egrep "Iface|bond0.1100|oobmgmt"
Iface        MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
bond0.1100  1496     7794      0     77 0          10099      0      0      0 BMRU
oobmgmt     1500   349289      0      0 0         284699      0      0      0 BMRU

root@rtp-f2-p1-apic1:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         172.18.242.65   0.0.0.0         UG        0 0          0 bond0.1100
0.0.0.0         10.122.254.1    0.0.0.0         UG        0 0          0 oobmgmt
10.122.254.0    0.0.0.0         255.255.255.0   U         0 0          0 oobmgmt
172.18.242.64   0.0.0.0         255.255.255.192 U         0 0          0 bond0.1100
172.18.242.65   0.0.0.0         255.255.255.255 UH        0 0          0 bond0.1100
```

*Note: Repeat on each APIC node having issues with the SYSLOG feature.*

# Debugging SYSLOG on APIC

## "iptables"

- On each APIC, check the "**iptables**" to see what rules are programmed for SYSLOG . The programming of "iptable" rules for SYSLOG is not necessary for the SYSLOG configuration and deployment to APICs. But since the APICs and the Leaf\Spine nodes share the same policy, you can check the programming on the APICs also. You can use the following commands to check the "iptable" rules:
  - iptables -S | grep 514
    *Note: 514 is the default Syslog port. If you use other ports for Syslog, make sure to check all ports.*
  - iptables --list | grep syslog
  - iptables --list -v | grep syslog
  - iptables --list -v

For Example:

```
root@rtp-f2-p1-apic1:~# iptables -S | grep 514
-A fp-28 -p udp -m udp --dport 514 -j ACCEPT

root@rtp-f2-p1-apic1:~# iptables --list | grep syslog
ACCEPT     udp  --  anywhere             anywhere             udp dpt:syslog

root@rtp-f2-p1-apic1:~# iptables --list -v | grep syslog
    0     0 ACCEPT     udp  --  any    any     anywhere             anywhere             udp dpt:syslog

root@rtp-f2-p1-apic1:~# iptables --list -v

Chain fp-28 (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  any    any     anywhere             anywhere             udp dpt:syslog

Note:  the "fp-28" listed above is the OOB contract & filters.  INB contracts & filters are not programmed
since the filtering is applied at the border or services leaf.
```

# Debugging SYSLOG on APIC

## Verify sending SYSLOG messages using “tcpdump”

- Access the APIC as "root" user and use "**tcpdump**" command to verify SYSLOG messages are being sent. Use UDP port **514** or any other UDP Ports that are configured for the SYSLOG server destinations in the ACI SYSLOG Monitoring Group. You can use the following "**tcpdump**" commands to check for SYSLOG messages on APIC Nodes:
  - tcpdump -i oobmgmt -f port 514
  - tcpdump -i bond0.1100 -f port 514
  - tcpdump -vvxi oobmgmt udp port 514
  - tcpdump -vvxi bond0.1100 udp port 514

For Example:

**APIC (INB) -> Destination Syslog Server address is 10.117.67.30**

```
root@rtp-f2-p1-apic1:~# tcpdump -i bond0.1100 -f port 514
tcpdump: /usr/lib64/libcrypto.so.10: no version information available (required by tcpdump)
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on bond0.1100, link-type EN10MB (Ethernet), capture size 65535 bytes
18:28:11.032900 IP rtp2-apic1-inb.cisco.com.58612 > 10.117.67.30.syslog: SYSLOG local7.info, length: 183
18:28:17.095071 IP rtp2-apic1-inb.cisco.com.58612 > 10.117.67.30.syslog: SYSLOG local7.info, length: 342
18:28:17.095186 IP rtp2-apic1-inb.cisco.com.58612 > 10.117.67.30.syslog: SYSLOG local7.info, length: 342
18:28:20.128574 IP rtp2-apic1-inb.cisco.com.58612 > 10.117.67.30.syslog: SYSLOG local7.info, length: 268
18:28:20.128683 IP rtp2-apic1-inb.cisco.com.58612 > 10.117.67.30.syslog: SYSLOG local7.info, length: 268
```

# Debugging SYSLOG on APIC

## Verify sending SYSLOG messages using “tcpdump”

For Example:

**APIC (INB) -> Destination Syslog Server address is 10.117.67.30**

```
root@rtp-f2-p1-apic1:~# tcpdump -vvxi bond0.1100 udp port 514
tcpdump: /usr/lib64/libcrypto.so.10: no version information available (required by tcpdump)
tcpdump: listening on bond0.1100, link-type EN10MB (Ethernet), capture size 65535 bytes

18:35:44.611564 IP (tos 0x0, ttl 64, id 62421, offset 0, flags [DF], proto UDP (17), length 222)
    rtp2-apic1-inb.cisco.com.58612 > rtp-tdeleon-88113.cisco.com.syslog: [udp sum ok] SYSLOG, length: 194
    Facility local7 (23), Severity alert (1)
    Msg: <185> Dec 10 18:35:44 rtp-f2-p1-apic1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info][sys] sent user
message to syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST Message\0x0a
    0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
    0x0010:  3020 3138 3a33 353a 3434 2072 7470 2d66
    0x0020:  322d 7031 2d61 7069 6331 2025 4c4f 475f
    0x0030:  4c4f 4341 4c37 2d31 2d53 5953 5445 4d5f
    0x0040:  4d53 4720 5b45 3432 3130 3437 325d 5b74


18:35:51.545480 IP (tos 0x0, ttl 64, id 1551, offset 0, flags [DF], proto UDP (17), length 212)
    rtp2-apic1-inb.cisco.com.58612 > rtp-tdeleon-88113.cisco.com.syslog: [udp sum ok] SYSLOG, length: 184
    Facility local7 (23), Severity info (6)
    Msg: <190> Dec 10 18:35:51 rtp-f2-p1-apic1 %LOG_LOCAL7-6-SYSTEM_MSG [refresh,session][info][subj-[uni/userext/
user-admin]/sess-4297292524] From-10.122.254.251-client-type-REST-Success\0x0a
    0x0000:  3c31 3930 3e3c 3139 303e 2044 6563 2031
    0x0010:  3020 3138 3a33 353a 3531 2072 7470 2d66
    0x0020:  322d 7031 2d61 7069 6331 2025 4c4f 475f
    0x0030:  4c4f 4341 4c37 2d36 2d53 5953 5445 4d5f
    0x0040:  4d53 4720 5b72 6566 7265 7368 2c73 6573
```

# Debugging SYSLOG on APIC

## Verify sending SYSLOG messages using “tcpdump”

For Example:

**APIC (OOB) -> Destination Syslog Server address is 10.122.254.251(nangaparbat.cisco.com)**

```
root@rtp-f2-p1-apic1:~# tcpdump -i oobmgmt -f port 514
tcpdump: /usr/lib64/libcrypto.so.10: no version information available (required by tcpdump)
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on oobmgmt, link-type EN10MB (Ethernet), capture size 65535 bytes
18:42:15.788536 IP fab2-p1-apic1.cisco.com.50217 > nangaparbat.cisco.com.syslog: SYSLOG local7.alert, length: 194
18:46:47.516551 IP fab2-p1-apic1.cisco.com.50217 > nangaparbat.cisco.com.syslog: SYSLOG local7.alert, length: 194
```

**APIC (OOB) -> Destination Syslog Server address is 10.122.254.251(nangaparbat.cisco.com)**

```
root@rtp-f2-p1-apic1:~# tcpdump -vvxi oobmgmt udp port 514
tcpdump: /usr/lib64/libcrypto.so.10: no version information available (required by tcpdump)
tcpdump: listening on oobmgmt, link-type EN10MB (Ethernet), capture size 65535 bytes
18:47:28.519564 IP (tos 0x0, ttl 64, id 43238, offset 0, flags [DF], proto UDP (17), length 222)
    fab2-p1-apic1.cisco.com.50217 > nangaparbat.cisco.com.syslog: [bad udp cksum 0x1359 -> 0x33d6!] SYSLOG, length: 194
     Facility local7 (23), Severity alert (1)
     Msg: <185> Dec 10 18:47:28 rtp-f2-p1-apic1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info][sys] sent user message to
syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST Message\0x0a
     0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
     0x0010:  3020 3138 3a34 373a 3238 2072 7470 2d66


18:48:09.017042 IP (tos 0x0, ttl 64, id 61443, offset 0, flags [DF], proto UDP (17), length 326)
    fab2-p1-apic1.cisco.com.50217 > nangaparbat.cisco.com.syslog: [bad udp cksum 0x13c1 -> 0x59dc!] SYSLOG, length: 298
     Facility local7 (23), Severity info (6)
     Msg: <190> Dec 10 18:48:09 rtp-f2-p1-apic1 %LOG_LOCAL7-6-SYSTEM_MSG [E4205038][transition][info][subj-[uni/fabric/slgroup-
deadbeef-syslogGrp/rdst-10.122.254.251]/mod-4294981149] Syslog Remote Destination 10.122.254.251 modified by user admin, change
set: severity (Old: warnings, New: information)\0x0a
     0x0000:  3c31 3930 3e3c 3139 303e 2044 6563 2031
     0x0010:  3020 3138 3a34 383a 3039 2072 7470 2d66
```

# Debugging SYSLOG on APIC

## Some Log Files to search when Troubleshooting

- Access the APIC as "admin" user and search the following log file when troubleshooting the RSYSLOG kernel process on the APIC:
  - cat /var/log/messages
  - cat /var/log/messages | grep rsyslog

For Example:

rtp-f2-p1-apic1:~# **cat /var/log/messages | grep rsyslog**
Dec 10 20:20:24 localhost rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-pid="2081" x-info="http://www.rsyslog.com"] start
Dec 10 20:20:24 localhost rsyslogd-2184: action '1' treated as ':omusrmsg:1' - please change syntax, '1' will not be supported in the future [try http://www.rsyslog.com/e/2184 ]
Dec 10 20:20:24 localhost rsyslogd-3000: invalid character in selector line - ';template' expected

- Access the APIC as "admin" user and search the following log file when troubleshooting the SYSLOG messages on the APIC:
  - cat /var/log/external/messages
  - **tail -f /var/log/external/messages**

For Example:

rtp-f2-p1-apic1:~# **tail -f /var/log/external/messages**
<1027> Dec 10 19:00:55 rtp-f2-p1-apic1 %LOG_LOCAL0-3-SYSTEM_MSG [F0053][deleted][configuration-failed][minor][uni/backupst/jobs-[uni/fabric/configexp-tn-deadbeef-common2]/run-2016-12-08T19-00-01/fault-F0053] Configuration backup/restore job 2016-12-08T19-00-01 failed with error: There is a backup job in progress

<1030> Dec 10 19:00:58 rtp-f2-p1-apic1 %LOG_LOCAL0-6-SYSTEM_MSG [E4204965][backup-finish][info][uni/backupst/jobs-[uni/fabric/configexp-tn-deadbeef-common2]/run-2016-12-10T19-00-52] Configuration import/export job 2016-12-10T19-00-52 finished with status: success

# Troubleshooting the ACI SYSLOG Configuration on the Leaf & Spine nodes.

The following section will give examples on how to troubleshoot the Syslog configuration on the Leaf & Spine nodes. *Note: Some of the following commands may require ROOT access. Temporary "Root" access requires assistance from a Cisco ACI TAC Engineer.*

# Debugging SYSLOG on LEAF\SPINE Nodes

**In addition to the "Show" commands that listed earlier to verify the SYSLOG configuration on Leaf\Spine Nodes, you can use some additional commands to gather more information in regards to SYSLOG. Some of the following commands may require ROOT access. Temporary "Root" access requires assistance from a Cisco ACI TAC Engineer.**

- Additional Commands to run on the leaf or spine prior to accessing ROOT:
  - **show vrf**

    *used to get the "VRF-ID" for "management" & "mgmt:inb". The VRF-IDs are used in reading the iptables.*
  - **show ip route vrf management**
  - **show ip route vrf mgmt:inb**

    *"show ip route vrf" commands are used to verify routes in the management VRFs.*

For Example:

```
rtp-f2-p1-leaf1# show vrf
 VRF-Name                           VRF-ID State    Reason
 management                              2 Up       --
 mgmt:inb                               15 Up       --
```

# Debugging SYSLOG on LEAF\SPINE (cont.)

- On each Leaf or Spine, verify the "logging" processes are running. Record the process ID (pid) for "syslog" processes. You can use one of the following commands:
  - netstat -p | grep syslog
  - ps aux | grep syslog

For Example:
(note: some output has been abbreviated for display purposes)

```
rtp-f2-p1-leaf1# netstat -p | grep syslog
unix  106    [ ]         DGRAM                    80499    6513/svc_ifc_eventm /var/run/mgmt/
syslog_socket


rtp-f2-p1-leaf1# ps aux | grep syslog
root      6427  0.0  0.0   2712   896 ?        Ss   15:55   0:00 /isan/sbin/xinetd -syslog local7
-loop 250 -stayalive -reuse -dontfork


Process IDs for Syslog feature
6513, 6427
```

*Note: Repeat on each Leaf or Spine node having issues with the SYSLOG feature.*

# Debugging SYSLOG on LEAF\SPINE

## "netstat"

- On each Leaf or Spine, gather some network statistics in relation to the "syslog" management interfaces. You use the output to verify the management interfaces are transmitting & recieving packets. You can also verify that the Leaf or Spine node has routes to the SYSLOG server(s). You can use the following commands to gather network status:
  - netstat -ai | grep eth0
  - netstat -ai | grep kpm_inb
  - netstat -nr

For Example:

(note: some output has been abbreviated for display purposes)

```
rtp-f2-p1-leaf1# netstat -ai | grep eth0
Kernel Interface table
Iface   MTU Met   RX-OK  RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500 0     51654      0      0 0          6171      0      0      0 BMRU

rtp-f2-p1-leaf1# netstat -ai | grep kpm_inb
Kernel Interface table
Iface   MTU Met   RX-OK  RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
kpm_inb 9300 0    378212      0      0 0        537802      0    128      0 BMRU

rtp-f2-p1-leaf1# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.122.254.1    0.0.0.0         UG        0 0          0 eth0
10.122.254.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
127.1.0.0       0.0.0.0         255.255.0.0     U         0 0          0 kpm_inb
```

*Note: Repeat on each Leaf or Spine node having issues with the SYSLOG feature.*

# Debugging SYSLOG on LEAF\SPINE

## "iptables"

- On each Leaf or Spine, check the "**iptables**" to see what rules are programmed for SYSLOG . If Syslog is configured to use the inband management EPG, the programming of "iptable" rules for the management inband VRF is necessary to the success of the SYSLOG configuration and deployment to Leaf & Spine nodes. You can use the following commands to check the "iptable" rules:
  - iptables --list | grep syslog
  - iptables -nvL

***Note: Refer to the "show vrf" commands mentioned earlier and repeat on each Leaf or Spine node having issues with the SYSLOG feature.***

For Example:
(note: some output has been abbreviated for display purposes)

```
rtp-f2-p1-leaf1# show vrf
 VRF-Name                           VRF-ID State    Reason
 management                              2 Up       --
 mgmt:inb                               15 Up       --
```

# Debugging SYSLOG on LEAF\SPINE

## "iptables"

For Example: (cont.)

(note: some output has been abbreviated for display purposes)

```
rtp-f2-p1-leaf1# iptables --list | grep syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  49155  udp dpt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  16386  udp dpt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  49155  udp spt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  16386  udp spt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  49154  udp dpt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  32770  udp dpt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  49154  udp spt:syslog
ACCEPT     udp  --  anywhere             anywhere          src-class-id  32770  udp spt:syslog

rtp-f2-p1-leaf1# iptables -nvL
Chain vrf_15_mrules (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  49155  udp dpt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  16386  udp dpt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  49155  udp spt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  16386  udp spt:514

Chain vrf_2_mrules (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  49154  udp dpt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  32770  udp dpt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  49154  udp spt:514
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           src-class-id  32770  udp spt:514
```

**Note: If SYSLOG processes are running and you are not seeing syslog in the IP Tables, run the cli command "acidiag restart mgmt" on the APIC. After restarting the mgmt process on the APIC, check the IP Tables again.**

# Debugging SYSLOG on LEAF\SPINE

## Verify sending SYSLOG messages using “tcpdump”

- Access the Leaf\Spine as "root" user and use "**tcpdump**" command to verify SYSLOG messages are being sent. Use UDP port **514** or any other UDP Ports that are configured for the SYSLOG server destinations in the ACI SYSLOG Monitoring Group. You can use the following "**tcpdump**" commands to check for SYSLOG messages on Leaf\Spine Nodes:
  - tcpdump -i eth6 -f port 514 -vv *(for modular spine)*
  - tcpdump -i eth0 -f port 514 -vv *(for fixed leaf)*
  - tcpdump -i kpm_inb -f port 514 -vv

For Example:

**LEAF (OOB) -> Destination Syslog Server address is 10.122.254.251(nangaparbat.cisco.com)**

```
rtp-f2-p1-leaf1# tcpdump -i eth0 -f port 514 -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:14:36.264209 IP (tos 0x0, ttl 64, id 21432, offset 0, flags [none], proto UDP (17), length 222)
    10.122.254.135.41272 > 10.122.254.251.syslog: [bad udp cksum 814c!] SYSLOG, length: 194
     Facility local7 (23), Severity alert (1)
     Msg: <185> Dec 10 23:14:36 rtp-f2-p1-leaf1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info][sys] sent user message to
syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST Message\0x0a
     0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
     0x0010:  3020 3233 3a31 343a 3336 2072 7470 2d66
     0x0020:  322d 7031 2d6c 6561 6631 2025 4c4f 475f

23:14:39.461508 IP (tos 0x0, ttl 64, id 21506, offset 0, flags [none], proto UDP (17), length 222)
    10.122.254.135.41272 > 10.122.254.251.syslog: [bad udp cksum 7e4c!] SYSLOG, length: 194
     Facility local7 (23), Severity alert (1)
     Msg: <185> Dec 10 23:14:39 rtp-f2-p1-leaf1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info][sys] sent user message to
syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST Message\0x0a
     0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
     0x0010:  3020 3233 3a31 343a 3339 2072 7470 2d66
     0x0020:  322d 7031 2d6c 6561 6631 2025 4c4f 475f
```

# Debugging SYSLOG on LEAF\SPINE

## Verify sending SYSLOG messages using "tcpdump" (cont.)

For Example:

**LEAF (INB) -> Destination Syslog Server address is 10.117.67.30**

```
rtp-f2-p1-leaf1# tcpdump -i kpm_inb  -f port 514 -vv
tcpdump: listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
23:19:10.251677 IP (tos 0x0, ttl 65, id 38801, offset 0, flags [none], proto UDP (17), length 222)
    172.18.242.114.48574 > 10.117.67.30.syslog: [udp sum ok] SYSLOG, length: 194
   Facility local7 (23), Severity alert (1)
   Msg: <185> Dec 10 23:19:10 rtp-f2-p1-leaf1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info]
[sys] sent user message to syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST
Message\0x0a
   0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
   0x0010:  3020 3233 3a31 393a 3130 2072 7470 2d66
   0x0020:  322d 7031 2d6c 6561 6631 2025 4c4f 475f

23:19:11.870008 IP (tos 0x0, ttl 65, id 39109, offset 0, flags [none], proto UDP (17), length 222)
    172.18.242.114.48574 > 10.117.67.30.syslog: [udp sum ok] SYSLOG, length: 194
   Facility local7 (23), Severity alert (1)
   Msg: <185> Dec 10 23:19:11 rtp-f2-p1-leaf1 %LOG_LOCAL7-1-SYSTEM_MSG [E4210472][transition][info]
[sys] sent user message to syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST
Message\0x0a
   0x0000:  3c31 3835 3e3c 3138 353e 2044 6563 2031
   0x0010:  3020 3233 3a31 393a 3131 2072 7470 2d66
   0x0020:  322d 7031 2d6c 6561 6631 2025 4c4f 475f
```

# Debugging SYSLOG on LEAF\SPINE

## Some Log Files to search when Troubleshooting

- Access the APIC as "admin" user and search the following log file when troubleshooting the SYSLOG messages on the APIC:
  - cat /var/log/external/messages
  - **tail -f /var/log/external/messages**

For Example:

```
rtp-f2-p1-leaf1# tail -f /var/log/external/messages

<1030> Dec 10 23:20:13 rtp-f2-p1-leaf1 %LOG_LOCAL0-6-SYSTEM_MSG [login,session][info][subj-[uni/userext/user-admin]/
sess-906238100538] From-127.0.0.1-client-type-ssh-Success

<1025> Dec 10 23:22:28 rtp-f2-p1-leaf1 %LOG_LOCAL0-1-SYSTEM_MSG [E4210472][transition][info][sys] sent user message to
syslog group:deadbeef-syslogGrp:This is a DEADBEEF SYSLOG TEST Message
```

- Access the Leaf\Spine as "admin" user and search some of the following logs when troubleshooting SYSLOG messages on Leaf\Spine Nodes:
  - zgrep "syslog" /var/log/dme/log/*
  - zgrep "syslog" /var/log/dme/log/svc_ifc_policyelem.*
  - zgrep "syslog" /var/log/dme/log/nginx.*
  - zgrep "syslog" /var/log/dme/log/svc_ifc_eventmgr.*
  - zgrep "syslogd_log" /var/log/dme/log/*

*Note: Some of the above commands may or may not produce output when performed on a Leaf or Spine node. These are just some examples which may point you in the right direction.*

# ACI SYSLOG Caveats - Issues

This section will discuss some known caveats or issues with the SYSLOG feature in the ACI Solution.  A few notable Caveats or Issues are:

# ACI SYSLOG Caveats - Issues - Gotchas

When SYSLOG is configured correctly for SYSLOG messaging & feature works as expected. Most of the issues relate to misconfiguration or issues with software programming. The following are some common gotchas that we see and you can use the material in the technote to troubleshoot syslog issues in the ACI Fabric.

- Verify Contract configuration for Management EPGs.
- If you are using SYSLOG ports other than port 514, make sure the non-standard ports are configured in your ACI SYSLOG configuration.
- Currently, ACI only uses UDP for Syslog message transport protocol.
- Facility or Severity mismatch between ACI Devices and Syslog messaging server
- Node Management Address(s) in the Tenant mgmt need to be configured for the APIC(s), Leaf(s), and Spine(s). Verify that the Node management address(s) are configured.
- The ACI Devices (APIC(s), Leaf(s), and Spine(s)) **IP addresses for OOB & INB** need to be added to configuration for allowed inputs on your SYSLOG Monitoring Application.
- Check Firewall configuration on the SYSLOG Monitoring Application Server.
- “iptables” programming on the ACI devices

# ACI SYSLOG Caveats - Issues - Gotchas (cont.)

When SYSLOG is configured correctly for SYSLOG messaging & feature works as expected. Most of the issues relate to misconfiguration or issues with software programming. The following are some know software defects related to Syslog feature which you may run into or unexpected behavior:

- CSCvb77141 ACI: acllog creates duplicate syslogs for single packet
  The two copies of the Syslog message that is seen on the Syslog data collector is to due to the result of two syslog sources (syslogSrc Mo) defined, one under common and one under default, both point to the same syslogDest, which cause the event manager produces two messages. For each defined source, the ACI node will send a Syslog message to the Syslog destination source. This is the expected behavior in current Syslog feature implementation.

- CSCvc00322 [apic syslog] Changes to existing Syslog Remote Destination configuration requires mgmt restart
  In the multiple releases of ACI firmware, there is an issue with deploying Syslog policy changes to existing Syslog Remote Destination configurations. One example of the the issue can be observed when the admin user changes the UDP port# used for the Syslog Remote Destination. The configuration change is accepted but not applied. Syslog messages continue to be sent on the previously configured UDP Port. To force the APIC to use the modified configuration changes, you have to restart the mgmt policies with the CLI command "acidiag restart mgmt".

# References & Resources

# References and Resources

**Reference Links**

- **[1] Using Syslog**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/basic-config/b_ACI_Config_Guide/b_ACI_Config_Guide_chapter_010.html#d2933e4611a1635
- **[2] Cisco ACI System Messages Reference Guide**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci_syslog/About.html
- **[3] Cisco System Messages Management Guide**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors/b_IFC_Faults_Errors_chapter_011.html
- **[4] Cisco APIC Events & Audit logs Management Guide**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors/b_IFC_Faults_Errors_chapter_010.html
- **[5] Cisco APIC Troubleshooting Guide**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/troubleshooting/b_APIC_Troubleshooting/b_APIC_Troubleshooting_chapter_01.html?referring_site=RE&pos=1&page=http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors/b_IFC_Faults_Errors_chapter_011.html

  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/troubleshooting/b_APIC_Troubleshooting/b_APIC_Troubleshooting_chapter_01.html?referring_site=RE&pos=1&page=http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors/b_IFC_Faults_Errors_chapter_011.html#id_37578

- **[6] Proactive Monitoring - Tenant and Fabric Policies**
  http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/Operating_ACI/guide/b_Cisco_Operating_ACI/b_Cisco_Operating_ACI_chapter_01011.html?referring_site=RE&pos=5&page=http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/faults/guide/b_APIC_Faults_Errors/b_IFC_Faults_Errors_chapter_011.html

# References and Resources (cont.)

**VISORE Class or DN**

- (syslogGroup, syslogRemoteDest, syslogProf, syslogRtDestGroup, syslogSrc, syslogFacilityFilter)
- (mgmtSubnet, mgmtRsOoBCons, vzOOBBrCP, vzEntry)

**APIC CLI "Show" Commands**

- show running-config logging
- show running-config logging server-group <syslog destination group>
- show running-config syslog
- show running-config syslog common

**LEAF\SPINE CLI Commands**

- cat /mit/uni/fabric/slgroup-syslogGroup-NAME/summary
- ls /mit/uni/fabric/slgroup-syslogGroup-NAME/rdst* | grep "rdst"
- cat /mit/uni/fabric/monfab-default/slsrc-syslogSource-NAME/summary
- cat /mit/uni/fabric/moncommon/slsrc-syslogSource-NAME/summary
- cat /mit/uni/infra/moninfra-default/slsrc-syslogSource-NAME/summary

# Review Questions

**1. Which of the following can trigger the APIC\LEAF\SPINE to send a system log (SYSLOG) message to the console and, optionally, to a logging server on another system? (Choose all that apply)**

a. Event
b. Upgrade
c. Fault
d. SNMP read queries (Get, Next, Bulk, Walk)
e. All of the above are triggers for sending system log (SYSLOG) messages

**2. Fault-generated system log (SYSLOG) messages are triggered by these mechanisms: (Choose all that apply)**

a. A failure of a task or finite state machine (FSM) sequence.
b. A threshold crossing.
c. A fault rule
d. All of the above are triggers for fault-generated system log (SYSLOG) messages in ACI.

# Review Questions

**3. Event-generated system log (SYSLOG) messages are triggered by these mechanisms: (Choose all that apply)**

a. A failure of a task or finite state machine (FSM) sequence.
b. An event in the NX-OS operating system of a leaf or spine switch
c. A threshold crossing.
d. An event rule
e. All of the above are triggers for event-generated system log (SYSLOG) messages in ACI.

**4. Which single ACI Fabric Monitoring policy can be configured to use an SYSLOG Source that will be applied to both fabric and access infrastructure hierarchies:**

a. Fabric Policies -> Default Policy "monFabricPol (uni/fabric/monfab-default)"
b. Access Policies -> Default Policy "monInfraPol (uni/infra/monifra-default)"
c. Fabric Policies -> Common Policy "monCommonPol (uni/fabric/moncommon)"
d. Tenant -> EPG Policy "monEPGPol (uni/tn-common/monepg-default)"
e. None of the above are correct

# Review Questions

**5. Which Severity level for the "default" facility filter is necessary to record %ACLLOG-5-ACLLOG_PKTLOG messages in SYSLOG: (Choose all that apply)**

a. alert
b. critical
c. error
d. warning
e. notification
f. informational
g. debugging

# Review Questions (Answer Key)

**1.** a, c

**2.** d

**3.** b, d

**4.** c

**5.** f