ESC System Logs

ESC System Logs

Log messages are created for ESC events throughout the VNF lifecycle. These can be external messages, messages from ESC to other external systems, error messages, warnings, events, failures and so on. The log file can be found at `/var/log/esc/escmanager_tagged.log`.

The log message format is as follows:

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

Sample log is as follows:

```
date=15:43:58,46022-Nov-2016]
[loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7] [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:CSCvd94541,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0] [msg=sleepingfor5seconds
to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

When a request is received, a RequestDetails object is created which autogenerates a unique transaction id. This value is carried forward across all threads. Classifications and tags are optional. These are prefixes added to the log messages to enhance readability, and help in debugging. With classifications and tags, the log messages can be easily parsed and filtered by the log analysis tools.

The following classifications are supported:

NBI	"com.cisco.esc.rest""com.cisco.esc.filter"(North Bound Interface - Clientinterface)
SBI	"com.cisco.esc.rest"- source is a callback handler or"EventsResource"(South Bound Interface - i.e. between ESC and the VIM)
SM	"com.cisco.esc.statemachines". stands for StateMachine. This classification indicates logs in the StateMachine category.

MONITORING	"com.cisco.esc.monitoring""com.cisco.esc.paadaptor"(MONA related logs)
DYNAMIC_MAPPING	"com.cisco.esc.dynamicmapping""com.cisco.esc.db.dynamicmapping"(MONA related logs)
CONFD	"com.cisco.esc.confd"
CONFD_NOTIFICATION	"com.cisco.esc.confd.notif""com.cisco.esc.confd.ConfdNBIAdapter"
OS	"com.cisco.esc.vim.openstack"
LIBVIRT	"com.cisco.esc.vim.vagrant
VIM	"com.cisco.esc.vim"
REST_EVENT	"ESCManager_Event""com.cisco.esc.util.RestUtils". indicates REST notifications in logs.
WD	"com.cisco.esc.watchdog"
DM	"com.cisco.esc.datamodel""com.cisco.esc.jaxb.parameters"(Datamodel and resource objects)
DB	"com.cisco.esc.db"(Database related logs)
GW	"com.cisco.esc.gateway"
LC	"com.cisco.esc.ESCManager"(Start up related logs)
SEC	"com.cisco.esc.jaas"
MOCONFIG	"com.cisco.esc.moconfig"(MOCONFIG object related logs --thisis internalforESC developers)
POLICY	"com.cisco.esc.policy"(Service/VM Policy related logs)
TP	"com.cisco.esc.threadpool"
ESC	"com.cisco.esc" Any other packages not mentioned above

The following tags are supported:

- **Workflow [wf:]**—Generated using action and resource from RequestDetails object. Example "wf:create_network"
- **Event type [eventType:]**—Event that triggered the current action. Example: "eventType:VM_DEPLOY_EVENT"
- **Resource based**—These values are generated based on the type of parameter used by the event. The hierarchy, that is, the tenant, the vm group and so on is added to the log.

Tenant	[tenant:<tenant name>]
Network	[tenant:<tenant id>, network:<network name>] **Note** The tenant appears only if applicable.

Subnet	[tenant:<tenant name or id>, network:<network name or id>, subnet:<subnet name>] **Note** The tenant appears only if applicable.
User	[tenant:<tenant name>, user:<user name or id>] **Note** The tenant appears only if applicable.
Image	[image:<image name>]
Flavor	[flavor:<flavor name>]
Deployment	[tenant:<tenant name or id>, depName:<deployment name>]
DeploymentDetails	[tenant:<tenant name or id>, depName:<deployment name>, vmGroup:<vm group name>, vmName:<vm name>]
Switch	[tenant:<tenant name or id>, switch:<switch name>]
Volume	[volume:<volume name>]
Service	[svcName:<Service Registration name>]

Further, ESC logs can also be forwarded to an rsyslog server for further analysis and log management.

Filtering Logs Using Confd APIs

You can query and retrieve logs (for example, deployment logs, or error logs) in ESC using log filters introduced in the confd APIs. New filters for Tenant, Deployment Name, and VM Name are introduced. This enables you to query the ESC logs further for most recent error logs using the log filters in Confd APIs. You can also retrieve ESC logs related to the communication between ESC and the OS (by setting the classification tag to "OS").

The log format to retrieve confd API logs:

```
date=<time-date>] [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

The sample log is as follows:

```
date=15:43:58,46022-Nov-2016] [loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7]
 [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[msg=sleepingfor5seconds to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7 name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

The parameters for log level, classification and tags are dependent on each other to retrieve the logs. You can successfully retrieve the logs with the following combination.

- log_level=ERROR, classifications=OS, tags=(depName:test-dep)
- log_level=ERROR, classifications=OS, tags=(tenant: test)

The log filter returns a value when all the following conditions are met:

- Log level

- Classifications (if provided)
- Tags (if provided)

Note If there are more than one classification listed, it has to match at least one of the classifications. The same applies to the tags as well.

For example, the following log filter criteria does not return the log sample mentioned earlier:

```
log_level=ERROR, classifications=VIM, tags=(depName:test-dep)
```

It does not return any value though the log level and tags match, the classification VIM does not match.

The data model is as follows:

```
rpc filterLog {
    description "Query and filter escmanager logs using given parameters";
    tailf:actionpoint escrpc;
    input {
      leaf log_level {
        mandatory false;
        description "One of DEBUG / INFO / WARNING / ERROR / TRACE / FATAL. Results will
include all logs at and
                     above the level specified";
        type types:log_level_types;
        default ERROR;
      }
      leaf log_count {
        mandatory false;
        description "Number of logs to return";
        type uint32;
        default 10;
      }
      container classifications {
        leaf-list classification {
         description "Classification values to be used for the log filtering.  For example:
 'OS', 'SM'.
                      Logs containing any of the provided classification values will be
returned.";
          type types:log_classification_types;
        }
      }
      container tags {
        list tag {
          key "name";
          leaf name {
            mandatory true;
           description "Tag name to be used for the log filtering.  For example: 'tenant',
 'depName'.
                        Logs containing any of the provided tag name plus the tag values
will be returned.";
            type types:log_tag_types;
          }
          leaf value {
            mandatory true;
            description "Tag value pairs to be used for the log filtering.  For example:
'adminTenant', 'CSRDeployment'";
            type string;
          }
        }
      }
```

```
  }
  output {
    container filterLogResults {
      leaf log_level {
        description "Log level used to filter for the logs.";
        type types:log_level_types;
      }
      list logs {
        container classifications {
          leaf-list classification {
            description "Classifications used to filter for the logs.";
            type types:log_classification_types;
          }
        }
        container tags {
          list tag {
            key "name";
            leaf name {
              mandatory true;
              description "Tag name used to filter for the logs.";
              type types:log_tag_types;
            }
            leaf value {
              mandatory true;
              description "Tag value used to filter for the logs.";
              type string;
            }
          }
        }
        leaf log_date_time {
          description "Timestamp of the log.";
          type string;
        }
        leaf log_message {
          description "The log message.";
          type string;
        }
      }
    }
  }
}
```

You can query for the confd API logs through the netconf console or esc_nc_cli

- Through the netconf-console, run the following query:

  ```
  /opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin
  --privKeyFile=/home/admin/.ssh/confd_id_dsa --privKeyType=dsa --rpc=log.xml
  ```

- Using the esc_nc_cli, run the following query:

  ```
  ./esc_nc_cli filter-log log.xml
  ```

  The sample log.xml is as follows:

  ```
  <filterLog xmlns="http://www.cisco.com/esc/esc">
    <log_level>INFO</log_level>
    <log_count>1</log_count>
    <classifications>
      <classification>OS</classification>
      <classification>SM</classification>
    </classifications>
    <tags>
      <tag>
        <name>depName</name>
  ```

```
      <value>CSR_ap1</value>
    </tag>
    <tag>
      <name>tenant</name>
      <value>admin</value>
    </tag>
  </tags>
</filterLog>
```

The response is as follows:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <filterLogResults xmlns="http://www.cisco.com/esc/esc">
    <log_level>INFO</log_level>
    <logs>
      <classifications>
        <classification>OS</classification>
        <classification>SM</classification>
      </classifications>
      <tags>
        <tag>
          <name>depName</name>
          <value>CSR_ap1</value>
        </tag>
        <tag>
          <name>tenant</name>
          <value>admin</value>
        </tag>
      </tags>
      <log_date_time>13:06:07,575 31-Oct-2016</log_date_time>
      <log_message> No pending work flow to start.</log_message>
    </logs>
  </filterLogResults>
</rpc-reply>
```

Note The logging API responses are in XML format. If the log messages contain any XML characters, then the characters will be escaped so not to break the XML conformance.

Viewing ESC Log Files

You can find the logs of various ESC components here:

File	Component	Description	Rotation Size	Number of backup files
/var/log/esc/escmanager.log	ESCManager	This contains logs of the ESC Manager which includes workflow, request and persistence.	150 MB	10

File	Component	Description	Rotation Size	Number of backup files
/var/log/esc/error_escmanager.log	ESCManager	This is the same as escmanager.log but with a format that is easier to read for netconf logging API but can be used by other parsers if needed.	150 MB	10
/var/log/esc/event_escmanager.log	ESCManager		150 MB	10
/var/log/esc/yangesc.log	ESCManager	This contains logs related to our netconf request and notifications	150 MB	10
/var/log/esc/debug_yangesc.log	ESCManager		51MB	2
/var/log/esc/mona/mona.log	MONA		150 MB	10
/var/log/esc/mona/actions_mona.log	MONA		150 MB	10
/var/log/esc/mona/rules_mona.log	MONA		150 MB	10
/var/log/esc/vimmanager/vimmanager.log	VIM Manager Service	A detailed VIM manager log.	150 MB	10
/var/log/esc/vimmanager/operations_vimmanager.log	VIM Manager Service	A simplified log which only lists the VIM Manager operations been processed.	150 MB	10
/var/log/esc/vimmanager/vim_vimmanager.log	VIM Manager Service	Raw HTTP request/response (including header) between VIM Manager and VIM. Note, for OpenStrack to track this log, log level has to set to DEBUG for VIM. Manager.	150 MB	10
/var/log/esc/<timestamp>-esc-portal-be.log	ESC Portal		10 MB	4
/var/log/esc/confd/audit.log	confd		10 MB	4
/var/log/esc/confd/browser.log	confd		10 MB	4
/var/log/esc/confd/confd.log	confd		10 MB	4

File	Component	Description	Rotation Size	Number of backup files
/var/log/esc/confd/devel.log	confd		10 MB	4
/var/log/esc/confd/netconf.log	confd		10 MB	4
/var/log/esc/confd/netconf.trace	confd		10 MB	4
/var/log/esc/confd/cli-history/admin.hist				
/var/log/esc/confd/cli-history/global.data				
`/var/log/esc/esc_haagent.log`	ESC INFRA or HA		10 MB	4
/var/log/esc/ esc_monitor.log	ESC INFRA or HA		10 MB	4
/var/log/esc/ esc_monitor_output.log	ESC INFRA or HA		10 MB	4
/var/log/esc/esc_confd.log	ESC INFRA or HA		10 MB	4
/var/log/esc/ pgstartup.log	ESC INFRA or HA		10 MB	4
/var/log/esc/spy.log	ESC INFRA or HA		No logs (size 0)	No ESC generated logs.
/var/log/tomcat6/catalina.out	Tomcat		Not rated	No ESC generated logs. Only Error.
/var/log/esc/esc_dbtool.log				
/var/log/esc/snmp/snmp.log				
/var/log/esc/etsi-vnfm/etsi-vnfm.log	ETSI-Service	This is the main log file for ETSI processing, including requests, response, payloads and general logging information where appropriate.	150MB	10

File	Component	Description	Rotation Size	Number of backup files
/var/log/esc/etsi-vnfm/events-etsi-vnfm.log	ETSI-Service	Logs only API requests, both entering and leaving.	150MB	10
/var/log/esc/etsi-vnfm/event-details-etsi-vnfm.log	ETSI-Service	Logs both the API requests (entering and leaving) along with the actual JSON payloads.	150MB	10
/var/log/esc/escadm.log	escadm	Logs to capture both manual and automated messages and errors from escadm.py. This log is useful to track startup and configuration changes to ESC.		