Using Logstash with EPICS IOCs

Andrew Johnson
anj@anl.gov

Controls Group,
APS Engineering Services Division,
Argonne National Laboratory

Logstash for IOC Error Logging

- EPICS applications can log errors to a central logging service
- The libCom errlog.h Log Client routines send messages in plain text to a configurable TCP socket (host:port)
- Logstash is a commonly used Open Source logging tool, part of ELK (ElasticSearch)
  - Other logging tools can be used instead, configured appropriately
- Logstash has many input plugins that support different ways of receiving and parsing log messages
- Here's how to configure Logstash to accept log messages from one or more IOCs:

errlog-pipeline.conf

```
# Simplest possible logstash configuration file

# Start using:
#   logstash -f errlog-pipeline.conf

input {
    tcp {
        host => "${EPICS_IOC_LOG_INET:loghost}"
        port => "${EPICS_IOC_LOG_PORT:7004}"
        codec => "line"
    }
}
output {
    file {
        path => "/log/ioc/errlog-%{+YYYY-MM-dd}.json"
    }
}
```

Configuring IOCs

- IOC st.cmd file for errlog

```
# Configure the errlog client
epicsEnvSet EPICS_IOC_LOG_INET loghost
epicsEnvSet EPICS_IOC_LOG_PORT 7004
iocLogInit
```

- Bonus: Configuring for caPutLog

```
# Configure and start the caPutLogger after iocInit
epicsEnvSet EPICS_AS_PUT_LOG_PV "${IOC}:caPutLatest"
caPutLogInit "loghost:7011"
```

caPutLog-pipeline.conf

```
# logstash configuration for caPutLog

input {
    tcp {
        host => "${EPICS_IOC_LOG_INET:loghost}"
        port => "${EPICS_IOC_LOG_PORT:7011}"
        codec => "line"
    }
}
filter {
    # see next slide...
}
output {
    file {
        path => "/log/ioc/putlog-%{+YYYY-MM-dd}.json"
    }
}
```

caPutLog message filters

```
filter {
  grok {
    pattern_definitions => {
      MONTHNAM => "(?:JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)"
      TIMESTAMP => "%{MONTHDAY}-%{MONTHNAM}-%{YEAR} %{TIME}"
      PV => "\S+"
    }
    named_captures_only => true
    # The following string must be all one line, one space between parts
    message => "%{TIMESTAMP:timestamp} %{HOSTNAME:client} %{USER:user}
      %{PV:pv} new=%{NUMBER:new} old=%{NUMBER:old}(?: min=%{NUMBER:min}
      max=%{NUMBER:max})?"
  }
  date {
    # Extract the timestamp from the message as a proper logstash
    # timestamp field type, but since the IOC's clock might not be
    # accurate we don't replace the locally generated @timestamp.
    match => ["timestamp", "dd-MMM-yy HH:mm:ss"]
    locale => "en"
    target => "@ioctimestamp"
  }
}
```