Logging Architecture

Artifacts

JIRAs | Git (pending) | Gerrit (pending) | Nexus | Jenkins (pending) | Sonar |

High Level Description

There are the 3 elk containers in the log pod in kubernetes off the OOM deployment. Currently all 3 ports are exposed, logstash, elasticsearch and kibana. We use Kubernetes 1.8 with Helm 2.8 charts via the OOM project deployment. The ports are all mapped as node ports in the external 302xx namespace. We use filebeat as a side car container for all the onap components to push logs to the logstash node port.

Logstash port mapping

https://git.onap.org/oom/tree/kubernetes/log/charts/log-logstash/templates/service.yaml

https://git.onap.org/oom/tree/kubernetes/log/charts/log-logstash/templates/deployment.yaml

port numbers are here - 5044

https://git.onap.org/oom/tree/kubernetes/log/charts/log-logstash/values.yaml

for example SDC filebeat config to log-ls on port 5044

https://git.onap.org/oom/tree/kubernetes/sdc/values.yaml

the filebeat sidecar docker image for sdc-be

https://git.onap.org/oom/tree/kubernetes/sdc/charts/sdc-be/values.yaml

Logging configmap

https://git.onap.org/oom/tree/kubernetes/sdc/charts/sdc-be/templates/configmap.yaml

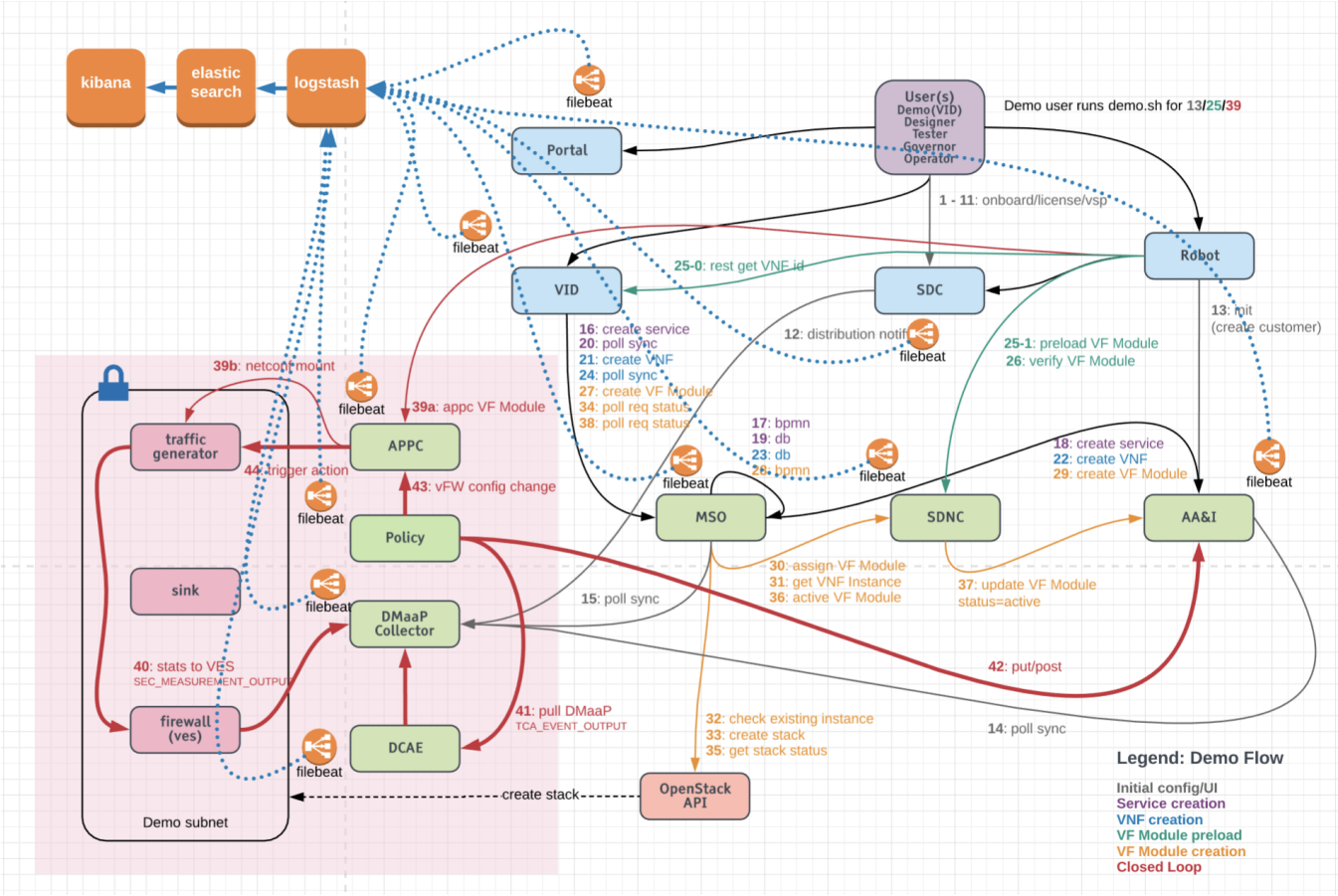
Logging filebeat deployment spec

https://git.onap.org/oom/tree/kubernetes/sdc/charts/sdc-be/templates/deployment.yaml

and the logback config

https://git.onap.org/oom/tree/kubernetes/sdc/charts/sdc-be/resources/config/logging/logback.xml

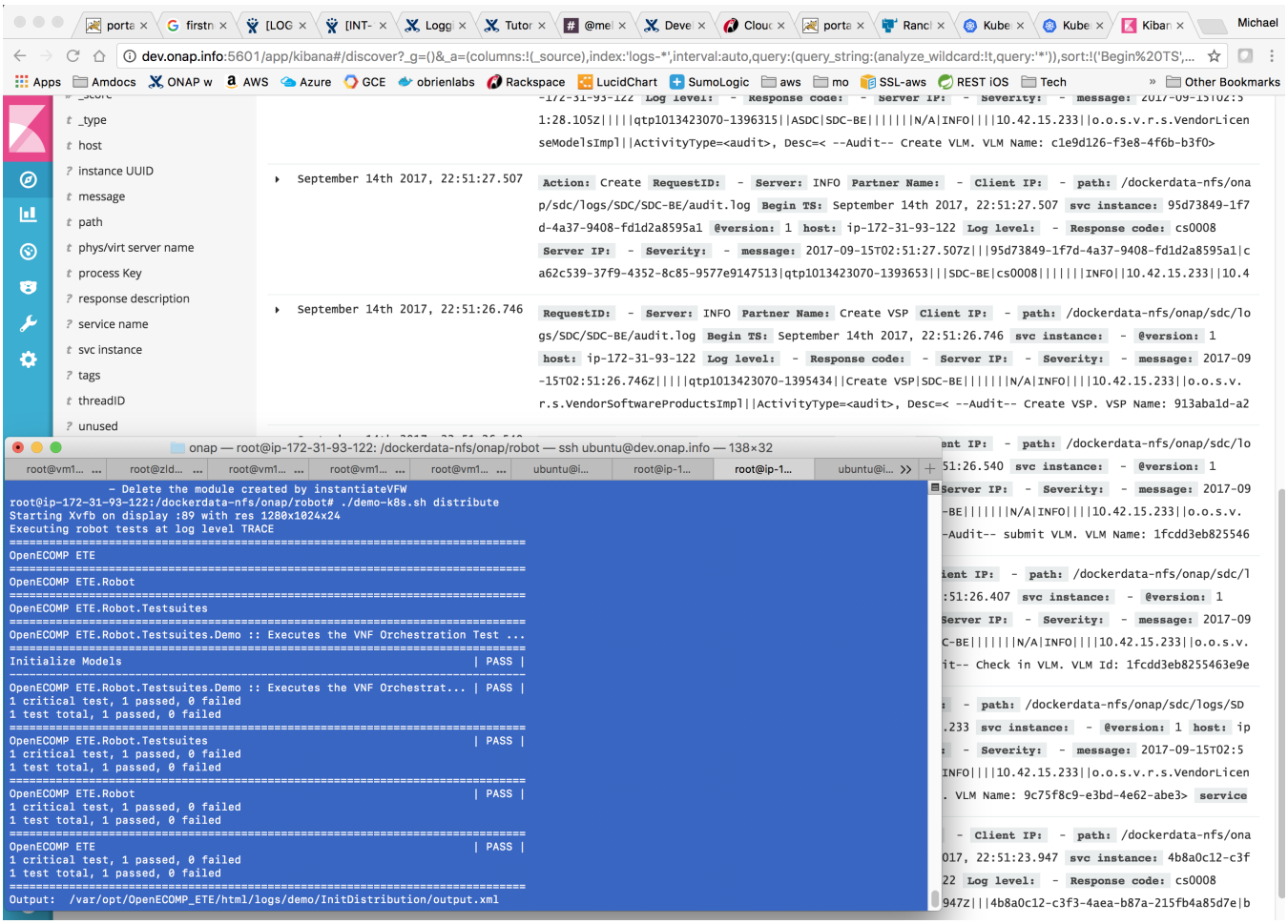
ELK stack in ONAP



Use Cases

Use Case 1: SDC distribution

We can distribute manually via SDC or automatically using robot ./demo.sh distribute



Use Case 2: Triaging a failed Service Deployment

Architecture Discussion

Log Generation

Log Collection

Log Processing

Log Analytics

Design Issues

DI 1: 20170803: Cross Project Collaboration

AAI

https://git.onap.org/aai/logging-service/

Audit and Metrics log formats are different

```
audit.log
  return startTimeString + "|" +                                  // 1 start time
      endTimeString + "|" +                                       // 2 end time
      getMdcValue(MdcContext.MDC_REQUEST_ID) + "|" +              // 3 transaction id
      getMdcValue(MdcContext.MDC_SERVICE_INSTANCE_ID) + "|" +     // 4 service instance
      Thread.currentThread().getName() + "|" +                    // 5 thread id
      getMdcValue(MdcContext.MDC_SERVER_FQDN) + "|" +             // 6 physical/virtual server name
      getMdcValue(MdcContext.MDC_SERVICE_NAME) + "|" +            // 7 service name
      getMdcValue(MdcContext.MDC_PARTNER_NAME) + "|" +            // 8 partner name
      fieldValue(DefinedFields.STATUS_CODE) + "|" +               // 9 status code
      fieldValue(DefinedFields.RESPONSE_CODE) + "|" +             // 10 response code
      fieldValue(DefinedFields.RESPONSE_DESCRIPTION) + "|" +      // 11 response description
      fieldValue(DefinedFields.INSTANCE_UUID) + "|" +             // 12 instance UUID
      level + "|" +                                               // 13 log level
      fieldValue(DefinedFields.SEVERITY) + "|" +                  // 14 log severity
      fieldValue(DefinedFields.SERVER_IP) + "|" +                 // 15 server ip
      elapsedTimeString + "|" +                                   // 16 elapsed time
      getMdcValue(MdcContext.MDC_SERVER_FQDN) + "|" +             // 17 server name
      getMdcValue(MdcContext.MDC_CLIENT_ADDRESS) + "|" +          // 18 client ip address
      fieldValue(DefinedFields.CLASS_NAME) + "|" +                // 19 class name
      "" + "|" +                                                  // 20 deprecated
      fieldValue(DefinedFields.PROCESS_KEY) + "|" +               // 21 process key
      fieldValue(DefinedFields.CUSTOM_1) + "|" +                  // 22 custom 1
      fieldValue(DefinedFields.CUSTOM_2) + "|" +                  // 23 custom 2
      fieldValue(DefinedFields.CUSTOM_3) + "|" +                  // 24 custom 3
      fieldValue(DefinedFields.CUSTOM_4) + "|" +                  // 25 custom 4
      message;                                                    // 26 details
}


metrics.log
  return startTimeString + "|" +                                  // 1 start time
      endTimeString + "|" +                                       // 2 end time
      getMdcValue(MdcContext.MDC_REQUEST_ID) + "|" +              // 3 transaction id
      getMdcValue(MdcContext.MDC_SERVICE_INSTANCE_ID) + "|" +     // 4 service instance
      Thread.currentThread().getName() + "|" +                    // 5 thread id
      getMdcValue(MdcContext.MDC_SERVER_FQDN) + "|" +             // 6 physical/virtual server name
      getMdcValue(MdcContext.MDC_SERVICE_NAME) + "|" +            // 7 service name
      getMdcValue(MdcContext.MDC_PARTNER_NAME) + "|" +            // 8 partner name
      fieldValue(DefinedFields.TARGET_ENTITY) + "|" +             // 9 target entity
      fieldValue(DefinedFields.TARGET_SVC_NAME) + "|" +           // 10 target service
      fieldValue(DefinedFields.STATUS_CODE) + "|" +               // 11 status code
      fieldValue(DefinedFields.RESPONSE_CODE) + "|" +             // 12 response code
      fieldValue(DefinedFields.RESPONSE_DESCRIPTION) + "|" +      // 13 response description
      fieldValue(DefinedFields.INSTANCE_UUID) + "|" +             // 14 instance UUID
      level + "|" +                                               // 15 log level
      fieldValue(DefinedFields.SEVERITY) + "|" +                  // 16 log severity
      fieldValue(DefinedFields.SERVER_IP) + "|" +                 // 17 server ip
      elapsedTimeString + "|" +                                   // 18 elapsed time
      getMdcValue(MdcContext.MDC_SERVER_FQDN) + "|" +             // 19 server name
      fieldValue(DefinedFields.CLIENT_IP) + "|" +                 // 20 client ip address
      fieldValue(DefinedFields.CLASS_NAME) + "|" +                // 21 class name
      "" + "|" +                                                  // 22 deprecated
      fieldValue(DefinedFields.PROCESS_KEY) + "|" +               // 23 process key
      fieldValue(DefinedFields.TARGET_ENTITY) + "|" +             // 24 target virtual entity
      fieldValue(DefinedFields.CUSTOM_1) + "|" +                  // 25 custom 1
      fieldValue(DefinedFields.CUSTOM_2) + "|" +                  // 26 custom 2
      fieldValue(DefinedFields.CUSTOM_3) + "|" +                  // 27 custom 3
      fieldValue(DefinedFields.CUSTOM_4) + "|" +                  // 28 custom 4
      message;                                                    // 29 detail message
}
```

DCAE

1.0.0 (deprecated) - https://git.onap.org/dcae/operation/utils/

DMaaP

DMaaP API

https://wiki.onap.org/download/attachments/11928249/DR-R1-Pub-Del-API-v1.5.pdf?version=2&modificationDate=1501879648000&api=v2

DI 2: 20170929: Volumetrics

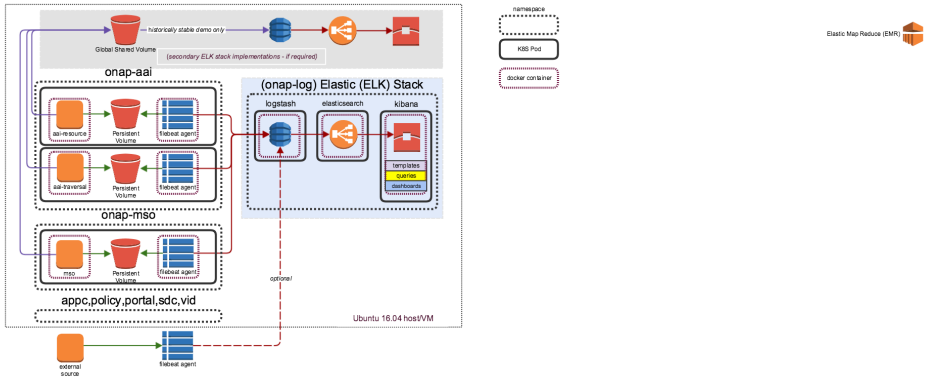
Per component traffic in a real set of use cases over time (# of requests, volume of requests, rate of requests)

DI 3: 20171010: AAI Logging API Debug

```
public class PreAaiAjscInterceptor implements AjscInterceptor {

        public boolean allowOrReject(HttpServletRequest req, HttpServletResponse resp, Map<?, ?> paramMap)


                LoggingContext.requestId(req.getHeader("X-TransactionId"));

                LoggingContext.partnerName(req.getHeader("X-FromAppId"));


public class AAIAppServletContextListener implements ServletContextListener {

        private static final EELFLogger LOGGER = EELFManager.getInstance().getLogger
(AAIAppServletContextListener.class.getName());        public void contextInitialized(ServletContextEvent arg0)
{

                System.setProperty("org.openecomp.aai.serverStarted", "false");

                LOGGER.info("***AAI Server initialization started...");
```

DevOps

OOM Deployment



Southbound APIs

Northbound APIs

https://github.com/opentracing

http://zipkin.io/

https://www.cncf.io/blog/2016/10/11/opentracing-joins-the-cloud-native-computing-foundation/

https://github.com/jaegertracing/jaeger

Discussions

Re: Carrier Grade Requirements (consolidated)

Links

Reference ELK stack - **LOG-50** - Getting issue details... STATUS

https://kubernetes.io/docs/concepts/cluster-administration/logging/

Operate