ONAP CDS – Error Handling in Production Deployments

Aarna Networks - Kavitha P
Aarna Networks - Vivekanandhan M
Agenda

› Problem statement
› ONAP CDS Audit logging feature contribution Jira tickets
› CDS Workflow Audit Logging Design
› CDS Workflow Audit Logging REST APIs
› Demo
Problem Statement

› CDS does not persist the workflow execution status.
› Most practical deployments have CDS to execute long running workflows, where workflow status often needs to be accessed in order to trigger workflow resubmit etc.
› Need to store the complete CDS workflow execution status for audit purpose in production.
CDS Audit logging and ELK Integration features

› We contributed the following ONAP Jira tickets
› CDS Audit Log persistence
   › CCSDK-3434 (CDS Blueprint Processor Changes)
   › CCSDK-3435 (CDS py-executor log persistence)
› CDS Helm chart changes for ELK integration
   › OOM-2801
   › OOM-2802
   › OOM-2803
   › CDS BP flag to enable/disable audit logging
CDS Workflow Auditing Sequence Diagram

RESTAPI

ExecutionServiceController

ExecutionServiceHandler

doProcess

storeExecutionInput

StoreAuditService

BlueprintWorkflowAuditStatus

storeExecutionOutput

MariaDB

BLUEPRINT_WORKFLOW_AUDIT_STATUS

JSON

JSON
CDS Workflow Auditing Data Model

- Introduced a New Database Table

```sql
desc blueprint_workflow_audit_status;
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow_audit_id</td>
<td>bigint(20)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
<tr>
<td>blueprint_name</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>blueprint_uuid</td>
<td>varchar(255)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>blueprint_version</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>end_time</td>
<td>datetime</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>originator_Id</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>request_Id</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>request_mode</td>
<td>varchar(255)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>start_time</td>
<td>datetime</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>varchar(255)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>subRequest_Id</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>updated_by</td>
<td>varchar(255)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>updated_date</td>
<td>datetime</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>workflow_name</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>workflow_response_content</td>
<td>longtext</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>workflow_task_content</td>
<td>longtext</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>
CDS Workflow Audit Logging REST API

- New REST API to get the WorkflowController health check
  
  ```
  curl -v --location \
  --request GET \n  http://{{host}}:{{port}}/api/v1/workflow/health-check \n  --header 'Content-Type: application/json;charset=UTF-8' \n  --header 'Accept: application/json;charset=UTF-8,application/json' \n  --header 'Authorization: Basic Y2NzZGthcHBzOmNjc2RrYXBwcw=='
  ```

- New REST API to get the CDS workflow status
  
  ```
  curl -v --location \
  --request GET \n  http://{{host}}:{{port}}/api/v1/workflow/audit-status/{{requestId}}/{{subRequestId}} \n  --header 'Content-Type: application/json;charset=UTF-8' \n  --header 'Accept: application/json;charset=UTF-8,application/json' \n  --header 'Authorization: Basic Y2NzZGthcHBzOmNjc2RrYXBwcw=='
  ```
Demo - Recorded Video
Thank You!