APPC - CDT tool and R3 Planning

By Takamune (Taka) Cho – AT&T
2018-3-26
Agenda

- APPC recap
- APPC - CDT (Configuration Design Tools)
- APPC - Casablanca Pre-Planning
APPC in R2 Architecture
APPC – Architecture Overview

[Diagram showing the APPC architecture with various components such as SDC, DCAE, SO, Portal, TechOps, DMaaP, Northbound REST Layer, Southbound Layer, Cross Cutting Components, and APPC Database.]

The diagram illustrates the flow of information and processes within the APPC architecture, detailing how different services and components interact with each other, focusing on the key elements such as the Service Logic Interpreter for DGs and Execute Nodes, DMaaP, and the integration with the ONAP platform through various adapters and providers.
APPCC Commands Supported (R2)

**Configuration**

Configure  ConfigModify  ConfigBackup  ConfigRestore  ConfigScaleOut

**OpenStack Commands**

Evacuate  Migrate  Rebuild  Restart  Snapshot  Start  Stop  AttachVolume  DetachVolume

**Health and Diagnostic**

HealthCheck

**Change Management Workflows (Such As Software Upgrade)**

QuiesceTraffic  ResumeTraffic  StartApplication  StopApplication  UpgradeSoftware  UpgradePrecheck  UpgradePostcheck  UpgradeBackup  UpgradeBackout

**Other**

Audit  Sync  CheckLock  Lock  Unlock
Common API For All LCM Commands

VNF Agnostic

DMaaP message bus support using APP-C provided client library

Synchronous and asynchronous responses

API call contains:

- Request / Sub-request id
- Action (single)
- Action identifier(s) [vnf-id, vserver-id, vnfc-name]
- Payload field [contains all instance specific parameter data in name / value for]

```json
{
  "input": {
    "common-header": {
      "timestamp": "2016-08-03T08:50:18.97Z",
      "api-ver": "3.00",
      "originator-id": "SO",
      "request-id": "123456abcd"
    },
    "action": "Configure",
    "action-identifiers": {
      "vnf-id": "ibcx0001"
    },
    "payload": {
      "request-parameters": {
        "vnf-host-ip-address": "135.21.166.39",
        "vnfc-type": "vSBC – mmc"
      },
      "configuration-parameters": {
        "serv1_gateway_ip_address": "192.168.30.44"
      }
    }
  }
}
```
APPC Self Service Approach

• VNF owners onboard new VNF’s using self-service

• APPC provides a Design Tool environment for:
  o Defining the VNF and actions supported
  o Creating the action-specific templates and parameter definitions
  o Testing the actions on a test VNF instance in Stage/ETE environment

• APPC team provides a self-service guide
APPC Use of Templates

- APPC receives all instance specific parameter names/values via a GUI using spreadsheet input
- The configuration parameters are stored in MySQL as a data block
- The template and parameters are sent to the Velocity Template Generator to generate the request data block
- The request is downloaded to the VNF
Supports VNF Self-Service Onboarding For All API Actions
- Create/validate models (artifacts)
- Store/retrieve artifacts in APPC self-service onboarding instance
APP-C Templates

Templates + Name/Value Data:

- The template contains the payload that is delivered to the VNF or Chef/Ansible server for many actions.
- The template is a combination of static parameters and instance specific / changeable parameters.
- The template contains imbedded commands needed to insert instance specific or changeable values at run time (i.e., name/value pair data).
- Other programming constructs can be added as needed (e.g., variable lists, ...)

```
<configuration xmlns="http://xml.juniper.net/xnm/1.1/xnm" >
 <system>
  <nat>
   <proxy-arp>
    <interface>
     <name>reth1.0</name>
     <address>
      <name>${trusted_interface-ip_address}</name>
     </address>
   </interface>
 </proxy-arp>
 </nat>
</configuration>
```

```
{
 "id": "102-49d2-j239",
 "PlaybookName": "comx/latest/ansible/upgrade/site.yml",
 "EnvParameters":{
  "snmp1_server_address": "$address1",
  "snmp2_server_address": "$address2",
  ...
 },
 "NodeList": ["$node1", "$node2", "$node3"],
 "Timeout": 600
}
```
APPCCDT tools

- CDT tools: angular cli/node.js based
- CDT Repo: https://gerrit.onap.org/r/#!/admin/projects/appc/cdt
- DEMO
## Casablanca R3 Planning

<table>
<thead>
<tr>
<th>Level*</th>
<th>Item</th>
<th>Collaborator</th>
</tr>
</thead>
<tbody>
<tr>
<td>likely</td>
<td>Maria DB with galera</td>
<td>CCSDK</td>
</tr>
<tr>
<td>likely</td>
<td>Test coverage cont’ &gt; 50%</td>
<td>contributors</td>
</tr>
<tr>
<td>likely</td>
<td>S3P cont’</td>
<td>contributors</td>
</tr>
<tr>
<td>possible</td>
<td>Geo Redundancy</td>
<td>OOM, INT</td>
</tr>
<tr>
<td>possible</td>
<td>AutoScaleOut usecase cont’</td>
<td>Usecase, AT&amp;T</td>
</tr>
<tr>
<td>reach</td>
<td>5G usecase</td>
<td>Usecase, Nokia</td>
</tr>
<tr>
<td>far reach</td>
<td>Generic Application Controller</td>
<td>Architecture, VFC, TSC</td>
</tr>
</tbody>
</table>

*four levels: likely, possible, reach, far reach
Thank You

PhD Takamune (Taka) Cho
Email: takamune.cho@att.com
LinkedIn: https://www.linkedin.com/in/takacho