

UCA-20 OSS JAX-RS 2 Client

- [Analysis](#)
 - vFirewall Demo Sequence of Events
 - [Status \(current/future\) Flow](#)
 - [Variables](#)
 - [Prerequisites](#)
 - [Config](#)
 - [Service Deployment](#)
 - [VNF Creation](#)
 - [VfModule Preload](#)
 - [VfModule Creation](#)
 - [Closed Loop](#)
 - [API History](#)
- [Design Issues](#)
 - [DI 1: 20170712: AAI Cert required for HTTPS REST calls](#)
 - [Installing the AAI certificate in your own keystore on the Robot VM](#)
 - [DI 2: 20170712: Spring Boot Backend Framework](#)
 - [DI 3: 20170712: JAX-RS 2.0 API Framework](#)
 - [DI 4: 20170712: 3D Framework: Three js OpenGL 3D wrapper on Canvas](#)
 - [DI 5: 20170712: Javascript Timer Framework](#)
 - [DI 6: 20170712: Three js Listener](#)
 - [DI 7: 20170712: Three js Edges between Nodes](#)
 - [DI 8: 20170712: Three js Click Context](#)
 - [DI 9: 20170712: AJAX JAX-RS Channel](#)
 - [Example: getCustomer\(\) from AAI](#)
 - [DI 10: 20170712: WebSocket Channel](#)
 - [DI 11: 20170712: React.js Framework](#)
 - [DI 12: 20170712: Three js Text](#)
 - [DI 13: 20170721: Demo Robot REST extraction](#)
 - [DI 20: 20170721: Rest API Chaining](#)
 - [DI 21: 20170727: Parse VF-Module IPs from endpoint after VF-Module creation](#)
 - [DI 22: 20170728: Determine Healthcheck for vFW VMs are ready for appc Closed-Loop](#)
 - [DI 23: 20170728: Add granular Policy healthcheck for PDBD/PAP/PDP](#)
 - [DI 24: 20170728: Add TCA topic event capture between Policy and APPC](#)
 - [DI 25: 20170806: Add jcloud API to query VMs via Nova](#)
 - [DI 26: 20170806: Add docker ps parse calls to get docker health](#)
- [Notes](#)



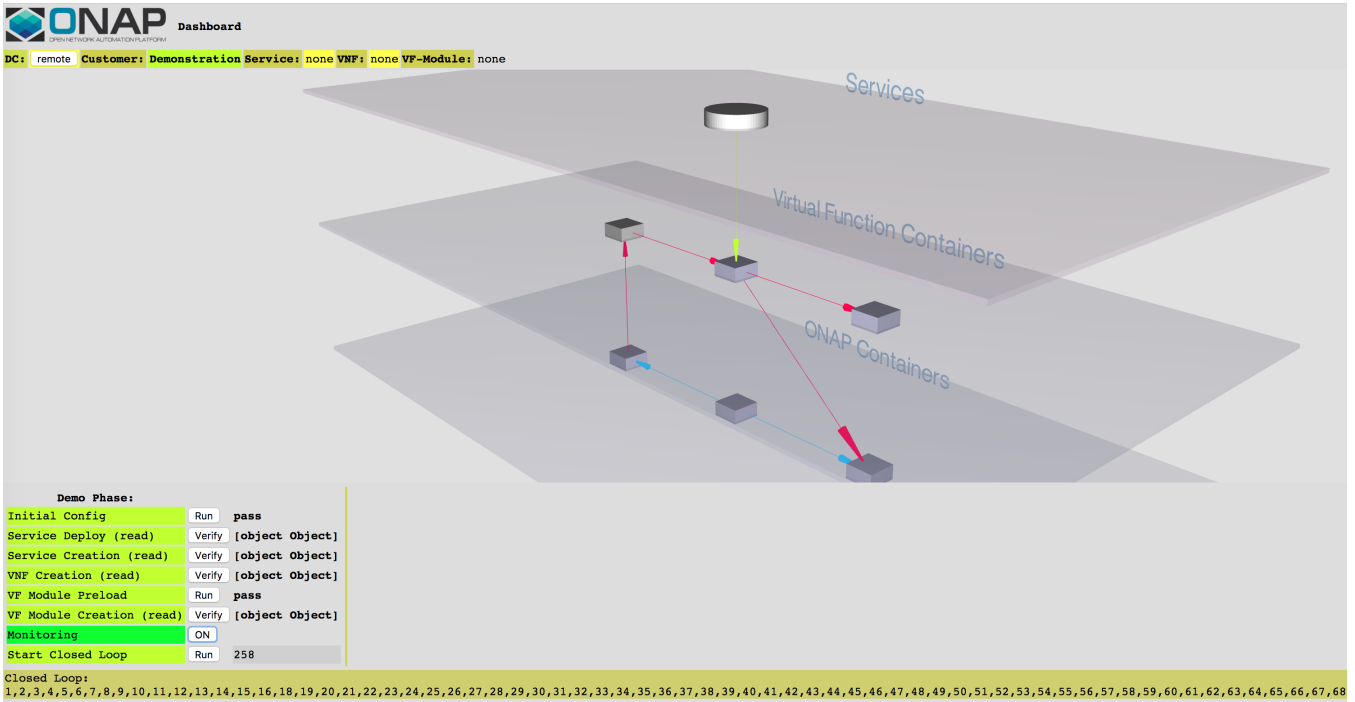
Warning: Draft Content

This wiki is under construction - this is an evolving POC

[UCA-20](#) - Getting issue details...

STATUS

Purpose: prototype a REST client on top of ONAP - specifically for VF-Module orchestration.



Analysis

vFirewall Demo Sequence of Events

All the events during a VF creation so we can understand all the orchestration steps at a granular REST level.

Passwords in /testsuite/properties/integration_robot_properties.py

Status (current/future) Flow

Before	After (REST extraction)
in demo.sh init (customer create, service model create)	2 AAI rest calls
in vid deploy service model	unchanged vid
in vid create service instance	unchanged vid
in vid add vnf to service instance	unchanged vid
in demo.sh preload	SDNC rest call
in vid vf-module creation	MSO rest call
in demo.sh appc	unchanged robot

Variables

IAD region

key	name	value
	service	c54316d8-464e-4967-bece-8c2b2f458b66
	VNF	6229cdee-10f6-4ec4-a5e0-0593154e6d83
	VfModule	5a08199b-4161-4181-9b2d-da29f6df1410

Prerequisites

Seq	Name	Source	Target	Actor	REST write	REST read
	License Model					http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-license-models FD0FBEDB1AB64D1CA1A4E266862CAEE7
	Licence Key Group					where id is from LM above http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-license-models /FD0FBEDB1AB64D1CA1A4E266862CAEE7/license-key-groups
	Entitlement Pool					where id is from LM above http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-license-models /FD0FBEDB1AB64D1CA1A4E266862CAEE7/entitlement-pools
	Feature Group					where id is from LM above http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-license-models /FD0FBEDB1AB64D1CA1A4E266862CAEE7/feature-groups
	License Agreement					where id is from LM above http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-license-models /FD0FBEDB1AB64D1CA1A4E266862CAEE7/license-agreements
	VSP					http://{sdc_ip}:8080/onboarding-api/v1.0/vendor-software-products get index by LM id "vendorId": "FD0FBEDB1AB64D1CA1A4E266862CAEE7"

Config

Seq	Name	Source	Target	Actor	robot	REST write	REST read
-----	------	--------	--------	-------	-------	------------	-----------

create customer	robot	aai	/demo.sh init	Executing robot tests at log level TRACE <pre>===== ===== ===== OpenECOMP ETE ===== ===== OpenECOMP ETE. Robot ===== ===== OpenECOMP ETE.Robot. Testsuites ===== ===== OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ... ===== ===== Initialize Customer And Models PASS ===== OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... PASS 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed ===== ===== OpenECOMP ETE.Robot. Testsuites PASS 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed ===== ===== OpenECOMP ETE. Robot PASS 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed ===== ===== OpenECOMP ETE PASS 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed ===== ===== Output: /share/logs/demo/InitDemo/output. xml Log: /share/logs/demo/InitDemo/log.html Report: /share/logs/demo/InitDemo/report. html</pre>	PUT <pre>https://{aai_ip}:8443/aai/v8/business/customers/customer/ONAP-User { "global-customer-id": "ONAP-User", "service-subscriptions": { "service-subscription": { "relationship-list": { "relationship": { "related-to": "tenant", "relationship-data": { "relationship-key": "cloud-region.cloud-owner", "relationship-value": "Rackspace" }, { "relationship-key": "cloud-region.cloud-region-id", "relationship-value": "DFW" }, { "relationship-key": "tenant.tenant-id", "relationship-value": "{{tenant_id}}" } } }, "service-type": "vFW" } }, "subscriber-name": "ONAP-User", "subscriber-type": "INFRA" }</pre> however on a clean system - where the tenant target node in the graph db does not exist yet - we fail <pre>{ "requestError": { "serviceException": { "messageId": "SVC3003", "text": "Failed to make edge to missing target node of type %3 with keys %4 performing %1 on %2 (msg=%5) (ec=%6)", "variables": ["PUT", "business/customers/customer/ONAP-User3", "tenant", "/cloud-infrastructure/cloud-regions/cloud-region/Rackspace/DFW/tenants/tenant/1035021", "Error making edge to target node:Node of type tenant. Could not find object at: /cloud-infrastructure/cloud-regions/cloud-region/Rackspace/DFW/tenants/tenant/1035021", "ERR.5.4.6129"] } } }</pre>	https://{aai_ip}:8443/aai/v8/business/customers/customer <pre>{ "global-customer-id": "Demonstration", "subscriber- name": "Demonstration",</pre>
-----------------	-------	-----	------------------	--	--	---

	create service model	robot	aai	/demo.sh init		PUT aai/v8/service-design-and-creation/services { "service-id": "\${UUID}", "service-description": "\${service_type}" }	https://[[aai_ip]]:8443/aai/v8/service-design-and-creation/models "model-name-version-id": "dcb40136-9cec-45be-b080-2a36b31c2f06", "model-type": "service", "model-name": "v1service", "model-id": "43d59b70-8829-4a50-a3f7-d261b220ceef"
--	----------------------------	-------	-----	------------------	--	---	---

Service Deployment

Seq	Name	Source	Target	Actor	REST write	REST read	REST delete
	deploy service model	vid	aai	demo user		9023f592-5c64-4a5b-9333-3287fdc9d304	

create service instance	vid	aai	demo user	Input: InstanceName=DemoInstance Subscriber=Demonstration ServiceType=vFW Output: SID=c54316d8-464e-4967-bece-8c2b2f458b66 07/27/17 12:10:39 HTTP Status: OK (200) <pre>{ "request": { "requestId": "96dcaa23-88cb-41ac-bf8c-e1e5bb423232", "startTime": "Wed, 26 Jul 2017 16:10:28 GMT", "requestScope": "service", "requestType": "createInstance", "requestDetails": { "modelInfo": { "modelCustomizationName": null, "modelInvariantId": "43d59b70-8829-4a50-a3f7-d261b220ceef", "modelType": "service", "modelNameVersionId": "dcb40136-9cec-45be-b080-2a36b31c2f06", "modelName": "vfwservice", "modelVersion": "1.0" }, "requestInfo": { "billingAccountNumber": null, "callbackUrl": null, "correlator": null, "orderNumber": null, "productFamilyId": null, "orderVersion": null, "source": "VID", "instanceName": "DemoInstance", "suppressRollback": false }, "relatedInstanceList": null, "subscriberInfo": { "globalSubscriberId": "Demonstration", "subscriberName": null }, "cloudConfiguration": null, "requestParameters": { "subscriptionServiceType": "vFW", "userParams": [] }, "instanceReferences": { "serviceInstanceId": "cd2eb659-2463-461b-9c3b-3bf03619c167", "serviceInstanceName": "DemoInstance", "vnfInstanceId": null, "vnfInstanceName": null, "vfModuleInstanceId": null, "vfModuleInstanceName": null, "volumeGroupInstanceId": null, "volumeGroupInstanceName": null, "networkInstanceId": null, "networkInstanceName": null }, "requestStatus": { "requestState": "COMPLETE", "statusMessage": "Service Instance has been created successfully.", "percentProgress": 100, "finishTime": "Wed, 26 Jul 2017 16:10:31 GMT" } } } }</pre> 07/27/17 12:10:28 HTTP Status: Accepted (202) <pre>{ "requestReferences": { "instanceId": "cd2eb659-2463-461b-9c3b-3bf03619c167", "requestId": "96dcaa23-88cb-41ac-bf8c-e1e5bb423232" } }</pre>	https://{{aai_ip}}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFW https://{{aai_ip}}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFW/service-instances <pre>{ "service-instance": { "service-instance-id": "c54316d8-464e-4967-bece-8c2b2f458b66", "service-instance-name": "DemoInstance", "persona-model-id": "2a01ef12-9bb1-4205-9941-a506cbae99c9", "requestReferences": { "instanceId": "cd2eb659-2463-461b-9c3b-3bf03619c167", "requestId": "1764e091-7d43-4eb0-92d6-b449a85926d2" } } }</pre>
-------------------------	-----	-----	-----------	---	---

VNF Creation

Seq	Name	Source	Target	Actor	REST write	REST read	REST delete
	Add VNF to Service Instance	vid	sdsc	demo user	<pre> 07/27/17 12:15:13 HTTP Status: OK (200) { "request": { "requestId": "e867be64- e725-4e29-8448-c9eceb51ce3c", "startTime": "Wed, 26 Jul 2017 16:15:02 GMT", "requestScope": "vnf", "requestType": "createInstance", "requestDetails": { "modelInfo": { "modelCustomizationName": "vsp 1", "modelInvariantId": "27036887-1053-48e8-a55e- 3d2a7264a658", "modelType": "vnf", "modelNameVersionId": "b63a0b42-a151-460a-9bb2- ba469948622b", "modelName": "vsp", "modelVersion": "1.0" }, "requestInfo": { "billingAccountNumber": null, "callbackUrl": null, "correlator": null, "orderNumber": null, "productFamilyId": "03606d31-c2f3-483e-9882- 6d050ab3610f", "orderVersion": null, "source": "VID", "instanceName": "DemoVNF", "suppressRollback": false }, "relatedInstanceList": [{ "relatedInstance": { "instanceName": null, "instanceId": "cd2eb659-2463-461b-9c3b- 3bf03619c167", "modelInfo": { "modelCustomizationName": null, "modelInvariantId": "43d59b70- 8829-4a50-a3f7-d261b220ccef", "modelType": "service", "modelNameVersionId": "dcb40136-9cec-45be-b080- 2a36b31c2f06", "modelName": "vfservice", "modelVersion": "1.0" } } }], "subscriberInfo": null, "cloudConfiguration": { "aicNodeCli": null, "tenantId": "your_tenant_id", "lcpCloudRegionId": "DFW" }, "requestParameters": { "subscriptionServiceType": null, "userParams": [] }, "instanceReferences": { "serviceInstanceId": "cd2eb659-2463-461b-9c3b- 3bf03619c167", "serviceInstanceName": null, "vnfInstanceId": "110f0e9c-52f9-4b12-8bd8- fa2980e48454", "vnfInstanceName": "DemoVNF", "vfModuleInstanceId": null, "vfModuleInstanceName": null, "volumeGroupInstanceId": null, "volumeGroupInstanceName": null, "networkInstanceId": null, "networkInstanceName": null }, "requestStatus": { "requestState": "COMPLETE", </pre>	<pre> get VSPs http://{[sdsc_ip]}:8080/onboarding-api/v1.0 /vendor-software-products "name": "vsp", "description": "vsp", "category": "resourceNewCategory. application I4+", https://{[aai_ip]}:8443/aai/v8/network /generic-vnfs/generic-vnf/6229cdee-10f6- 4ec4-a5e0-0593154e6d83 </pre>	<pre> DEL http://{[mso_ip]}:8080/ecom/mso/infra/serviceInstances/v2/cd2 eb659-2463-461b-9c3b-3bf03619c167/vnfs/42aa89af-cd4e- 43d1-8b58-73e8f6b33097 </pre>

					<pre> "statusMessage": "Vnf has been created successfully.", "percentProgress": 100, "finishTime": "Wed, 26 Jul 2017 16:15:03 GMT" } } } 07/27/17 12:15:02 HTTP Status: Accepted (202) { "requestReferences": { "instanceId": "110f0e9c- 52f9-4b12-8bd8-fa2980e48454", "requestId": "e867be64- e725-4e29-8448-c9eceb51ce3c" } } </pre>		

VfModule Preload

Seq	Name	Source	Target	Actor	robot	REST write	REST read
	preload vnf	robot		./demo.sh preload <DemoVN F> <DemoMo dule> or http://sdnc:8843/signup http://sdnc:8843/login Add VNF Profile	Starting Xvfb on display :89 with res 1280x1024x24 Executing robot tests at log level TRACE ===== ===== ===== ===== ===== OpenECOMP ETE ===== ===== ===== ===== ===== OpenECOMP ETE. Robot ===== ===== ===== ===== ===== OpenECOMP ETE.Robot. Testsuites ===== ===== ===== ===== ===== OpenECOMP ETE.Robot. Testsuites.Demo :: Executes the VNF Orchestration Test ... ===== ===== ===== ===== ===== Preload VNF PASS ----- ----- ---	<pre> http://{sdnc_ip}:8282/restconf/operations/VNF-API:preload-vnf-topology-operation where "service-type" = the root service instance id at top of vid page (do a get on - https://{aai_ip}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions /service-subscription/VFW/service-instances/ Accept:application/json Content-Type:application/json X-TransactionId:0a3f6713-ba96-4971-a6f8-c2da85a3176e X-FromAppld:API client Authorization:Basic YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VlbGNzZTJnQXc4NHZhbk0dHbUp2VXkyVQ== { "input": { "request-information": { "notification-url": "openecomp.org", "order-number": "1", "order-version": "1", "request-action": "PreloadVNFRequest", "request-id": "test" }, "sdnc-request-header": { "svc-action": "reserve", "svc-notification-url": "http://openecomp.org:8080/adapters/restVSDNCNotify", "svc-request-id": "test" }, "vnf-topology-information": { "vnf-assignments": { "availability-zones": [], "vnf-networks": [], "vnf-vms": [] }, "vnf-parameters": [{ "vnf-parameter-name": "vfw_private_ip_0", "vnf-parameter-value": "192.168.110.100" }, { "vnf-parameter-name": "vfw_private_ip_1", "vnf-parameter-value": "192.168.120.100" }, { "vnf-parameter-name": "vfw_private_ip_2", "vnf-parameter-value": "10.1.0.11" }, { </pre>	<pre> http://{sdnc_ip}: 8282/restconf/config /VNF-API:preload- vnfs { "preload-vnfs": { "vnf-preload-list": [{ "vnf-name": "DemoModule", "vnf-type": "Vsp.. base_vfw..module- 0", http://{sdnc_ip}: 8282/restconf/config /VNF-API:preload- vnfs/vnf-preload-list /DemoModule/Vsp.. base_vfw..module-0 { "vnf-preload-list": [{ "vnf-name": "DemoModule", "vnf-type": "Vsp.. base_vfw..module- 0", </pre>

OpenECOMP
ETE.Robot.
TestSuites.Demo
:: Executes the
VNF Orchestrat...
| PASS |

1 critical test, 1
passed, 0 failed

1 test total, 1
passed, 0 failed

=====
=====
=====
=====
=====

OpenECOMP
ETE.Robot.
TestSuites

| PASS |

1 critical test, 1
passed, 0 failed

1 test total, 1
passed, 0 failed

=====
=====
=====
=====
=====

OpenECOMP
ETE.
Robot

| PASS |

1 critical test, 1
passed, 0 failed

1 test total, 1
passed, 0 failed

=====
=====
=====
=====
=====

OpenECOMP
ETE

| PASS |

1 critical test, 1
passed, 0 failed

1 test total, 1
passed, 0 failed

=====
=====
=====
=====
=====

Output: /share
/logs/demo
/PreloadDemo
/output.xml

Log: /share/logs
/demo
/PreloadDemo/log.
html

Report: /share
/logs/demo
/PreloadDemo
/report.html

```
"vnf-parameter-name": "vpg_name_0",  
"vnf-parameter-value": "demofw01pgn"  
}, {  
"vnf-parameter-name": "demo_artifacts_version",  
"vnf-parameter-value": "1.0.0"  
}, {  
"vnf-parameter-name": "ecomp_private_net_cidr",  
"vnf-parameter-value": "10.0.0.0/8"  
}, {  
"vnf-parameter-name": "key_name",  
"vnf-parameter-value": "vfw_keydemo"  
}, {  
"vnf-parameter-name": "repo_url_artifacts",  
"vnf-parameter-value": "https://nexus.onap.org/content/groups/staging"  
}, {  
"vnf-parameter-name": "protected_private_net_id",  
"vnf-parameter-value": "demofw_protected"  
}, {  
"vnf-parameter-name": "vpg_private_ip_0",  
"vnf-parameter-value": "192.168.110.200"  
}, {  
"vnf-parameter-name": "vpg_private_ip_1",  
"vnf-parameter-value": "10.1.0.12"  
}, {  
"vnf-parameter-name": "unprotected_private_net_cidr",  
"vnf-parameter-value": "192.168.110.0/24"  
}, {  
"vnf-parameter-name": "dcae_collector_ip",  
"vnf-parameter-value": "10.0.4.102"  
}, {  
"vnf-parameter-name": "repo_url_blob",  
"vnf-parameter-value": "https://nexus.onap.org/content/repositories/raw"  
}, {  
"vnf-parameter-name": "vsn_private_ip_0",  
"vnf-parameter-value": "192.168.120.250"  
}, {  
"vnf-parameter-name": "vsn_private_ip_1",  
"vnf-parameter-value": "10.1.0.13"  
}, {  
"vnf-parameter-name": "vsn_name_0",  
"vnf-parameter-value": "demofw01snk"  
}, {  
"vnf-parameter-name": "vfw_name_0",  
"vnf-parameter-value": "demofw01fwl"  
}, {  
"vnf-parameter-name": "pub_key",  
"vnf-parameter-value": "ssh-rsa  
AAAAB3NzaC1yc2EAAAABJQAAAAQEAQgnA9BAiMLtjOPSYBfhzLu4CiBoIWoskDg4KVvhTJVTTeB6CqrQNC  
adlGXxOHhCYuNCKkUmlVF4WTOisVOJ75Z1c4OMoZLL85xVPKSlEh63kgVugwgPYQu55NbbWX+rsbUha3  
LnEIDhNviMM3iTPbD5nnhKixNERAJMTLKXvZZZGqxW94bREknYPQTT2qrk3YRqWldncopp6Nkgv3AnSJz2g  
c9tjxnWF0poTQnQm/3D6hiJlCrzKfAV0EaPN0KdtYjPhKrYoy6Qb/tKOVaaqsvwfKBjGrT9LfcA7D7M  
/yj292RT1XN63hi84WC383LsaPJ6eWdDTE6zUP1eGTWCOW== rsa-key-20161026"  
}, {  
"vnf-parameter-name": "ecomp_private_net_id",
```

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VfModule Creation

Seq	Name	Source	Target	Actor	REST write	REST read	REST delete
	create Vf module <DemoModule>	vid		demo user	Success without running VID! - workaround for VID-19 - Getting issue details... STATUS Output: VfModuleID=5a08199b-4161-4181-9b2d-da29f6df1410 HeatStackID=DemoModule/55bd16d6-6bac-4638-9e87-9063df6d03c3 via OpenStack openstack server list 4be336b3-a31c-4be0-9261-0cb1cb51227d demofwl01fwl ACTIVE oam_ecomp_LbMF=10.1.0.11; demofwl_unprotected=192.168.110.100; public=104.239.230.93, 2001:4802:7803:104:be76:4eff:fe20:377b; demofwl_protected=192.168.120.100 post call http://{{mso_ip}}:8080/ecomp/mso/infra/serviceInstances/v2/<id>/vnfs/<id>/vfModules inputs: serviceInstance id = c543... from relatedInstance:type=service vnf id = 6229.. from relatedInstance:type=vnf using https://{{aai_ip}}:8443/aai/v8/network/generic-vnfs get: {"relatedInstance": { "instanceId": "6229cdee-10f6-4ec4-a5e0-0593154e6d83", from: {"generic-vnf": [{"vnf-id": "6229cdee-10f6-4ec4-a5e0-0593154e6d83"	http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfModuleInstanceId%3AEQUALS%3A5a08199b-4161-4181-9b2d-da29f6df1410 http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfInstanceId%3AEQUALS%3A110f0e9c-52f9-4b12-8bd8-fa2980e48454 http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfModuleInstanceId%3AEQUALS%3Ad5508965-35c6-4720-a7f0-4b0e17f25c9d { "requestList": [{ "request": { "requestId": "9a3510d2-9aa7-486a-8b39-fab543c718ec", "startTime": "Wed, 26 Jul 2017 21:10:37 GMT", "requestScope": "VfModule", "requestType": "createInstance",	get ID via http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfModuleInstanceId%3AEQUALS%3Ad5508965-35c6-4720-a7f0-4b0e17f25c9d { ... }, "instanceReferences": { "serviceInstance": { "cd2eb659-2463-461b-9c3b-3bf03619c167", "serviceInstanceName": null, "vnfInstance": { "110f0e9c-52f9-4b12-8bd8-fa2980e48454", "vnfInstanceName": null, "vnfModuleInstance": { "d5508965-35c6-4720-a7f0-4b0e17f25c9d", "vnfModuleInstanceName": "DemoModule", "volumeGroupInstance": null, "volumeGroupInstanceName": null, "networkInstance": null, "networkInstanceName": null }, "requestStatus": { "requestState": "COMPLETE", "statusMessage": "Vf Module has been created successfully.", "percentProgress": 100, "finishTime": "Wed, 26 Jul 2017 21:14:26 GMT" }}] } DEL to

				<pre> get: modelInvariantId": "7425ba97-e159-4c0f-87dd-3af3b439148b" from: "persona-model-id": "7425ba97-e159-4c0f-87dd-3af3b439148b" using https://{[aai_ip]}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFW/service-instances/ get: "instanceReferences": { "serviceInstanceid": "c54316d8-464e-4967-bece-8c2b2f458b66" from: { "service-instance": [{"service-instance-id": "c54316d8-464e-4967-bece-8c2b2f458b66", body : { "requestDetails": { "requestInfo": { "instanceName": "DemoModule", "source": "VID", "suppressRollback": false }, "modelInfo": { "modelType": "vfModule", "modelInvariantId": "1051e887-365f-4e90-b4cc-3f5c6f588188", "modelNameVersionId": "02b1cf61-2dc9-4be9-869c-354395670faf", "modelName": "Vsp..base_vfw..module-0", "modelVersion": "1.0" }, "requestParameters": { "userParams": [] }, "relatedInstanceList": [{"relatedInstance": { "instanceid": "cd2eb659-2463-461b-9c3b-3bf03619c167", // from "service-instance-id" of https://{[aai_ip]}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFW/service-instances/ "modelInfo": { "modelType": "service", "modelName": "vfservice", // from vnf-type of https://{[aai_ip]}:8443/aai/v8/network/generic-vnfs "modelInvariantId": "43d59b70-8829-4a50-a3f7-d261b220ceef", // from "persona-model-id" of https://{[aai_ip]}:8443/aai/v8/business/customers/customer/Demonstration/service-subscriptions/service-subscription/vFW/service-instances/ "modelVersion": "1.0", "modelNameVersionId": "831ed809-345e-4ed0-b216-6308c0f0d8e4" } }], "relatedInstance": { "instanceid": "110f0e9c-52f9-4b12-8bd8-fa2980e48454", // from "vnf-id" of https://{[aai_ip]}:8443/aai/v8/network/generic-vnfs "modelInfo": { "modelType": "vnf", "modelName": "vsp", "modelInvariantId": "27036887-1053-48e8-a55e-3d2a7264a658", // from "persona-model-id" of https://{[aai_ip]}:8443/aai/v8/network/generic-vnfs "modelVersion": "1.0", "modelNameVersionId": "9c265d0b-e606-4242-9ace-98962a1b93e7", "modelCustomizationName": "vsp 1" } } }, "cloudConfiguration": { "lcpCloudRegionId": "DFW", "tenantId": "{tenant_id}" } } however there is more to this than just a rest call - no heat stack means no vf instances yet https://{[mso_ip]}:8080/ecomp/mso/infra/serviceInstances/v2/cd2eb659-2463-461b-9c3b-3bf03619c167/vnfs/110f0e9c-52f9-4b12-8bd8-fa2980e48454/vfModules { "requestReferences": { "instanceid": "52d939c6-e603-4782-92e8-962387ae5de8", "requestId": "8869bc00-f07e-48d2-a55b-c725cfca44f4" } } however a call to the following - shows we need to get the right network name in the preload https://{[mso_ip]}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfInstanceid%3AEQUALS%3A110f0e9c-52f9-4b12-8bd8-fa2980e48454 "requestStatus": { "requestState": "FAILED", "statusMessage": "Received vfModuleException from VnfAdapter: category=INTERNAL: message=Exception during create VF 400 Bad Request: The server could not comply with the request since it is either malformed or otherwise incorrect., error.type=StackValidationFailed, error.message=Property error: : resources.vfw_private_2_port.properties.fixed_ips[0].subnet: : Error validating value 'oam_ecomp_LbMf': Unable to find subnet with name or id 'oam_ecomp_LbMf' rolledBack=true", retry after redoing preload with proper subnet name oam_ecomp_QbZ1 200 OK { "requestReferences": { "instanceid": "d5508965-35c6-4720-a7f0-4b0e17f25c9d", "requestId": "9a3510d2-9aa7-486a-8b39-fab543c718ec" } } </pre>	<pre> "requestDetails": { "modelInfo": { "modelCustomizationName": null, "modelInvariantId": "1051e887-365f-4e90-b4cc-3f5c6f588188", "modelType": "vfModule", "modelNameVersionId": "02b1cf61-2dc9-4be9-869c-354395670faf", "modelName": "Vsp..base_vfw..module-0", "modelVersion": "1.0" }, "requestInfo": { "billingAccountNumber": null, "callbackUrl": null, "correlator": null, "orderNumber": null, "productFamilyId": null, "orderVersion": null, "source": "VID", "instanceName": "DemoModule", "suppressRollback": false }, "relatedInstanceList": [{ "relatedInstance": { "instanceName": null, "instanceid": "cd2eb659-2463-461b-9c3b-3bf03619c167", "modelInfo": { "modelCustomizationName": null, "modelInvariantId": "43d59b70-8829-4a50-a3f7-d261b220ceef", "modelType": "service", "modelNameVersionId": "15eb2af8-766c-4e53-9038-a4a95e79f105", "modelName": "vfservice", "modelVersion": "1.0" } } }, "relatedInstance": { "instanceName": null, "instanceid": "110f0e9c-52f9-4b12-8bd8-fa2980e48454", "modelInfo": { "modelCustomizationName": "vsp 1", "modelInvariantId": "27036887-1053-48e8-a55e-3d2a7264a658", "modelType": "vnf", "modelNameVersionId": "2e5273f3-a6a4-4c50-9b0a-bc9b25749ca3", "modelName": "vsp", "modelVersion": "1.0" } } }, "subscriberInfo": null, "cloudConfiguration": { "aicNodeCli": null, "tenantId": "your_tenant_id", "lcpCloudRegionId": "DFW" }, "requestParameters": { "subscriptionServiceType": null, "userParams": [] }, "instanceReferences": { "serviceInstanceid": "cd2eb659-2463-461b-9c3b-3bf03619c167", "serviceInstanceName": null, "vnfInstanceid": "110f0e9c-52f9-4b12-8bd8-fa2980e48454", "vnfInstanceName": null, "vfModuleInstanceid": "d5508965-35c6-4720-a7f0-4b0e17f25c9d", "vfModuleInstanceName": "DemoModule", "volumeGroupInstanceid": null, "volumeGroupInstanceName": null, "networkInstanceid": null, "networkInstanceName": null }, "requestStatus": { "requestState": "COMPLETE", "statusMessage": "Vf Module has been created successfully.", "percentProgress": 100, "finishTime": "Wed, 26 Jul 2017 21:14:26 GMT" } } } </pre> http://{[mso_ip]}:8080/ecomp/mso/infra/serviceInstances/v2/cd2eb659-2463-461b-9c3b-3bf03619c167/vnfs/110f0e9c-52f9-4b12-8bd8-fa2980e48454/vfModules/d5508965-35c6-4720-a7f0-4b0e17f25c9d fails - referencing an older preload <pre> { "serviceException": { "messageId": "SVC0002", "text": "Error parsing request. No valid serviceInstanceid matching the serviceInstanceid in request URI is specified" } } </pre>

Closed Loop

Seq	Name	Source	Target	Actor	Robot	REST write	REST read
	appc	robot		.demo.sh appc. <DemoModule>	Starting Xvfb on display :89 with res 1280x1024x24 Executing robot tests at log level TRACE ===== OpenECOMP ETE ===== OpenECOMP ETE. Robot ===== OpenECOMP ETE.Robot. Testsuites ===== OpenECOMP ETE.Robot. Testsuites. Demo :: Executes the VNF Orchestration Test ... ===== Create APPC Mount Point PASS ----- -- OpenECOMP ETE.Robot. Testsuites. Demo :: Executes the VNF Orchestrat... PASS 1 critical test, 1 passed, 0 failed		

1 test total, 1
passed, 0
failed

=====
=====
=====
=====
=====
=====

OpenECOMP
ETE.Robot.
Testsuites

| PASS |

1 critical test,
1 passed, 0
failed

1 test total, 1
passed, 0
failed

=====
=====
=====
=====
=====
=====

OpenECOMP
ETE.
Robot

|
PASS |

1 critical test,
1 passed, 0
failed

1 test total, 1
passed, 0
failed

=====
=====
=====
=====
=====
=====

OpenECOMP
ETE

|
PASS |

1 critical test,
1 passed, 0
failed

1 test total, 1
passed, 0
failed

=====
=====
=====
=====
=====
=====

Output:
/share/logs
/demo
/APPCMountP
ointDemo
/output.xml

Log: /share
/logs/demo
/APPCMountP
ointDemo/log.
html

Report:
/share/logs
/demo
/APPCMountP
ointDemo
/report.html

			DMaaP			http://{{collector_ip}}:3904/events/unauthenticated.SEC_MEASUREMENT_OUTPUT/group3/sub1?timeout=9000 see JSON.parse(returnedData[i].event.commonEventHeader.eventId <pre>[{"event":{"measurementsForVfScalingFields":{"measurementInterval":"10","measurementsForVfScalingVersion":"1.1","vNicUsageArray":{"multicastPacketsIn":"0","bytesIn":"14500","unicastPacketsIn":"0","multicastPacketsOut":"0","broadcastPacketsOut":"0","packetsOut":"0","bytesOut":"0","broadcastPacketsIn":"0","packetsIn":"500","unicastPacketsOut":"0","vNicIdentifier":{"eth1"}}},"commonEventHeader":{"reportingEntityName":{"demofw01fwl"},"startEpochMicrosec":"1500849487268085","lastEpochMicrosec":"1500849497268085","eventId":{"43434"},"sourceName":{"Dummy VM name - No Metadata available"},"sequence":{"43434"},"priority":{"Normal"},"functionalRole":{"vFirewall"},"domain":{"measurementsForVfScaling"},"reportingEntityId":{"No UUID available"},"sourceId":{"Dummy VM UUID - No Metadata available"},"version":{"1.2}}},</pre>
			DMaaP			http://{{collector_ip}}:3904/events/unauthenticated.TCA_EVENT_OUTPUT/group3/sub1?timeout=30000 <pre>[{"closedLoopEventClient":{"DCAE_INSTANCE_ID.dcae-tcal"},"policyVersion":{"1.0.0.5"},"policyName":{"vFirewall"},"policyScope":{"resource=SampleResource,service=SampleService,type=SampleType,closedLoopControlName=SampleClosedLoop"},"target_type":{"VNF"},"AAI":{"generic-vnf.vnf-id":{"demofw01fwl"},"closedLoopAlarmStart":{"1501103790415551"},"closedLoopEventStatus":{"ONSET"},"closedLoopControlName":{"CL-FRWL-LOW-TRAFFIC-SIG-d925ed73-8231-4d02-9545-db4e101f88f8"},"version":{"1.0.2"},"target":{"generic-vnf.vnf-id"},"requestID":{"05ac4b8f-b844-439f-a6d1-c2713356e102"},"from":{"DCAE"}},</pre> <pre>"closedLoopEventClient":{"DCAE_INSTANCE_ID.dcae-tcal"},"policyVersion":{"1.0.0.5"},"policyName":{"vFirewall"},"policyScope":{"resource=SampleResource,service=SampleService,type=SampleType,closedLoopControlName=SampleClosedLoop"},"target_type":{"VNF"},"AAI":{"generic-vnf.vnf-id":{"demofw01fwl"},"closedLoopAlarmStart":{"1501103870657180"},"closedLoopEventStatus":{"ONSET"},"closedLoopControlName":{"CL-FRWL-LOW-TRAFFIC-SIG-d925ed73-8231-4d02-9545-db4e101f88f8"},"version":{"1.0.2"},"target":{"generic-vnf.vnf-id"},"requestID":{"cb65cec6-4058-42a2-9e19-8ef61fea1fbf"},"from":{"DCAE"}},</pre>

API History

http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=serviceInstanceId%3AEQUALS%3Ac54316d8-464e-4967-bece-8c2b2f458b66

http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vnfInstanceId%3AEQUALS%3A6229cdee-10f6-4ec4-a5e0-0593154e6d83

http://{{mso_ip}}:8080/ecomp/mso/infra/orchestrationRequests/v2/?filter=vfModuleInstanceId%3AEQUALS%3A5a08199b-4161-4181-9b2d-da29f6df1410

see also

http://{{sdc_ip}}:8080/sdc2/rest/v1/catalog/services/latestversion/notabstract/uidonly?internalComponentType=SERVICE

feeds

http://{{sdc_ip}}:8080/sdc2/rest/v1/catalog/services/378bc9b7-2b7d-4a4f-96d9-1c654c2242fa

Design Issues

DI 1: 20170712: AAI Cert required for HTTPS REST calls

Use [postman for adhoc rest calls](#) - but if you want to code up call chains or hammer an endpoint use Spring RestController or the Rest client in JAX-RS 2.0

Calls to AAI such as the following require both the authentication header and an imported certificate. When running Postman - because it is a Chrome app - this is a simple case of loading a REST url in the browser and importing the certificate after an authentication challenge (AAI:AAI). However for a java client like a JAX-RS 2.0 client we need the certificate in a keystore (the default or a specially defined one).

Below we import the cert into the default keystore. Where did I get the cert? by extracting it from Firefox - however it is in the code base - looking it up

CURL commands need the certificate - see [Verifying your ONAP Deployment#Postman/CurlRESTcalls](#)

Note: we will need to replace the AAI Certificate before 1 Dec 2017 -

[AAI-94](#) - Getting issue details...

STATUS

```
$ ls $JAVA_HOME/jre/lib/security/cacerts

/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/jre/lib/security/cacerts

sudo keytool -import -trustcacerts -alias aai -file /config/certs/aai/aaiapisimpledemoopenecomporg.cer -keystore $JAVA_HOME/jre/lib/security/cacerts

on robot

root@vm1-robot:~# keytool -list -keystore /usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/cacerts -v

sudo scp aaiapisimpledemoopenecomporg.cer root@robot:/opt

keytool -import -trustcacerts -alias aai -file aaiapisimpledemoopenecomporg.cer -keystore /usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/cacerts

the aai keystore password does not work - I will use my own truststore

"AAI_KEystore_PASSWD_X": "0BF:1vn21ugu1saj1v9i1v941sar1ugw1vo0"
```

Installing the AAI certificate in your own keystore on the Robot VM

```
root@vm1-robot:/opt# keytool -import -trustcacerts -alias aai -file aaiapisimpledemoopenecomporg.cer -keystore /usr/lib/jvm/java-1.8.0-openjdk-amd64/jre/lib/security/cacerts

Enter keystore password:

// use our own

keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect

root@vm1-robot:/opt# keytool -genkey -alias onap -keyalg RSA -keystore KeyStore.jks -keysize 2048

Enter keystore password:

Is CN=Michael OBrien, OU=onap, O=onap, L=Unknown, ST=Unknown, C=Unknown correct?

Enter key password for <onap>

(RETURN if same as keystore password):

Re-enter new password:


// import cer

root@vm1-robot:/opt# keytool -import -trustcacerts -alias aai -file aaiapisimpledemoopenecomporg.cer -keystore KeyStore.jks

Enter keystore password:

Owner: EMAILADDRESS=aai-host@api.simpLEDemo.openecomp.org, CN=aai.api.simpLEDemo.openecomp.org, OU=SimpleDemo, O=OpenECOMP, L=Bedminster, ST=NJ, C=US

Issuer: EMAILADDRESS=simpLEDemo@openecomp.org, CN=OpenECOMP simpLEDemo Server CA X1, OU=simpLEDemo, O=OpenECOMP, L=Bedminster, ST=NJ, C=US

Serial number: 83ca7c32dc9f7329

Valid from: Wed Nov 30 15:38:39 UTC 2016 until: Thu Nov 30 15:38:39 UTC 2017

Trust this certificate? [no]: yes

Certificate was added to keystore
```

Running an https rest target using this certificate - if the cert is in a default keystore - you dont need to define it - otherwise set your own trust/keystore as below.

```
// require: cert, username, password, headers(X-FromAppId,Accept), Authenticator
public String run(boolean isSSL, String url, String port, String path) {

    String record = null;
    Client client = null;
    WebTarget latestTarget = null;
    WebTarget rootTarget = null;
    if(isSSL) {
        // Use only for default JDK keystore

        // SslConfigurator sslConfig = SslConfigurator.newInstance();

        SslConfigurator sslConfig = SslConfigurator.newInstance()

        .trustStoreFile("/opt/KeyStore.jks")

        .trustStorePassword("changeme")/"changeit")

        .keyStoreFile("/opt/KeyStore.jks")

        .keyPassword("changeme")/"changeit");

        SSLContext sslContext = sslConfig.createSSLContext();
        HostnameVerifier verifier = new HostnameVerifier() {
            public boolean verify(String hostname, SSLSession sslSession) {
                return true; // TODO: security breach
            }
        };
        client = ClientBuilder.newBuilder().sslContext(sslContext).hostnameVerifier(verifier).build();
        client.register(new Authenticator("AAI", "AAI"));
    } else {

        client = ClientBuilder.newClient();
    }
    rootTarget = client.target(url);
    latestTarget = rootTarget.path(path);
    try {
        try { Thread.sleep(1); } catch (InterruptedException ie) { Thread.currentThread().interrupt(); }
        record = finalTarget.request()
            .header("X-FromAppId", "AAI").header("Accept", "application/json")
            .get(String.class);
    }
}
```

```
public class Authenticator implements
ClientRequestFilter {
    private final String user;
    private final String pass;

    public Authenticator(String user,
String password) {
        this.user = user; this.pass =
password;
    }

    public void filter
(ClientRequestContext requestContext)
throws IOException {
        MultivaluedMap<String, Object>
headMap = requestContext.getHeaders();
        String basicAuth = null;
        try {
            String aToken = user + ":" +
pass;
            basicAuth = "BASIC " +
DatatypeConverter.printBase64Binary
(aToken.getBytes("UTF-8"));
        } catch (UnsupportedEncodingException
uee) { throw new IllegalStateException(
"Encoding with UTF-8 failed", uee);}
        headMap.add("Authorization",
basicAuth);
    }
}
```

Creating a custom keystore on the robot VM

DI 2: 20170712: Spring Boot Backend Framework

Instead of using a full Tomcat deployment server - we will use an embedded Jetty container.

pom.xml

```
<parent><groupId>org.springframework.boot</groupId><artifactId>spring-boot-starter-parent</artifactId><version>1.5.3.RELEASE</version></parent>

<dependencies>
<dependency><groupId>org.springframework.boot</groupId><artifactId>spring-boot-starter-web</artifactId></dependency>

    <!-- avoid restarts --><dependency><groupId>org.springframework.boot</groupId><artifactId>spring-boot-devtools</artifactId><optional>true</optional></dependency>

    <dependency><groupId>org.springframework.boot</groupId><artifactId>spring-boot-starter-actuator</artifactId></dependency>
<dependency><groupId>org.springframework.boot</groupId><artifactId>spring-boot-starter-test</artifactId><scope>test</scope></dependency>
</dependencies>

<build><plugins><plugin><groupId>org.springframework.boot</groupId><artifactId>spring-boot-maven-plugin</artifactId>
    <!-- may not be required --><configuration><addResources>true</addResources> </configuration></plugin></plugins></build>
```

DI 3: 20170712: JAX-RS 2.0 API Framework

DI 4: 20170712: 3D Framework: Three js OpenGL 3D wrapper on Canvas

changing state

```
scene.getObjectByName(arrowToFlash).line.material.color.set(flashOnColor);
scene.getObjectByName(arrowToFlash).cone.material.color.set(flashOnColor);
aDmaap.material.color.set(flashOnColor);
aFirewall.material.color.set(flashOnColor);
```

DI 5: 20170712: Javascript Timer Framework

DI 6: 20170712: Three js Listener

DI 7: 20170712: Three js Edges between Nodes

```

var vmFromNames = ["generator", "firewall", "service-firewall", "firewall", "appc", "policy", "policy"];
var vmToNames = ["firewall", "sink", "firewall", "dmaap", "generator", "dmaap", "appc"];
var edgeColors = [ colorRed, colorRed, colorGreen, colorRed, colorRed, colorBlue, colorBlue];
for(var vm=0; vm<vmFromNames.length; vm++) {
threejsFlashStates[vmNames[vm]] = 0;

var startObject = scene.getObjectByName(vmFromNames[vm]);
var endObject = scene.getObjectByName(vmToNames[vm]);

var startPoint = new THREE.Vector3(startObject.position.x, startObject.position.y, startObject.position.z);
var endPoint = new THREE.Vector3(endObject.position.x, endObject.position.y, endObject.position.z);
var direction = new THREE.Vector3().subVectors(endPoint, startPoint).normalize();

var arrow = new THREE.ArrowHelper(direction, startPoint, startPoint.distanceTo(endPoint), edgeColors[vm] );
arrow.line.material.linewidth = 4; // no effect on windows

var vName = "edge-" + vmFromNames[vm] + "-" + vmToNames[vm];

console.log(vName);

arrow.name = vName;

group.add(arrow);
}

```

DI 8: 20170712: Three js Click Context

DI 9: 20170712: AJAX JAX-RS Channel

Example: getCustomer() from AAI

AJAX client

```

function getCustomer() {
$.ajax({url: "/api",
data: { action: 'customer-read' },
success: function( data ) {
var returnedData = JSON.parse(data.content);
console.log(returnedData);
document.getElementById('config-customer-value').bgColor='#c0ff30'
$("#config-customer-value" ).html( "<strong>" + returnedData['global-customer-id'] + "</strong>" );
}}); }

```

JAX-RS NBI

```

@Controller @RequestMapping("/api")
public class ApiController {

    @RequestMapping(method=RequestMethod.GET)

    public @ResponseBody Api process(@RequestParam(value="action", required=true, defaultValue="undefined") String action) {
content = client.run(true, Configuration.get(Configuration.DC, "aai-ip"), "8443", "aai/v8/business/customers/customer", "AAI", "AAI", "AAI");
    }
}

```

JAX-RS SBI

```
record = rootTarget.request()
.header("X-FromAppId", appId).header("Accept", "application/json")
.get(String.class);
```

DI 10: 20170712: WebSocket Channel

DI 11: 20170712: React.js Framework

DI 12: 20170712: Three js Text

DI 13: 20170721: Demo Robot REST extraction

Reverse engineer the orchestration rest calls coming out of demo.sh

```

/testsuite/properties/demo.sh not /testsuite/docker/demo.sh

/testsuite/runtags.sh

/testsuite/resources/demo_preload.robot "Load Customer And Models"

"Create Customer For VNF Demo"

create_customer.robot (aai put /aai/v8//business/customers/customer/)

create_service.robot (aai put /aai/v8/service-design-and-creation/services

"Load Models" | Distribute Model | Model Distribution For Directory in model_test_template.robot | Distribute Model From ASDC in
addc_interface.robot

(Add ASDC Catalog Service, Setup ASDC Catalog Resource, Get ASDC Catalog Resource,
Add ASDC Resource Instance, Get ASDC Catalog Service, Checkin ASDC Catalog Service,
Request Certify ASDC Catalog Service, Start Certify ASDC Catalog Service, Certify ASDC Catalog Service,
Approve ASDC Catalog Service,
Distribute ASDC Catalog Service, Get ASDC Catalog Service, Check Catalog Service Distributed)

Preload VNF:demo.robot (
  Preload User Model:demo_preload.robot (
    Get Service Instance
    Get Relationship Data
    Get Persona Model Id
    Login To VID GUI
    Get Module Names from VID
    Preload Vnf:sdngc_interface.robot(
      Update Module Name
      Preload Vnf Profile(
        Login To SDNGC Admin GUI
goto sdnc:8843/mobility/getVnfProfile
      click add_vnf_profile

inserts directly to the DB from the client (for now)

/sdnc-oam/admportal/mobility.js

router.post('/addVnfNetwork', csp.checkAuth, function(req,res){
  var sql = "INSERT INTO VNF_NETWORKS (vnf_type,network_role) VALUES ("
  + "'" + req.body.nf_vnf_type + "'," + "'" + req.body.nf_network_role + "'" );
  tasks.push( function(callback) { dbRoutes.executeSQL(sql,req,res,callback); } );
}

  Preload One Vnf Topology
)))

```

DI 20: 20170721: Rest API Chaining

DI 21: 20170727: Parse VF-Module IPs from endpoint after VF-Module creation

DI 22: 20170728: Determine Healthcheck for vFW VMs are ready for appc Closed-Loop

The box should be ready for DMaaP TCA events - not just up with a 200.

Check port 667 on the sink vm.

DI 23: 20170728: Add granular Policy healthcheck for PDBD/PAP/PDP

```
http://{{policy_ip}}:6969/healthcheck
```

```
Content-Type:application/json
//ClientAuth:cHl0aG9uOnRlc3Q=
///User-Agent:libcurl-agent/1.0
Authorization:Basic aGVhbHRoY2h1Y2s6emlhWHp0RzM0
```

DI 24: 20170728: Add TCA topic event capture between Policy and APPC

DI 25: 20170806: Add jcloud API to query VMs via Nova

<http://jclouds.apache.org/guides/openstack/>

DI 26: 20170806: Add docker ps parse calls to get docker health

```
ssh root@aai1 -o StrictHostKeyChecking=no 'docker ps'
```

will return results to calling server.

Notes

The MSO Rest post workaround to avoid VID for VF-Module creation fixes

[VID-19 - Getting issue details...](#)

STATUS