KNF Orchestration: Kubernetes Functions, Helm and Juju Bundles

Gülsüm Atıcı (Canonical, OSM RO MDL)
October 19, 2022
Welcome to the Hackfest
Agenda

- Hackfest Environment
- OSM Overview
- Use Case: Magma Orchestration
- Hands On Session: Deployment of Network Services
- Kubernetes Functions
- Hands On Session: Running Day 2 Actions
Hackfest Environment

SSID: OSM_Hackfest
Password: WIFI4hackfest!

Remote Desktop

You

OpenStack

Virtual Desktop

OSM

Kubernetes

Hub for Interoperability and Validation at ETSI (HIVE)
Logging Into OSM

Remote Desktop

firefox https://ui.osm

Username: admin
Password: hackfest
Your Openstack Tenant
OSM Overview
OSM Concepts

- VNF/PNF/KNF
- VNFD, NSD
- Network Service
OSM Concepts: Information Model & Packages

OSM IM reference link: https://osm.etsi.org/docs/user-guide/latest/11-osm-im.html
OSM Concepts: VNF

**Virtual Network Function**
- One or more Deployment Units
- Internal networks
- Internal connection points (interfaces)
- Mapping VDU connections to the networks
- External connection points

**VDU: Virtual Deployment Unit**
- Virtual Machines
- OSM models vCPUs, RAM, Storage, Interfaces, Performance Capabilities (SR-IOV, EPA)
OSM Concepts: PNF

Physical Network Function

- Models an already existing physical application.
- It uses the same concepts as the VNF

PDU: Physical Deployment Unit

- Already existing application
- No control over the Lifecycle
- Perform operations on it
**OSM Concepts: KNF**

**Kubernetes Network Function**

Composed of one or more KDUs and the connection points to communicate with other KNFs/VNFs/PNFs.

Specify the networks that need to be already present in the K8s cluster.

**KDU: Kubernetes Deployment Unit**

- Kubernetes applications
- A KDU represents a Helm Chart or a Juju Bundle
OSM Concepts: VNFd

- Metadata

  Operations Package - “Charm”
  - Lifecycle
  - Configuration
  - Operation
  - Integration

 DECLARATIVE

 CODE
OSM Concepts: Network Service

- One or more xNFs
- Networks
- Mapping xNF connections to the networks
- Network Service level connection points
OSM Primitives
OSM Primitives are actions exposed by the Charm

- Backup
- Monitor
- Debug
- Add users, policies, rules, etc.
- Manage certificates, keys, etc.
- Rotate logs

Each ‘ primitive’ is a charm action that takes parameters and produces output.
VCA coordinates all OSM Primitives
VNFds map OSM Primitives to Charm Actions

VNFd

primitives:
- restore:
  mariadb:restore
- copy:
  mediawiki:zzz

“Mariadb”
actions:
- restore
- backup

“Mediawiki”
actions:
- yyy
- copy
VCA uses multiple models for Scenario

Juju client

Juju Controller

LCM

VCA

Model K8s cloud

Model VIM cloud

Kubernetes

VIM

KNF

KNF

VNF

VNF

VNF
Performs the operations on VNF/PNF/CNF instances by using actions:
Charms
What is a charm?

A charm is a set of scripts for deploying and operating application

- Built-in event handling
- Organized by layers → Helps reusing code
- Provide/require interfaces to exchange data with other charms
- Utilizes Juju to deploy across multiple substrates

wordpress

mysql

mysql-interface

Provides
mysql-interface

Requires
mysql-interface
Charms are packages of scripts to drive apps

Lifecycle Scripts
- install
- config
- update
- remove
- scale

“Action” Scripts are OSM Primitives
- “action: backup”
- “action: restore”
- “action: scan-viruses”
- “action: health-check”
- “action: add-repo”
- “action: ...”
- “action: ...”
- “action: ...”

Integration Scripts
- relate-mysql
- relate-ldap
- relate-proxy
- relate-...

These are your operation primitives.
Native Charm Approach

Workloads optimised for OSM

Workloads have the charm that drives itself directly
Proxy Charm Approach

- LXD Operators
- K8s Operators

‘Operator’ instance

Proxy Charm

OSM

Operations code run outside of the workload

Operators run the code (K8s, LXD)
What can we do by using Charms?
Configure

https_proxy: xxx
ca_cert: yyyyy
...

LCM → VCA

Charm

Workload
PNF/VNF/KNF
Actions

Add user!
Do backup!

LCM → VCA → Charm → Workload (PNF/VNF/KNF)
Integrate!
VNFds can describe Complex Integrations

Lifecycle scripts
Config scripts
Integration scripts
Action scripts
Relations

Matching integration points can be related

Lines of integration between matching integration points on different charms
Magma Orchestration Use Case
Magma: A Free Wireless Mobile Core Network

Learn more on www.magmacore.org
Magma Components

Access Gateway
- Contains core network elements
- In an LTE network:
  - Evolved Packet Core (EPC)
  - Connectivity with EnodeB’s

Orchestrator
- Cloud service
- Allows user to configure and monitor the wireless network
- Allows management of multiple Access Gateways
- Web UI

Federation Gateway
- Cloud service
- Integration with Mobile Network Operator
Magma is an Evolved Packet Core that runs on K8s
Onboarding Scenario

Access Orch8r GUI

Orchestrator

Name Resolution

DNS Server

K8s Cluster
Onboarding Scenario

OSM

Proxy Charm
PowerDNS

Native Charm
Orchestrator

Kubernetes
Magma Orchestrator as a Juju bundle

Model

Data platform

postgresql

accessd configurator directoryd ctraced device
nms-magmalte orchestrator bootstrapper nginx

dispatcher streamer ha obsidian

eventd analytics service-registry metricsd

nms-nginx-proxy orchestrator bootstrapper nginx

prometheus

alertmanager grafana

prometheus-cache

Observability

postgres
NS Diagram

NS: magma_orc8r_ns

VL: mgmtnet

CP: mgmtnet-ext

NF: magma_orc8r_cnf
KNF Diagram

External CP: mgmt-ext

NF: magma_orc8r_cnf

KDU: orc8r
- juju-bundle: bundle.yaml
KNF Diagram: K8s Cluster Requirements

K8s Cluster is already deployed and ready to use.

- Net 1: mgmtnet
- External CP: mgmtnet-ext
- NF: magma_orc8r_cnf
- K8s cluster
  - Version: 1.23
Referencing the Juju-Bundle (VNFD)

```yaml
vnfd:
  [...]
  kdu:
    - name: magma-orc-kdu
  juju-bundle: bundle.yaml
```
bundle: kubernetes
applications:
  nms-magmalte:
    charm: magma-nms-magmalte
    # ...
  orc8r-certifier:
    charm: magma-orc8r-certifier
    channel: edge
    scale: 1
    trust: true
    options:
      domain: osm.magma.com
      # ...
relations:
- nms-magmalte:magma-orc8r-certifier
- orc8r-certifier:magma-orc8r-certifier
Day-2 Operations (VNFD)

vnfd:
  description: K8s container deployment of Magma Orchestrator
df:
  - id: default-df
lcm-operations-configuration:
  operate-vnf-op-config:
    day1-2:
      - id: magma-orc-kdu
        config-primitive:
          - name: get-master-admin-credentials
            parameter:
              - name: application-name
                data-type: STRING
                default-value: nms-magmalte
[...]

© ETSI
Hands On Session

Deploying Magma Orchestrator (Juju Bundle)
Download osm-packages

git clone --recurse-submodules -j8
https://osm.etsi.org/gitlab/vnf-onboarding/osm-packages.git

cd osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/magma-orc-scripts
Onboard the Magma Orc8r Packages

Run the following script under `magma-orc-scripts`

```bash
./1-onboard-packages.sh
```

Check the status of onboarded packages:

```bash
osm vnfd-list
osm nsd-list
```
Instantiate Magma Orc8r NS

Run the following script under **magma-orc-scripts**

```
./2-onboard-ns.sh
```

Check the status of NS using the command:

```
osm ns-list
```

Deployment takes around 15-20 minutes.

```
<table>
<thead>
<tr>
<th>ns instance name</th>
<th>td</th>
<th>date</th>
<th>ns state</th>
<th>current operation</th>
<th>error details</th>
</tr>
</thead>
<tbody>
<tr>
<td>magma_orc_ns</td>
<td>0c58f6e-e733-4497-9271-44897179b162</td>
<td>2022-10-17T11:57:41</td>
<td>BUILDING</td>
<td>INSTANTIATING (fa434e7b-b1c0-4898-bcf6-c7b99670fc1c)</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```
Deploying PowerDNS (Helm Chart)
NS Diagram

NS: powerdns_ns

VL: mgmtnet

CP: mgmt-ext

NF: powerdns_knf
KNF Diagram

External CP: mgmt-ext

NF: powerdns_knf

KDU: powerdns
- helm-chart: osm-helm/powerdns
KNF Diagram: K8s Cluster Requirements

K8s cluster
- Version: 1.23

Net 1: mgmtnet

External CP: mgmt-ext

NF: magma_orc8r_cnf
Referencing the Helm Chart (VNFD)

```yaml
vnfd:
  [...]
kdu:
  - name: powerdns
    helm-chart: osm-helm/powerdns
    service:
      - name: webserver-osm-helm-powerdns
```
helm-chart/Chart.yaml (VNFD)

apiVersion: v2
appVersion: v4.3.1
dependencies:
- condition: postgresql.enabled
  name: postgresql
  repository: https://charts.bitnami.com/bitnami
  version: 10.16.2
- condition: mariadb.enabled
  name: mariadb
  repository: https://charts.bitnami.com/bitnami
  version: 10.5.1
name: powerdns
sources:
- http://www.github.com/PowerDNS/
version: 5.0.0
replicaCount: 1
image:
  repository: naps/powerdns
  tag: 4.3.1
  pullPolicy: Always
serviceAccount:
  create: true
service:
  externalTrafficPolicy: ""
  type: LoadBalancer
postgresql:
  enabled: true
  postgresqlUsername: pdns
  postgresqlPassword: pdnspass
  postgresqlPostgresPassword: pdnsadminpass
  postgresqlDatabase: pdns
Day-2 Operations (VNFD)

vnfd:
  description:
  df:
    - id: default-df
      lcm-operations-configuration:
        operate-vnf-op-config:
          day1-2:
            config-primitive:
              - name: add-zone
                execution-environment-ref: powerdns-ee
                parameter:
                  - name: zone_name
                    data-type: STRING
                    default-value: ""

[...]
Add Helm Repository

Go to the `powerdns-scripts` path:

```
cd ~/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts
```

Run the script:

```
./1-add-helm-repo.sh
```

Check the repository:

```
osm repo-list
```

```
ubuntu@osn-1:~/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts$ osm repo-list

<table>
<thead>
<tr>
<th>Name</th>
<th>Id</th>
<th>Type</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>osm-helm</td>
<td>95003bdf-8a3d-432b-8d6b-d0ec5764beef</td>
<td>helm-chart</td>
<td><a href="https://gatitc.github.io/helm-repo/">https://gatitc.github.io/helm-repo/</a></td>
<td>Repository for Powerdns helm Chart</td>
</tr>
</tbody>
</table>
```
Onboard PowerDNS Packages

Run the following script under `powerdns-scripts`

```
./2-onboard-packages.sh
```

Check the status of onboarded packages:

`osm vnfd-list`
`osm nsd-list`
Instantiate PowerDNS NS

Run the following script under `powerdns-scripts`:

```
./3-instantiate-ns.sh
```

Check the status of NS using the command:

```
osm ns-list
```
Check the status of Network Services

WAIT TILL BOTH NS STATUS BECOME READY!

Check the status of NS using the command:

```
ospn ns-list
```

```
<table>
<thead>
<tr>
<th>ns instance name</th>
<th>id</th>
<th>date</th>
<th>ns state</th>
<th>current operation</th>
<th>error details</th>
</tr>
</thead>
<tbody>
<tr>
<td>magma_orc_ns</td>
<td>0c58f8e6e-f730-4497-9271-4489719b162</td>
<td>2022-10-17T11:57:41</td>
<td>INSTANTIATING</td>
<td>(fa434e7b-b1c0-4898-bc16-c7b9967f1c1)</td>
<td>N/A</td>
</tr>
<tr>
<td>powerns_ns</td>
<td>c0501d24-0422-4aba-b744-479c0753df6</td>
<td>2022-10-17T12:02:20</td>
<td>INSTANTIATING</td>
<td>(d84234e1-8a74-4736-86dc-5a8b2ed6aef)</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```
Kubernetes Functions
Why using K8s in OSM?

- Applications based in micro-services
- OSM is, in fact, already running in K8s, both distros and community installer
- Upcoming NFV use cases: 5G Core, uCPE/SD-WAN...
- K8s apps and clusters are essential ingredients for many Edge use cases
How K8s-based apps are modelled?

K8s provides a huge number of high-level service objects, which are the core of its functionality:

- Pod sets*: deployments (+replicasets), statefulsets
- Services: clusterIP, NodePort, Load Balancer
- Storage: persistent volumes, persistent volume claims
- more...

2 ways to deploy a K8s app:

- **Helm Charts**: packaged format + indirect call to the K8s API via Helm
- **Juju Charms and Bundles**: packaged format + indirect call to the K8s API via Juju
What is Helm?

- Helm is the package manager
- Helm manages charts, releases and versions
What are Helm Charts?

- Helm chart describes any Kubernetes resources such as deployment, services etc.
- Uses a rich templating system for making charts generic and highly customizable
- Organized as a collection of files inside a directory
- Charts are versioned
  - **appVersion** -> version of app
  - **kubeVersion** -> define constraints on supported Kubernetes versions
What are Juju Bundles?

- Bundles are collections of charms.
- They represent an entire model, rather than a single application.
- A bundle is a YAML file

Charms + Config + Relations →

```
bundle: kubernetes
applications:
mariadb-k8s:
  charm: cs:~juju/mariadb-k8s-2
  scale: 1
mediawiki-k8s:
  charm: cs:~juju/mediawiki-k8s-3
  scale: 1
  options:
    debug: true
relations:
- mariadb-k8s:server
- mediawiki-k8s:db
```
Requirements of K8s-based Applications

- The K8s cluster
  - Can be created in different ways:
    - Standalone: Openshift, Charmed K8s, Kubeadm
    - As part of a VIM: VMware Cloud PKS, AWS, GCP etc.
    - Can run on Bare Metal, on VMs running in a VIM or runs as a service of platform such as EKS, AKS, GKE.
    - Once created, each cluster provides an identical K8s API depending on version, irrespective of the way it was created.

- Specific versions of K8s or CNI plugins might be required
Modelling KNFs

NF composition specified in the VNF descriptor using **deployment units:**

- Virtual (VDU) = VM
- Physical (PDU) = Physical Node
- Kubernetes (KDU) = K8s app

Modelling in the VNF descriptor

**KDU based on helm charts or juju bundles**

```bash
---rw kdu:kdu* [name]
  +---rw kdu:name     string
  +---rw kdu:description? string
  +---rw (kdu:kdu-model)?
    +---(kdu:helm-chart)
      | +---rw kdu:helm-chart? string
      | +---rw kdu:helm-version? enumeration
      +---(kdu:juju-bundle)
      | +---rw kdu:juju-bundle? string
      +---rw kdu:service* [name]
        | +---rw kdu:name     string
        | +---rw kdu:external-connection-point-ref? string
```

**K8s cluster requirements**

```bash
---rw kdu:k8s-cluster
  +---rw kdu:version* string
  +---rw kdu:cni* enumeration
  +---rw kdu:nets* [id]
    +---rw kdu:id string
```
Life Cycle Management of KDU

**OSM NBI** abstracts the operations required to manage the life cycle of KDU in the context of a **NS**

**OSM operations:**
- NS instantiate
- NS primitive
- NS termination

**Full K8s app lifecycle operations:**
- install
- upgrade
- rollback
- delete
CNF operator runs in a Kubernetes Pod

- A CNF Charm is almost identical to the charms for VNF and PNF workloads
- Runs in a standalone pod if it's proxy charm
- Runs in the same pod as different container if it's native charm
Hands On Session

- Registering Magma Services using PowerDNS
- Accessing Magma Orchestrator GUI
Check the status of Network Services

Check the status of NS using the command:

```
osm ns-list
```
Check the Operational Dashboard
Run Day 2 Actions
Set DNS server in OSM machine

Run the following script under `powerdns-scripts`:

```
./4-setting-dns-server.sh
```

Run the following command to check:

```
cat /etc/resolv.conf | head -1
```

```
ubuntuosm-1:/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts$ cat /etc/resolv.conf | head -1
nameserver 10.0.0.202
```
Download admin operator https certificate

Run the following script under \texttt{magma-orc-scripts}

\texttt{./3-download-admin-operator-cert.sh}

\texttt{admin_operator.pfx} file appears in your current working directory.

\texttt{ls | grep *.pfx}

\texttt{ubuntuosm-1:/osm-packages/HackTest_Demos/OSM-MR13/2.1-KNF/magma-orc-scripts$ ls | grep *.pfx}
\texttt{admin_operator.pfx}

\texttt{ubuntuosm-1:/osm-packages/HackTest_Demos/OSM-MR13/2.1-KNF/magma-orc-scripts$ ls | grep *.pfx}
\texttt{admin_operator.pfx}
Run get-pfx-package-password action

Run the following script under `magma-orc-scripts`

```bash
./4-get-admin-operator-credential.sh
```

pfx_package pass appears in the operation output, please save it.
Run the following script under **magma-orc-scripts**

```
./5-get-magma-orc-gui-credential.sh
```

admin-password and admin-username appear in the operation output, please save it.
Run get-load-balancer-services action

Run the following script under **magma-orc-scripts**

```
./6-get-loadbalancer-services.sh
```

Load Balancer service names and External IP's are visible in the action output as following:

```
echo "Load Balancer service names and External IP's are visible in the action output."
echo "'nginx-proxy': <nginx-proxy External IP> -> master.nms.osm.magma.com"
echo "'orc8r-bootstrap-nginx': <orc8r-bootstrap-nginx External IP> -> bootstrapper-controller.osm.magma.com"
echo "'orc8r-clientcert-nginx': <orc8r-clientcert-nginx External IP> -> controller.osm.magma.com"
echo "'orc8r-nginx-proxy': <orc8r-nginx-proxy External IP> -> api.osm.magma.com"
```

Save the output to be used in the next steps.
Run add-zone action

Run the following script under `powerdns-scripts`:

```
./5-add-zone-action.sh
```

When script asks you the zone, enter the following input:

`osm.magma.com.`
### Output of add-zone action

<table>
<thead>
<tr>
<th>field</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>&quot;be4b48b3-8e95-4d9c-9788-6fadd873dce8&quot;</td>
</tr>
<tr>
<td>operationState</td>
<td>&quot;PROCESSING&quot;</td>
</tr>
<tr>
<td>queuePosition</td>
<td>null</td>
</tr>
<tr>
<td>stage</td>
<td>null</td>
</tr>
<tr>
<td>errorMessage</td>
<td>null</td>
</tr>
<tr>
<td>detailedStatus</td>
<td>null</td>
</tr>
<tr>
<td>statusEnteredTime</td>
<td>1666018788.9984512</td>
</tr>
<tr>
<td>nsInstanceId</td>
<td>&quot;c09581d4-8422-4aba-b744-47d9c8753df6&quot;</td>
</tr>
<tr>
<td>lcmOperationType</td>
<td>&quot;action&quot;</td>
</tr>
<tr>
<td>startTime</td>
<td>1666018788.9984512</td>
</tr>
<tr>
<td>isAutomaticInvocation</td>
<td>false</td>
</tr>
<tr>
<td>operationParams</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;member_vnf_index&quot;: &quot;powerdns&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;kdu_name&quot;: &quot;powerdns&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;primitive&quot;: &quot;add-zone&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;primitive_params&quot;: &quot;{&quot;zone_name&quot;: &quot;osm.nman.com&quot;}&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;lcmOperationType&quot;: &quot;action&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;nsInstanceId&quot;: &quot;c09581d4-8422-4aba-b744-47d9c8753df6&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>isCancelPending</td>
<td>false</td>
</tr>
<tr>
<td>links</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;self&quot;: &quot;/osm/nslc/v1/ns_lcm_ops/1666018788.9984758&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;nsInstanceId&quot;: &quot;c09581d4-8422-4aba-b744-47d9c8753df6&quot;</td>
</tr>
<tr>
<td>_admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;created&quot;: 1666018788.9984758,</td>
</tr>
<tr>
<td></td>
<td>&quot;modified&quot;: 1666018788.9984758,</td>
</tr>
<tr>
<td></td>
<td>&quot;projects_read&quot;: [</td>
</tr>
<tr>
<td></td>
<td>&quot;25cb68c47886454cbbae07d524081d7&quot;</td>
</tr>
<tr>
<td></td>
<td>],</td>
</tr>
<tr>
<td></td>
<td>&quot;projects_write&quot;: [</td>
</tr>
<tr>
<td></td>
<td>&quot;25cb68c47886454cbbae07d524081d7&quot;</td>
</tr>
<tr>
<td></td>
<td>],</td>
</tr>
<tr>
<td></td>
<td>&quot;worker&quot;: &quot;13a604de97c&quot;</td>
</tr>
</tbody>
</table>
Run add-domain action

Run the following script under powerdns-scripts:

`.6-add-domain-action.sh`

Use the output of get-load-balancer-services action.

Replace the `<External IP>` with the exact service IP:

- `<nginx-proxy External IP>` -> `master.nms.osm.magma.com`
- `<orc8r-bootstrap-nginx External IP>` -> `bootstrap-controller.osm.magma.com`
- `<orc8r-clientcert-nginx External IP>` -> `controller.osm.magma.com`
- `<orc8r-nginx-proxy External IP>` -> `api.osm.magma.com`
Run add-domain action

When script asks you the inputs, register domain name by entering inputs zone, domain and ip.

Run the action 4 times to register all domain names.

Sample domain name registration using action:

api.osm.magma.com
ZONE=osm.magma.com.
DOMAIN=api.
IP=<orc8r-nginx-proxy External IP>
Output of add-domain action

<table>
<thead>
<tr>
<th>field</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id</td>
<td>&quot;ca247f6-8faa-46ee-b396-19efb2998131&quot;</td>
</tr>
<tr>
<td>id</td>
<td>&quot;ca247f6-8faa-46ee-b396-19efb2998131&quot;</td>
</tr>
<tr>
<td>operationState</td>
<td>&quot;PROCESSING&quot;</td>
</tr>
<tr>
<td>queuePosition</td>
<td>null</td>
</tr>
<tr>
<td>stage</td>
<td>null</td>
</tr>
<tr>
<td>errorMessage</td>
<td>null</td>
</tr>
<tr>
<td>detailedStatus</td>
<td>null</td>
</tr>
<tr>
<td>statusInterfaced</td>
<td>1666011765.8786762</td>
</tr>
<tr>
<td>nsInstanceId</td>
<td>&quot;c6590d24-6422-4aba-b744-47d9c1753df6&quot;</td>
</tr>
<tr>
<td>lcnOperationType</td>
<td>&quot;action&quot;</td>
</tr>
<tr>
<td>starttime</td>
<td>1666011765.8786762</td>
</tr>
<tr>
<td>lsAutomaticInvocation</td>
<td>false</td>
</tr>
<tr>
<td>operationParams</td>
<td>{}</td>
</tr>
<tr>
<td>member_vnf_index</td>
<td>&quot;powerdns&quot;</td>
</tr>
<tr>
<td>kdu_name</td>
<td>&quot;powerdns&quot;</td>
</tr>
<tr>
<td>primitive</td>
<td>&quot;add-domain&quot;</td>
</tr>
<tr>
<td>primitive_params</td>
<td>&quot;{&quot;zone_name&quot;: &quot;osm.nagna.com&quot;, &quot;subdomain&quot;: &quot;master.nns.&quot;}&quot;, &quot;ip&quot;:</td>
</tr>
<tr>
<td></td>
<td>[&quot;16.0.0.204&quot;]</td>
</tr>
<tr>
<td>lcnOperationType</td>
<td>&quot;action&quot;</td>
</tr>
<tr>
<td>nsInstanceId</td>
<td>&quot;c6590d24-6422-4aba-b744-47d9c1753df6&quot;</td>
</tr>
<tr>
<td>isCancelPending</td>
<td>false</td>
</tr>
<tr>
<td>links</td>
<td>{}</td>
</tr>
<tr>
<td>_created</td>
<td>1666011765.8787055</td>
</tr>
<tr>
<td>_modified</td>
<td>1666011765.8787055</td>
</tr>
<tr>
<td>_projects_read</td>
<td>[]</td>
</tr>
<tr>
<td>projects_read</td>
<td>[&quot;25cb68c47886454cc3bbae79d524881d7&quot;]</td>
</tr>
<tr>
<td>projects_write</td>
<td>[&quot;25cb68c47886454cc3bbae79d524881d7&quot;]</td>
</tr>
<tr>
<td>_worker</td>
<td>&quot;136884de697c&quot;</td>
</tr>
</tbody>
</table>
Test domain name resolution

Run the following script under powerdns-scripts:

```
./7-test-dns-record.sh
```

When script asks you the input, add domain names registered at previous step.

Run the script 4 times to check all domain names.

Sample input:

```
master.nms.osm.magma.com
```
Output of domain name resolution test

```
ubuntu@osn-1:~$ ./osn-packages/HackFest_Demos/OSM-MR13/2.1-KNF/powervms-scripts$ ./7-test-dns-record.sh
Enter DNS RECORD: master.nns.osn.magna.com
You entered RECORD master.nns.osn.magna.com
Testing record

@ >>> Dig 9.16.1-Ubuntu >>> master.nns.osn.magna.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37886
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;@master.nns.osn.magna.com. IN A

;; ANSWER SECTION:
master.nns.osn.magna.com. 86400 IN A 10.8.0.284

;; Query time: 4 msec
;; SERVER: 10.0.0.202#53(10.0.0.282)
;; WHEN: Mon Oct 17 13:32:57 UTC 2022
;; MSG SIZE  rcvd: 69
```
Access Magma Orc8r GUI

Firefox browser is already installed in your remote machine.

Open the browser and import the admin_operator.pfx using certificate import.

When it asks you a password, please enter pfx_package pass.

Try to reach following URL using your browser: https://master.nms.osm.magma.com.

Use the admin-username and admin-password to login.

Import certificate by following steps in your browser:
Firefox settings -> Privacy & Security -> Certificates
Access Magma Orc8r GUI

![Magma Orc8r GUI screenshot]

<table>
<thead>
<tr>
<th>Name</th>
<th>Network IDs</th>
<th>Tabs</th>
</tr>
</thead>
<tbody>
<tr>
<td>fb-test</td>
<td></td>
<td>inventory, workorders &amp; 2 others</td>
</tr>
<tr>
<td>magma-test</td>
<td></td>
<td>nms</td>
</tr>
<tr>
<td>master</td>
<td></td>
<td>admin</td>
</tr>
</tbody>
</table>

© ETSI
Run scale out/scale in actions

Run the following script under powerdns-scripts:

```
./12-scale-out.sh
```

```
ubuntu@osn-10:/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts$ ./12-scale-out.sh
Enter replicaCount number to scale as number greater than 1:3
You entered NUM 3
Scaling out

Showing action status
Check the action status using osm ns-op-show 1d7abe3d-c6be-4ddf-bf65-8ff081935128 --literal | yq .operationState

PROCCESSING

Showing replicaCount number
Check the replicaCount using osm vnf-show 812a7f40-abee-4e63-9dd6-7d180e0e5030 --kdu powerdns | yq .config.replicaCount

ubuntu@osn-10:/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts$ osm vnf-show 812a7f40-abee-4e63-9dd6-7d180e0e5030 --kdu powerdns | yq .config.replicaCount
```

```
./12-scale-in.sh
```

```
ubuntu@osn-10:/osm-packages/Hackfest_Demos/OSM-MR13/2.1-KNF/powerdns-scripts$ ./12-scale-in.sh

Rolling back

Showing action status
Check the action status using osm ns-op-show e1ccfb4-051b-438f-a57b-720083ef3e2a

PROCCESSING

Showing replicaCount number
Check the replicaCount number using osm vnf-show 812a7f40-abee-4e63-9dd6-7d180e0e5030 --kdu powerdns | yq .config.replicaCount
```

© ETSI
Join us!

ETSI members, non-members, individual developers and users.

Learn how to join
Follow us!
Thank You!