

# DMaaP Edge Deployment

NOTE: Originally, the 5G Edge Use Case was planned for Dublin. But that was deferred because an ONAP-wide multi-site (routing) solution was not implemented. So, this page represents notes for the Work In Progress...as DMaaP team experiments with various approaches.

The 5G Use Case has dependencies on DMaaP services. The 5G components (Data File Collector and 3GPP PM Mapper) will be deployed at the Edge, so DMaaP services should be made available to them to avoid a data flow path through the Central Kubernetes cluster. This 5G Use Case relies on both Data Router and Message Router.

This Use Case will help flesh out the requirements and techniques for DMaaP Edge deployments.

## Definitions

1. Dublin introduces the notion of a multi-cloud deployment consisting of a single "central" kubernetes High Availability (HA) installation, and 0 or more "edge" kubernetes installations.
2. Geo-redundancy applies to a multi-site central k8s deployment. This shouldn't be confused with a multi-site ONAP deployment consisting of central and edge sites.

## Assumptions

1. DMaaP will maintain a single set of Helm Charts in the oom/kubernetes repo. Said a different way, we will strive to not maintain separate DMaaP Central charts and DMaaP Edge charts.
  - a. The DMaaP Helm charts will continue to be maintained as a single oom kubernetes directory, with sub-directories for each component.
2. The "central" site will always be deployed before any edge sites.
  - a. The Edge deployment (and operation) will rely on central ONAP services (e.g. AAF)
  - b. This will allow a human (at least) to capture any values representing central deployment details (such as a K8S gateway IP address)
3. All DMaaP components will continue to be deployed in the "central" k8s. The details of what components will be deployed at any Edge, and how it will be deployed are the subject of this page.
4. An "edge" site can be deployed any time after the "central" site.
5. Not all edge sites need be deployed at the same time.
6. As a Platform Service, DMaaP will be deployed before any application/microservice.
7. SSL Server Certificates will be created in advance of deployment, and not generated at deployment time. (This is a feature for EI Alto)
8. By convention, the kubernetes cluster name will be used as the name of the site.

## Requirements

1. A Central-deployed DMaaP component must be able to route to an Edge-deployed component, and distinguish between the same component deployed at different Edge sites. Examples include:
  - a. dr-prov periodically sends provisioning info to each dr-node
  - b. A centrally-deployed dr-node may transfer a file to an Edge-deployed dr-node for delivery to a subscriber in that Edge, based on an egress rule
  - c. A central mirrormaker subscribes to an Edge-deployed message-router kafka
2. An Edge-deployed DMaaP component must be able to route to a central-deployed service. Examples include:
  - a. dr-node periodically syncs with dr-prov
  - b. dr-node authenticates publish requests using aaf
  - c. message-router authenticates client requests using aaf
  - d. dbc-client makes request to dmaap-bc API during post-install provisioning
  - e. Edge mirrormaker subscribes to central message-router kafka
3. Localized DR Routing between a Data File Collector (DFC) and a PM Mapper deployed in the same Edge X.
  - a. *Localized DR Routing* means DR Node is deployed in the same Edge site so data doesn't need to leave the site.
  - b. DFC will be a publisher to a feed provisioned at deployment time.
  - c. PM Mapper will be a subscriber provisioned at deployment time.
  - d. The feed should be unique per site so that when there are multiple sites, PM Mapper only receives its locally produced data.
4. Localized messaging from PM Mapper and DFC. This will signal DFC that a file was processed.
  - a. *Localized messaging* implies a Message Router instance in the same edge location.
  - b. PM Mapper will be a publisher provisioned at deployment time
  - c. DFC will be a subscriber provisioned at deployment time.
  - d. Communication will utilize an authenticated topic in the MR deployed in the same edge site.
    - i. PM Mapper and DFC will use AAF credentials to authenticate.
    - ii. PM Mapper identity will be authorized to publish on the topic
    - iii. DFC identity will be authorized to subscribe on the topic
5. Inter-site messaging from PM Mapper to VES perf3gpp
  - a. *Inter-site messaging* means sending a message from an edge location publisher to a central location subscriber.
  - b. PM Mapper, deployed at Edge, will be a publisher using AAF credentials
  - c. VES perf3gpp, deployed in Central, will be a subscriber using AAF credentials
  - d. Communication will utilize an authenticated topic on the MR deployed in the same edge site.
    - i. PM Mapper and DFC will use AAF credentials to authenticate.
    - ii. PM Mapper identity will be authorized to publish on the topic
    - iii. VES perf3gpp identity will be authorized to subscribe on the topic
  - e. Furthermore, messages on this topic will be replicated to the central MR instance.
  - f. Are there any other subscribers? (esp, are there any other at edge?)

## Solution Options for Dublin

NOTE: planning for Dublin assumed that the AAI component would provide an API that served as a registry of each ONAP site. This did not happen.

This section is based on a discussion with [Jack Lucas](#) about possible approaches that we might consider within the Dublin feature set.


Ways to route to a k8s service in another k8s cluster:

1. Extend the configuration of the Jack's proxy to include DMaaP services. Note: Current capability will route from edge to central. (See Jack's demo from ~ 0:29:40)
  - a. Include central deployed DMaaP services with existing node ports in proxy config: dr-prov, message-router, dmaap-bc (Completed: see <https://gerrit.onap.org/r/#/c/87710/>)
  - b. Expose central deployed DMaaP service on node port and add to proxy configuration: dr-node (Completed: see <https://gerrit.onap.org/r/#/c/87710/>)
  - c. NOTE: proxy can subsequently route by FQDN (for HTTP only).
2. K8S External Service. Deploy services at Edge which map to central services.
  - a. REF: <https://cloud.google.com/blog/products/gcp/kubernetes-best-practices-mapping-external-services>
3. Add entries for central services into /etc/hosts on Edge pods so they can route properly
4. Provision some external DNS service that is able to resolve to required IP addresses in other k8s cluster
  - a. Will require establishing a convention for FQDN. eg. <Release>-<service>.<namespace>
  - b. Convention should leverage assumptions of using same value for Release and k8s cluster name.
5. Determine how clients can specify FQDN (service name) but designate IP address to use.
  - a. See --resolve option in curl for example of how this might work.
6. Apply k8s thinking to DMaaP component design:
  - a. Abandon the DR publish redirect protocol and simply use dr-node service instead.
    - i. if dr-node is local to the cluster, then client will route to local dr-node pod for publishing (which is desired)
    - ii. if dr-node isn't local to cluster, then client will route to central dr-node via proxy (fallback)
  - b. Change dr-prov algorithm for distributing prov data to dr-node so dr-prov doesn't need to know how to address every pod
    - i. consider simple periodic polling by dr-node
    - ii. consider using an MR topic to trigger dr-node to poll for prov data
  - c. migrate to ELK design for logging, removes need for dr-prov to gather logs from each dr-node. (already in progress)

Upon review of this list, some concern was expressed about entertaining options that involve code changes given where we are in Dublin. Also, there is a desire for being directionally consistent with future ONAP OOM plans.

Subsequently, [Fiachra Corcoran](#) inquired at OOM meeting about approaches consistent with future directions, and learned:


- intent is to utilize Ingress Controllers
- RKE deployment has Ingress Controller support (although selection of which Ingress Controller technology is not finalized)
- Some useful notes:
  - From Michael O'Brien(Amdocs, LOG) to Everyone: 10:09 AM  
default rke ingres [https://git.onap.org/oom/tree/kubernetes/contrib/tools/rke/rke\\_setup.sh#n177](https://git.onap.org/oom/tree/kubernetes/contrib/tools/rke/rke_setup.sh#n177) ingress: rancher/nginx-ingress-controller: 0.21.0-rancher3 ingress\_backend: rancher/nginx-ingress-controller-defaultbackend:1.4-rancher1
  - From Michael O'Brien(Amdocs, LOG) to Everyone: 10:20 AM

 **OOM-1598** - Document a Highly-Available K8s Cluster Deployment (RKE 0.2.1 / K8S 1.13.5 / Helm (2.12.3 - not 2.13.1) / Docker 18.09.5) CLOSED

Document a Highly-Available K8s Cluster Deployment (RKE 0.2.1 / K8S 1.13.5 / Helm (2.12.3 - not 2.13.1) / Docker 18.09.5)

- Much of this is now under discussion in the Edge Automation Working Group. (meets wed @11am EST)
- Also, [Fiachra](#) and [Mike Elliott](#) agreed to continue discussion on how DMaaP POC might proceed. Possible meeting next week.

## Open Issues

REF	Status	Discussion
1	Open	<p>DNS Update for inter-site routing</p> <p>We have several examples of an edge component which needs to communicate to a central service. Mike suggested that edge DNS might be updated such that edge clients could resolve to central services. This might satisfy a common need across several components. e.g. access to central AAF comes to mind</p> <p>05/02:</p> <p>Another alternative was demoed by DCAE where an nginx container is deployed at edge site which proxies service traffic to the relevant NodePort on the central k8s cluster.</p> <p>This may be suitable for some of DMaaP components (as a POC) but not a preferred solution.</p> <p>Work is ongoing in OOM to provide this (with input from the community)</p> <p> <b>OOM-1572</b> - Multi-cluster support via Helm deploy <span>CLOSED</span></p>

2	Open	<p>Location discovery</p> <p>Bus Controller manages dcaeLocations as the name of different sites. What mechanism can be used to:</p> <p>a) register dcaeLocations when each k8s cluster is deployed.</p> <p>b) serve as an attribute when MR and DR clients are provisioned. Current expectation is that there is some k8s info in A&amp;AI API that might be useful.</p> <p>05/02:</p> <p>Agreement from DCAE on requirement to involve all ONAP components (AAI, OOF, etc) to find a suitable solution here.</p> <p>Defined use-case defined here <a href="#">OOM-1579 - Central/Edge indicator convention in Helm charts</a> <span>CLOSED</span></p>
3	Closed	<p>Relying on Helm chart enabled flag</p> <p>2/12:</p> <p>"Mike,</p> <p>Last week we discussed using a helm configuration override file to control which components get deployed at edge.</p> <p>The idea being we would set enabled: false for a component that shouldn't be deployed.</p> <p>But dmaap chart actually consists of several sub-charts, each of these sub-charts correspond to a specific dmaap component which we may want to deploy at edge or not.</p> <p>So, curious if you know the syntax for this – I haven't been able to find a reference for how <i>enabled</i> is actually used, and I don't see that value referenced in our charts so not clear what is reading it.</p> <p>Wondering if our edge config override would be something like:</p> <pre> dmaap:   dmaap-message-router:     enabled: true   dmaap-bus-controller:     enabled: false   dmaap-dr-prov:     enabled: false   dmaap-dr-node:     enabled: true </pre> <p>or, do charts for our individual components need to be top level directories under oom/kubernetes in order to use the <i>enabled</i> flag?"</p> <p>2/13: From Mike Elliot:</p> <p>"I've been trying to allow for the conditional control over the dr-prov and dr-node as well, with no success.</p> <p>Still investigating options for this. Hope to have a solution on this by EOD."</p> <p>05/02:</p> <p>Current chart structure allows deployment of individual components. (BC, MR, DR).</p> <p>One caveat to this is a dependency on AAF being reachable by BC &amp; MR. (DR soon to follow)</p> <p>See the <a href="#">DMaaP Deployment Guide - Dublin</a> for more details.</p>

4	Open	<p>05/02:</p> <p>Helm chart edge deploy.</p> <ul style="list-style-type: none"> <li>POC procedure demoed using multiple <code>"kube-config --contexts"</code> to target the edge site/cluster during helm deploy. (Inter cluster security may come into play here also)</li> </ul> <p>"edge charts" may require several override params to cater for the following.</p> <ol style="list-style-type: none"> <li>dcaeLocation (see issue 2)</li> <li>pod specs - size, resources, etc</li> <li>readiness configuration?</li> <li>potential service endpoint changes/proxies?</li> </ol>
5	Open	<p>05/02:</p> <p>Need to identify if all of the required services (logstash, AAF, dr-node, mr-kafka, etc) have exposed NodePorts available for bi-directional traffic between sites.</p>

## Development

- Helm configuration overrides will be collected in a single file (e.g. dmaap-edge.yaml) and delivered to oom/kubernetes/onap/charts/resource/environments. Examples of what kinds of overrides will be present in this file include:
  - Setting the standard *enabled* indicator to true for dmaap, but false for other components.

```

dmaap:
  enabled: true

```
  - Setting an *edge* indicator to drive any edge-specific logic. TBD if this is really useful - hopefully other overrides in this file are edge specific.
  - Setting the values for a central service which may be needed at the edge. Known examples include:
    - Message Router must be configured to access the central AAF instance. (DR Node may have this requirement in the near future)
    - Data Router Node must be configured to access the central DR Prov
    - Both MR and DR Node must register with central Bus Controller
  - Setting scaling values appropriate to the edge. e.g. perhaps a single kafka broker is appropriate at the edge
- DMaaP Chart changes
  - Reorder charts:
    - Bus Controller must be up and running if other components are going to register with it. [Jira to remove any dependencies on MR.](#)
    - MR
    - Mirror Maker
    - DR Prov
    - DR Node (DR Prov must be up for Node to retrieve provisioning info)
  - Post-install hooks:
    - Bus controller:
      - POST <central dmaap-bc>/webapi/dmaap
      - POST <central dmaap-bc>/webapi/dcaeLocation (for central)
    - MR:
      - POST <central dmaap-bc>/webapi/mr\_clusters [DMAAP-534](#) [Jira to add kafka brokers to endpoint](#)
    - DR Node
      - POST <central dmaap-bc>/webapi/dr\_node [DMAAP-534](#)

## Step-by-step guide

This outlines the approach for solving this Edge deployment, and will undoubtedly be refined over time.

- Central K8S Deployment
- Central DMaaP Deployment
  - Use k8s cluster name as the Release. e.g. "central"
  - Deploy aaf
  - ~~Deploy aaf~~
  - Deploy dmaap
  - Deploy dcae
  - Deploy VES perf3gpp via dcae
- Edge K8S Deployment
  - ~~Register Edge K8S deployment in AAF (how?)~~
  - Add dcaeLocation (for new Edge K8S) to DMaaP Bus Controller
- Edge DMaaP Deployment
  - Update dmaap-edge.yaml configuration override file with values from central
  - Use k8s cluster name as the Release. e.g. "edge1"
  - deploy dmaap
  - deploy PM Mapper via dcae

## Presentation

The following deck can be used to discuss the concepts on this page.



DMaaP Edge - Dublin.pptx



## Related articles

[DCAE MultiSite demo](#)

[5G Use Case \(R4 Dublin\)](#)

- [Deploy OOM and SDC \(or ONAP\) on a single VM with microk8s - Honolulu Setup](#)
- [Setup helm3 for oom](#)
- [model-driven DMaaP Agent](#)
- [DMaaP Edge Deployment](#)
- [Cloud Agnostic Intent and Mappings](#)