

Istio for DCM

The Distributed Cloud Manager (DCM) uses Istio to manage each logical cloud. It installs a control plane per cluster per logical cloud and it uses [Istio Replicated Control Planes](#) for this. For each logical cloud, it will install istio in an istio namespace for that logical cloud

Steps Required to Install Istio for DCM

- **Install Istio in custom namespace**

After creating the custom namespace and the secrets using the instructions in the Istio Replicated Control Planes. Add the following lines to the Istio manifest file before applying the Istio Manifest in order to install it in the custom namespace

```
#####Original File#####
values:
  global:
    # Provides dns resolution for global services
    podDNSSearchNamespaces:
      - global
      - "{{ valueOrDefault .DeploymentMeta.Namespace \"default\" }}.global"

#####Modified File#####
values:
  global:
    istioNamespace: istio-lcl
    # Provides dns resolution for global services
    podDNSSearchNamespaces:
      - global
      - "{{ valueOrDefault .DeploymentMeta.Namespace \"default\" }}.global"
```

- **Add stub domain with suffix for the logical cloud to the kube-system coredns configmap**

In order to tell the kubernetes coredns to forward traffic to the Istio coredns, a stub domain is added to the kubernetes coredns configmap as shown in [Setup DNS](#). The default suffix for istio is "*global*". With the DCM, we want to have a different suffix for each logical cloud, so we need to add a similar entry for each logical cloud

Example

The example below shows how the kubernetes configmap will look with the default global stub domain

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
data:
  Corefile: |
    .:53 {
      errors
      health
      ready
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        upstream
        fallthrough in-addr.arpa ip6.arpa
      }
      prometheus :9153
      forward . /etc/resolv.conf
      cache 30
      loop
      reload
      loadbalance
    }
    global:53 {
      errors
      cache 30
      forward . $(kubectl get svc -n istio-system istiocoredns -o jsonpath={.spec.clusterIP}):53
    }
```

Assuming we have two logical clouds lc1 and lc2 with their istio namespaces installed in istio-lc1 and istio-lc2 respectively, then we would add the following stub domains to the kubernetes coredns configmap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
data:
  Corefile: |
    .:53 {
      errors
      health
      ready
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        upstream
        fallthrough in-addr.arpa ip6.arpa
      }
      prometheus :9153
      forward . /etc/resolv.conf
      cache 30
      loop
      reload
      loadbalance
    }
    lc1:53 {
      errors
      cache 30
      forward . $(kubectl get svc -n istio-lc1 istiocoredns -o jsonpath={.spec.clusterIP}):53
    }
    lc2:53 {
      errors
      cache 30
      forward . $(kubectl get svc -n istio-lc2 istiocoredns -o jsonpath={.spec.clusterIP}):53
    }
```

- **Enable forwarding Traffic to custom suffix**

By default, when istio is installed, it only allows forwarding traffic to services in the remote cluster with the "*global*" suffix. In order to enable istio to forward traffic, we need to change the suffix in some of its resources. Get the following istio resources and edit them (kubectl edit <resource-type> <resource-name> -n <resource-namespace>), change all occurrences of the global suffix in each of these resources to your custom suffix.

1. Add lc1 stubdomain in kube-system coredns cm
2. Change istio coredns cm to new suffix
3. Edit envoyfilter to use customsuffix
4. Edit istio ingressgateway to use custom suffix
5. Edit istio destination rule to use custom suffix
6. Make sure service entry uses the right suffix

Note: I have only tested using one custom suffix (e.g) lc1 at a time, I have not tested adding multiple custom suffixes.

The Link below can also provide guidance on how to enable forwarding traffic to custom suffix, the method i used above is slightly different though Guide: <https://gist.github.com/aattuluri/baf6b8aac10471acc73aa1d6262a65bb>.