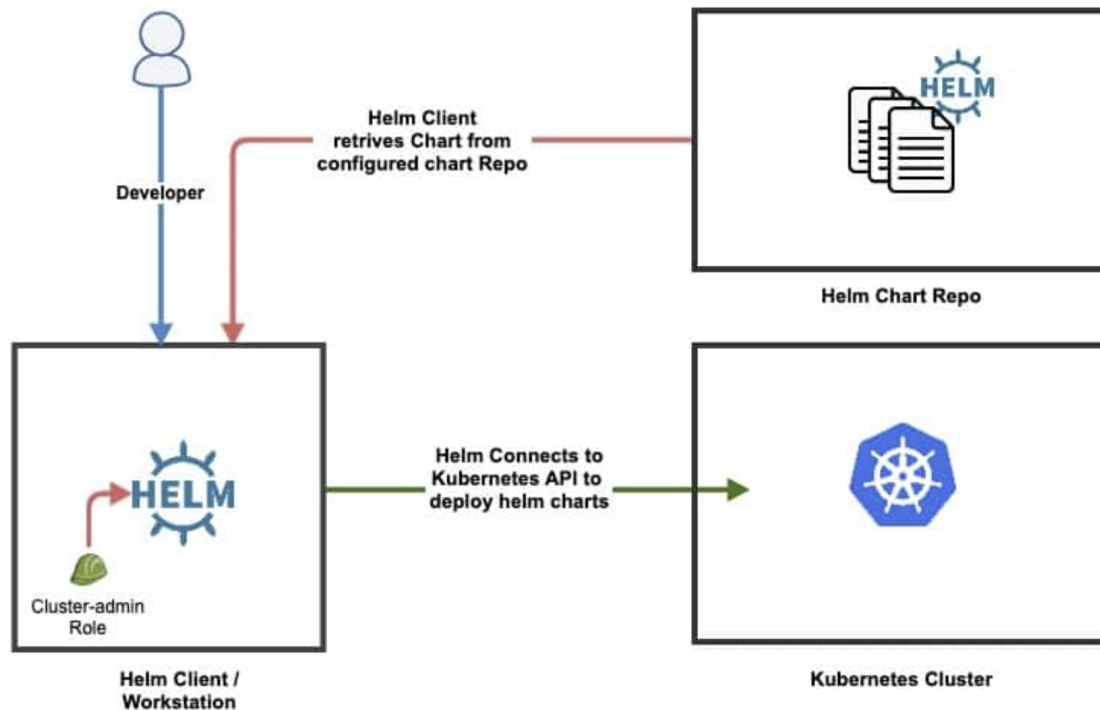


# The CLAMP Kubernetes Participant

The CLAMP Kubernetes Participant performs Helm chart installation and LCM for Kubernetes (K8S) microservices that take part in Control loops. It implements the participant-Intermediary API to receive events from the CLAMP runtime and then interacts with Helm CLI. It acts as a wrapper around the Helm CLI to manage Helm charts and K8s pods deployed in the cluster.

## Helm 3 overview:



In Helm3, the Helm CLI acts as an interface towards a Kubernetes cluster, and allows the user to deploy and manage helm charts. It also supports configuration of multiple chart repositories for the helm client and enables access to charts from those repos for installation of K8S microservices. The repositories can be local chart servers running on the same machine or third party chart servers running elsewhere. The URL of a chart server can be configured on the Helm repo list to permit access to Helm charts hosted in those repos. Users can add repositories and push Helm charts to the repositories via Helm CLI commands.

**Example: Configured chart repos and the available charts are listed via the Helm CLI**

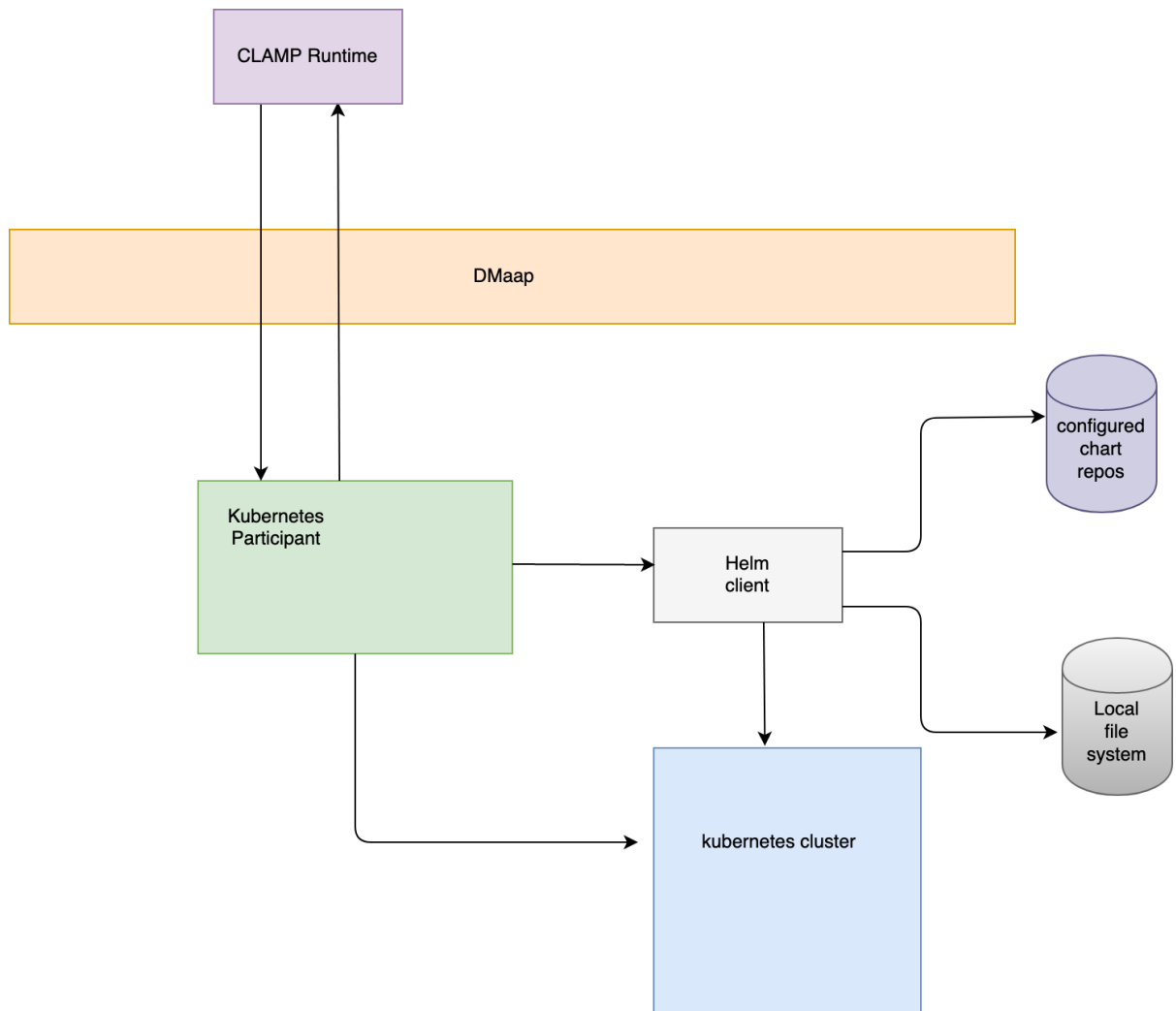
```
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$ helm repo list  
NAME          URL  
nginx-stable  https://helm.nginx.com/stable  
local         http://127.0.0.1:8879  
chartMuseum   http://localhost:8084  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$ helm search repo  
NAME                  CHART VERSION  APP VERSION  DESCRIPTION  
chartMuseum/hello     0.1.0          1.16.0       A Helm chart for Kubernetes  
nginx-stable/nginx-ingress 0.9.1          1.11.1       NGINX Ingress Controller  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$  
ramesh@ubuntu-mac:~$
```

**CLAMP Kubernetes Participant Architecture in Istanbul:**

#### Prerequisites for Kubernetes participant:

- Kubernetes cluster running.
- Currently, the operator is expected to provide the repository information where the helm chart is available, via TOSCA /REST end point or ensure that the charts are available on the local storage directory of kubernetes-participant. (This directory is a preconfigured path specified on the participant config that acts as a local chart repository). Charts can be onboarded to local chart repository via REST endpoints.

**Note:** When using the Kubernetes participant in docker container, the config file of the required kubernetes cluster should be copied to the k8s-participant's docker container's home directory under /.kube folder in order to make the participant work with the external cluster.



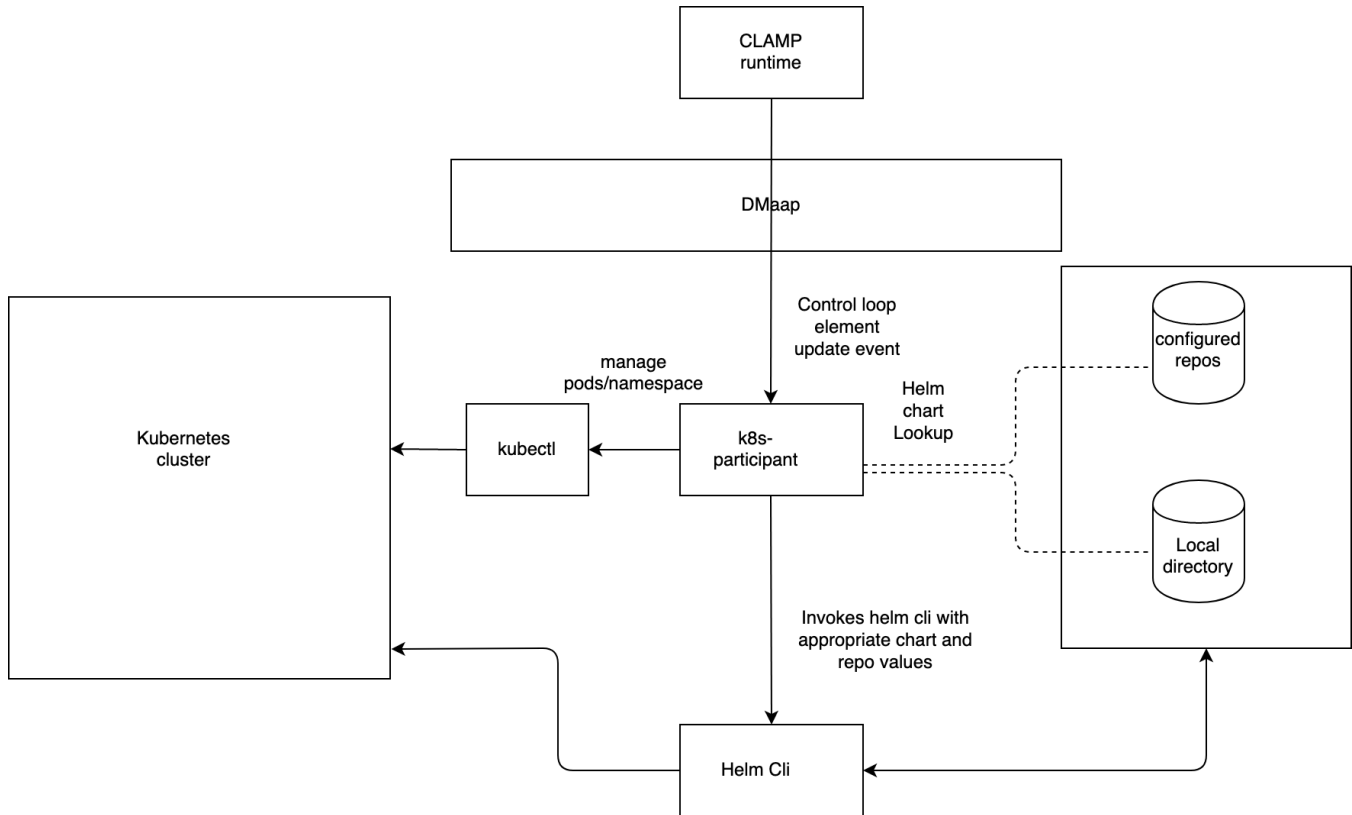
In the Istanbul release, the CLAMP Kubernetes participant supports the following methods for installation of helm charts via CLAMP:

- Installation of a Helm chart that is present in the same local file system as where the kubernetes participant is hosted
- Installation of Helm charts from any remote helm repositories. (The remote repository details needs to be passed on via TOSCA or REST API to the kubernetes participant)

The CLAMP Kubernetes participant acts as a mediator between the CLAMP runtime and the Helm Client.

While commissioning a control loop, the Helm chart parameters are passed via TOSCA template to the control loop runtime database. And when the control loop is instantiated, the Kubernetes participant receives control loop element update event from the CLAMP runtime. It then invokes the Helm client running on the host machine to install the corresponding Helm charts associated with the affected control loop elements. The Kubernetes participant gets the parameters (*chart name, version, release name and namespace*) of the chart from the control loop runtime. If the repository of the chart had not been specified in the TOSCA, it performs a "chart lookup" on all the configured repositories in the Helm client as well as on the local chart directory where the Helm charts may be manually onboarded by the operator. It fetches the appropriate repository info and installs the chart via the helm CLI.

The Kubernetes participant takes care of creating a namespace on the cluster if required, fetching the Helm chart from the available repositories, installing and uninstalling the chart in to the cluster. In upcoming releases, additional options will be supported in CLAMP for onboarding Helm charts to the repositories via TOSCA (both configured Helm repos and local chart directory) before instantiation. (Under discussion)



The code block below shows a Sample TOSCA service template passed during commissioning of control loops. Charts can be commissioned via TOSCA in the following mentioned ways:

- Chart parameters: chartId(name, version), releaseName and namespace are mandatory in TOSCA.
- Repository can be specified with either remote helm repository or any local directory where the helm charts are available.

**(Note:** Repository name is an optional parameter in control loop TOSCA template. If not specified, the Kubernetes participant will do a look up on the local chart storage and configured helm repos)

## Tosca template

```
org.onap.domain.database.HelloWorld_K8SMicroserviceControlLoopElement:
# Chart from local file system (pre onboarded via REST API).
version: 1.2.3
type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
type_version: 1.0.0
description: Control loop element for the K8S microservice for Hello World
properties:
provider: ONAP
participant_id:
name: org.onap.k8s.controlloop.K8SControlLoopParticipant
version: 2.3.4
chart:
releaseName: helloworld
chartId:
  name: hello
  version: 1.0
namespace: onap

org.onap.domain.database.PMSH_K8SMicroserviceControlLoopElement:
# Chart from any remote repository
version: 1.2.3
type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
type_version: 1.0.0
description: Control loop element for the K8S microservice for PMSH
properties:
provider: ONAP
participant_id:
name: org.onap.k8s.controlloop.K8SControlLoopParticipant
version: 2.3.4
chart:
release_name: pmsmicroservice
chartId:
  name: dcae_pms
  version: 8.0
repository:
  repoName: chartMuseum
  address: 172.125.12.1
  port: 8082
  protocol: http
  username: username
  password: password
namespace: onap

org.onap.domain.database.Local_K8SMicroserviceControlLoopElement:
# Chart installation without passing repository name (chart lookup happens on local chart storage and
preconfigured helm repos)
version: 1.2.3
type: org.onap.policy.clamp.controlloop.K8SMicroserviceControlLoopElement
type_version: 1.0.0
description: Control loop element for the K8S microservice for any chart
properties:
provider: ONAP
participant_id:
name: org.onap.k8s.controlloop.K8SControlLoopParticipant
version: 2.3.4
chart:
release_name: nginxms
chart_name: nginx-ingress
version: 0.9.1
namespace: onap
```

The Kubernetes participant receives messages through the participant-intermediary common code, and handles them by invoking the Kubernetes Open API via the Helm client. For example, When a ControlLoopUpdate message is received by Kubernetes participant, the control loop element state changed from **UNINITIALISED** to **PASSIVE**, Kubernetes-participant triggers Kubernetes Open API and installs the HELM charts on the cluster.

## Run CLAMP Kubernetes Participant command line using Maven

```
mvn spring-boot:run -Dspring-boot.run.arguments="--topicServer=localhost"
```

## Run CLAMP Kubernetes Participant command line using Jar

```
java -jar -DtopicServer=localhost target/policy-clamp-participant-impl-kubernetes-6.1.2-SNAPSHOT.jar
```

### REST APIs on Kubernetes participant:

Kubernetes participant can also be installed as a standalone application which exposes REST endpoints for onboarding, installing, uninstalling helm charts from local chart directories.

**k8s-participant** Chart Controller

DELETE /onap/k8sparticipant/helm/chart/{name}/{version} Delete the chart

GET /onap/k8sparticipant/helm/charts Return all Charts

POST /onap/k8sparticipant/helm/install Install the chart

POST /onap/k8sparticipant/helm/onboard/chart Onboard the Chart

POST /onap/k8sparticipant/helm/repo Configure helm repository

DELETE /onap/k8sparticipant/helm/uninstall/{name}/{version} Uninstall the Chart

## K8s-participant User guide (PMSH usecase)

