# Getting Started for EMCO v2

## Download and Build Docker Images

Download EMCO code from Github and build docker container. Currently all services are in the same Docker container. Once built Docker image can be pushed to docker hub.

**Download**

```
$ git clone https://git.onap.org/multicloud/k8s
$ cd k8s
$ docker build --rm -f build/Dockerfile . -t <tag>
```

## Deploying EMCO in Kubernetes Cluster

### Using Helm Charts

Follow the steps in here https://github.com/onap/multicloud-k8s/tree/master/deployments/helm/v2/onap4k8s to install EMCO. Update image: rtsood/emco:stable in values.yaml file for all the services to the name of the image built in the previous step.

### Using Kubectl (for developers)

Yamls for installing EMCO can be found here: https://github.com/onap/multicloud-k8s/tree/master/deployments/kubernetes. Update the image: rtsood/emco:stable in the onap4k8s.yaml for all the services to the name of the image built in the previous step.

**Installing EMCO**

```
1. Create namespace
$ kubectl create namespace emco

2. Install Databases (Etcd and Mongo)
$ kubectl apply -f onap4k8sdb.yaml -n emco

3. Install Emco Microservices
$ kubectl apply -f onap4k8s.yaml -n emco
```

## Running Prometheus+Collectd example with EMCO

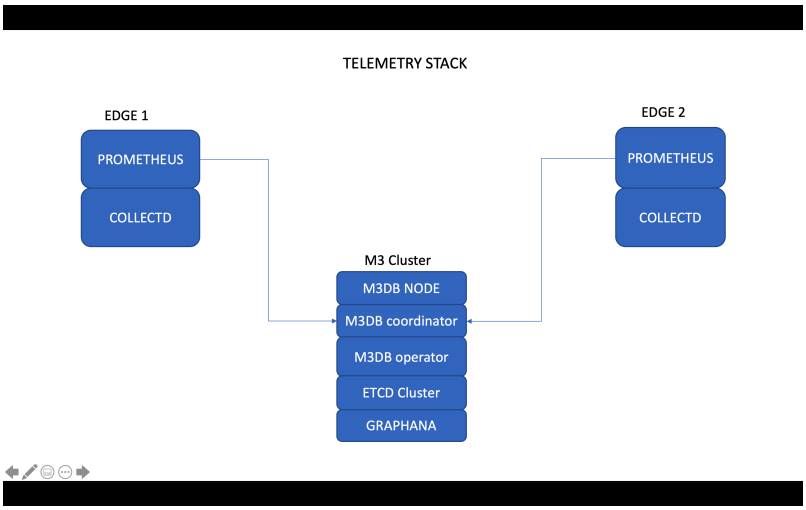
Run the script to run an end 2 end use case : https://github.com/onap/multicloud-k8s/blob/master/kud/tests/sanity-check-for-v2.sh

## Running vFw Testcase with EMCO

Run the vFw script to run an end 2 end use case testing all the current microservices: https://github.com/onap/multicloud-k8s/blob/master/kud/tests/vfw-test.sh

## Running the Telemetry stack with EMCO

Below is the multi cluster telemetry stack that we support.




Details of the installation is available at : Deploying a multi cluster distributed telemetry stack of Prometheus, M3DB and Collectd using EMCO

## Using Emco CLI (emcoctl)

Follow the Readme: https://github.com/onap/multicloud-k8s/blob/master/src/tools/emcoctl/Readme.md

Follow the example: https://github.com/onap/multicloud-k8s/tree/master/kud/tests/vnfs/comp-app/collection

**Using Emcoctl**

```
1. Create a Helm chart tar.gz. Use the example here: https://github.com/onap/multicloud-k8s/tree/master/kud
/tests/vnfs/comp-app/collection
  $ tar -czf collectd.tar.gz -C $test_folder/vnfs/comp-app/collection/app1/helm .
2. Create Profile in tar.gz format like below
  $ tar -czf collectd_profile.tar.gz -C $test_folder/vnfs/comp-app/collection/app1/profile .
3. Build emcoctl
  $ cd $MULTICLOUD-K8s_HOME/src/tools/emcoctl
  $ make
4. Update the examples/emco-cfg.yaml to  match your environment
5. Update examples/test.yaml to include the paths as above for Helm chart, Profile and location of kubeconfigs
6. Running emcoctl to apply
  $ ./emcoctl --config ./examples/emco-cfg.yaml apply -f ./examples/test.yaml
```

## Steps for creating Profile

### Adding profile per application

A per application profile contains the following:

1. **manifest.yaml**
    a. Contains the details for the profile and everything contained within
2. A **HELM** values override yaml file.
    a. It can have any name as long as it matches the corresponding entry in the **manifest.yaml**
3. Any number of files organized in a folder structure
    a. All these files should have a corresponding entry in **manifest.yaml** file

Sample Profile is described below:

Create the profile artifact
**Creating a Profile Artifact**

```
> cd vagrant/tests/vnfs/test/helm/profile
> find .
manifest.yaml
override_values.yaml
testfol
testfol/subdir
testfol/subdir/deployment.yaml

#Create profile tar.gz
> cd profile
> tar -cf profile.tar *
> gzip profile.tar
> mv profile.tar.gz ../
```

The manifest file contains the following:
**manifest.yaml**

```
---
version: v1
type:
  values: "override_values.yaml"
  configresource:
    - filepath: testfol/subdir/deployment.yaml
      chartpath: vault-consul-dev/templates/deployment.yaml
```