mojaloop

Deployment & Setup Introduction

Road to Production

By: Miguel de Barros
miguel.debarros@modusbox.com

Local Deployment & Testing Tools

Tool	Required/Optional	Description	Install Info
Docker	• Required	• Docker Engine and CLI Client • Local Kubernetes single node cluster	• https://docs.docker.com/install
Kubectl	• Required	• Kubernetes CLI for Kubernetes Management • Note Docker installs this as part of Kubernetes install	• https://kubernetes.io/docs/tasks/tools/install-kubectl • Docker Kubernetes Install (***as per this guide***) • Mac: **brew install kubernetes-cli**
Kubectx	• Optional (useful tool)	• Kubernetes CLI for Kubernetes Context Management Helper • Kubectl CLI can be used alternatively	• https://github.com/ahmetb/kubectx • Mac: **brew install kubectx**
Helm	• Required	• Helm helps you manage Kubernetes applications — Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application	• https://docs.helm.sh/using_helm/#installing-helm • Mac: **brew install kubernetes-helm**
Postman	• Required	• Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses.	• **https://www.getpostman.com/apps**

Kubernetes Concepts (1 of 2)

Deployment
A Deployment controller provides declarative management of Pods and ReplicaSets.

Pod
Smallest unit that you create or deploy. A Pod represents a running process on your cluster.

ReplicaSets
Ensures that a specified number of pod replicas are running at any one time

Service
Service is an abstraction which defines a logical set of Pods and a policy by which to access them

Ingress
An API object that manages external access to the services in a cluster, typically HTTP.

StatefulSet
A StatefulSet controller provides a declarative way to manage stateful applications, and provides guarantees about the ordering and uniqueness of these Pods

DaemonSet
A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

Ingress Controller
Ingress Controller manages the access and rules for the underlying Ingress API Objects.

Ref: https://kubernetes.io/docs/concepts/

Kubernetes Concepts (2 of 2)

ConfigMap

General application configurations that can be injected into Pods via environmental variables or file system.

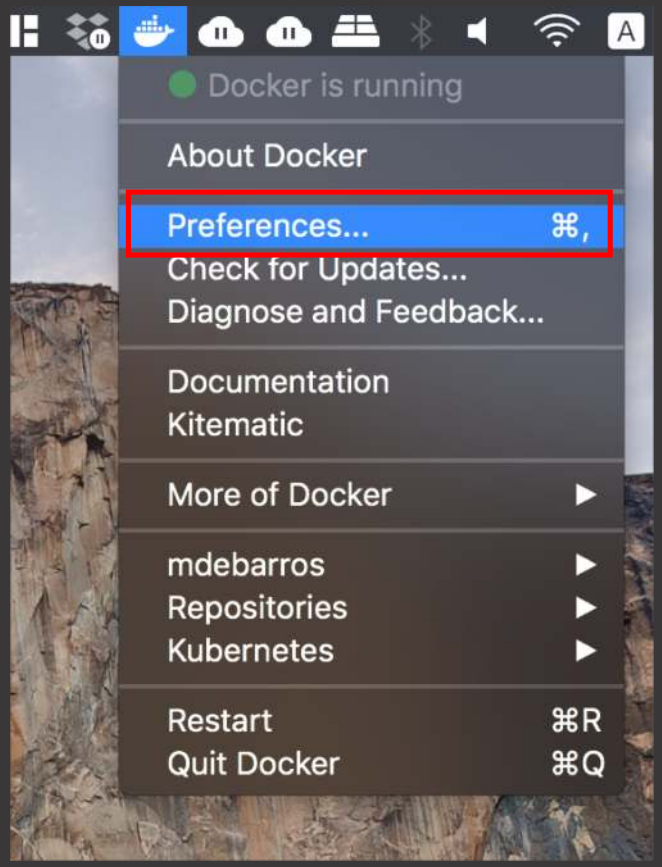
Secret

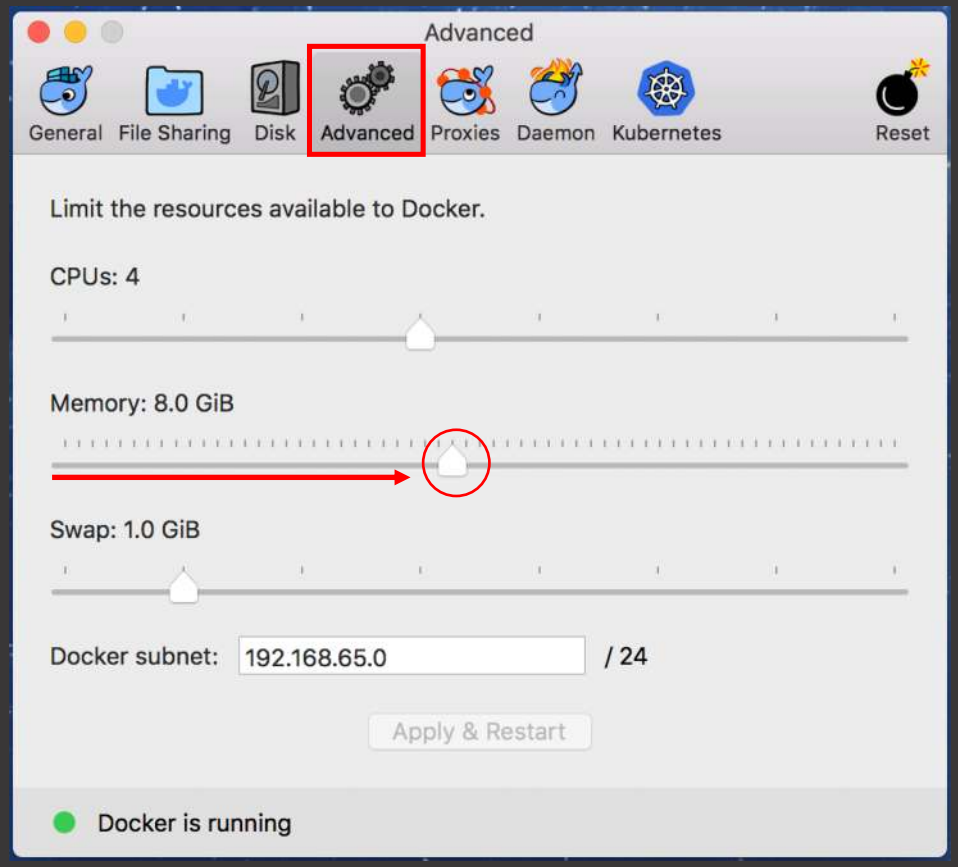
Configuration to hold sensitive information, such as passwords, OAuth tokens, and ssh keys, which can be injected into Pods via environmental variables or through a file system mount.

.

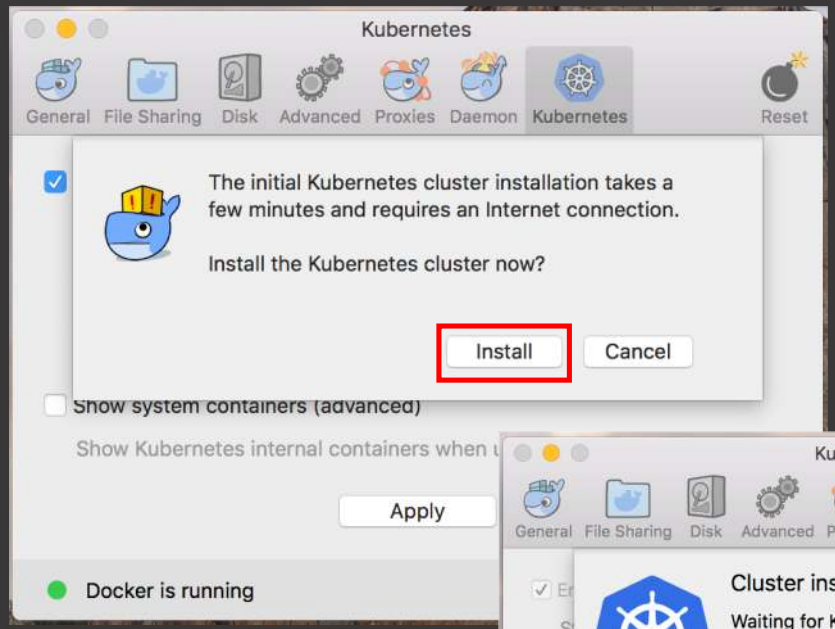
Ref: https://kubernetes.io/docs/concepts/

Kubernetes Installation with Docker - 1 of 2





Kubernetes Installation with Docker - 2 of 2



Kubernetes Environment Setup – 1 of 5

1. List your k8s Context

```
$ kubectl config get-contexts
CURRENT   NAME                  CLUSTER                    AUTHINFO          NAMESPACE
*         docker-for-desktop    docker-for-desktop-cluster docker-for-desktop
          k8s-pegasys-dev0      k8s-pegasys-dev0           user-####
          k8s-phoenix-dev0      k8s-phoenix-dev0           user-####
```

```
$ kubectx
docker-for-desktop
k8s-pegasys-dev0
k8s-phoenix-dev0
```

2. Change your Context

```
$ kubectl config use-context docker-for-desktop
Switched to context "docker-for-desktop".
```

```
$ kubectx docker-for-desktop
Switched to context "docker-for-desktop".
```

Kubernetes Environment Setup – 2 of 5

3. Install Dashboard Roles, Services & Deployment *:

```
$ kubectl create -f
https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
secret "kubernetes-dashboard-certs" created
serviceaccount "kubernetes-dashboard" created
role.rbac.authorization.k8s.io "kubernetes-dashboard-minimal" created
rolebinding.rbac.authorization.k8s.io "kubernetes-dashboard-minimal" created
deployment.apps "kubernetes-dashboard" created
service "kubernetes-dashboard" created
```

Ref: https://github.com/kubernetes/dashboard

4. Verify Kubernetes Dashboard

```
$ kubectl get pod --namespace=kube-system | grep dashboard
kubernetes-dashboard-7798c48646-2cdd7 1/1 Running 0 10h
```

5. Start proxy for local UI in a new terminal

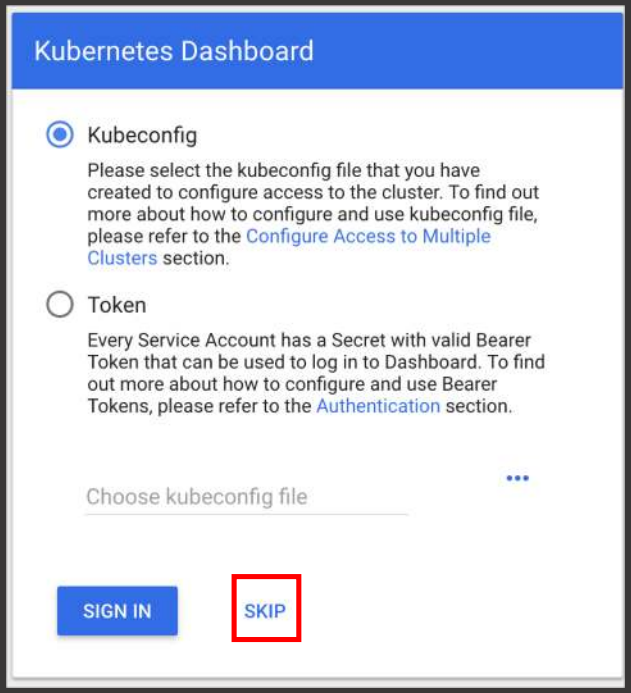
```
$ kubectl proxy ui
Starting to serve on 127.0.0.1:8001
```


*Alternative install for Dashboard using Helm: https://github.com/helm/charts/tree/master/stable/kubernetes-dashboard

Kubernetes Environment Setup – 3 of 5

6. Open the following URI in your browser

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/



Kubernetes Environment Setup – 4 of 5

7. Configure Helm CLI and install Helm Tiller on k8s cluster

```
$ helm init
$HELM_HOME has been configured at /Users/<username>/.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users'
policy.
For more information on securing your installation see:
https://docs.helm.sh/using_helm/#securing-your-helm-installation
Happy Helming!
```

Ref: https://docs.helm.sh/using_helm/#quickstart-guide

8. Validate Helm Tiller is up and running

```
$ kubectl -n kube-system get po | grep tiller
tiller-deploy-f9b8476d-85r68          1/1     Running   0          5h
```

Kubernetes Environment Setup – 5 of 5

9. Add Mojaloop repo to your Helm config (optional)

```
$ helm repo add mojaloop http://mojaloop.io/helm/repo/
```

10. Update helm repositories

```
$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Skip local chart repository
...Successfully got an update from the "mojaloop" chart repository
...Successfully got an update from the "incubator" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. ⎈ Happy Helming!⎈
```

11. Install nginx-ingress for load balancing & external access

```
$ helm --namespace kube-public install stable/nginx-ingress
NAME:   zeroed-dingo
LAST DEPLOYED: Wed Sep  5 18:29:21 2018
NAMESPACE: kube-public
STATUS: DEPLOYED
...
```

Kubernetes Environment Setup – 6 of 6

12. Add the following to your /etc/hosts

```
127.0.0.1 interop-switch.local central-kms.local forensic-logging-sidecar.local central-ledger.local central-end-user-registry.local central-directory.local central-hub.local central-settlement.local ml-api-adapter.local
```

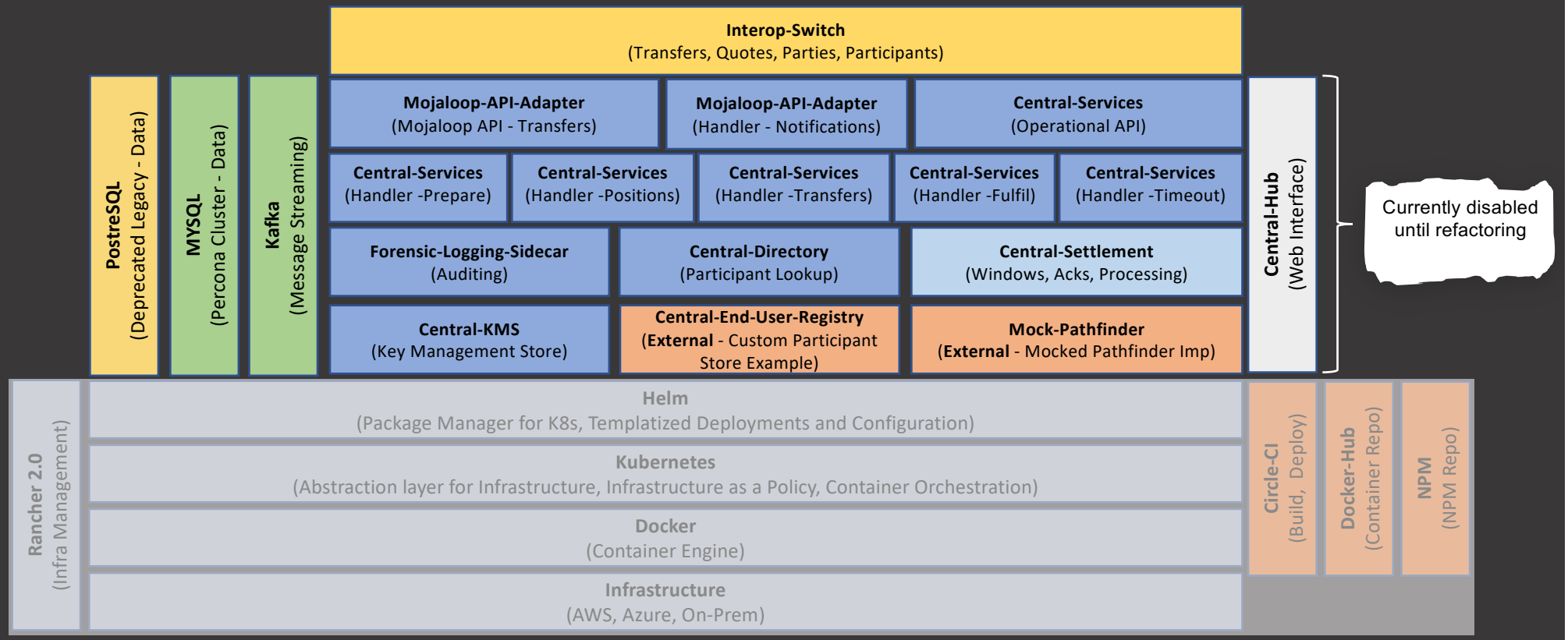
13. Test ML-API-Adapter and Central-Ledger health end-points in browser after successful installation

http://central-ledger.local/health
{"status":"OK"}

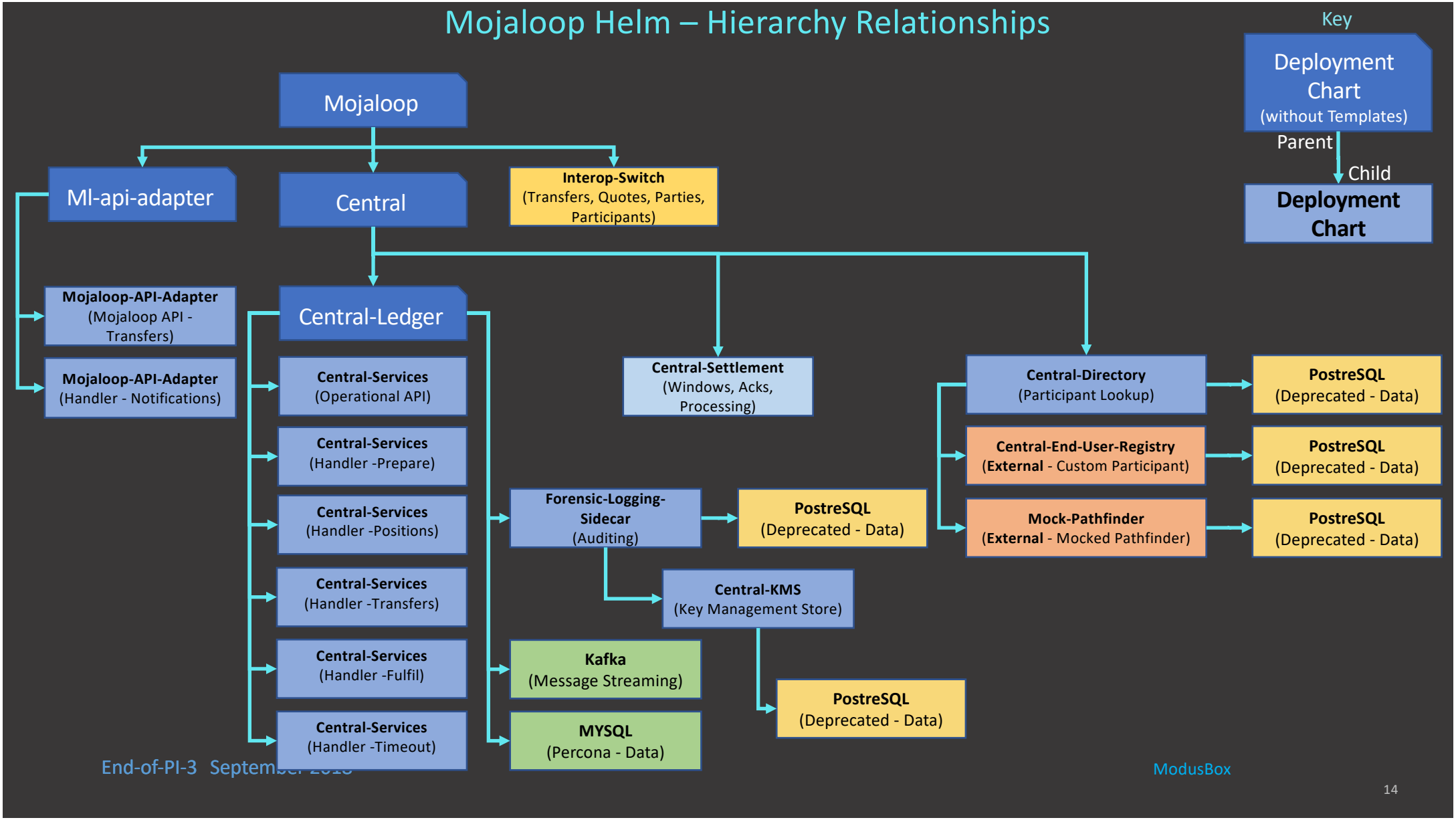
http://ml-api-adapter.local/health
{"status":"OK"}

Mojaloop Helm - Chart Overview

Helm Chart Repo: http://mojaloop.io/helm/repo ← Helm repo to be added to config
Helm Github: http://github.com/mojaloop/helm ← Location for source and values.yaml (configs)



Mojaloop Helm – Hierarchy Relationships



Mojaloop Helm - Hierarchy of Values.yaml

mojaloop/values.yaml

- global
- central
  - centralledger
    - centralledger-service
    - centralledger-handler-transfer-prepare
    - centralledger-handler-transfer-position
    - centralledger-handler-transfer-transfer
    - centralledger-handler-transfer-fulfil
    - centralledger-handler-transfer-timeout
    - forensicloggingsidecar
      - centralkms
    - mysql
  - centraldirectory
  - centralsettlements
- Interop-switch
- ml-api-adapter
  - ml-api-adapter-service
  - ml-api-adapter-handler-notification

centralledger/values.yaml

- centralledger-service
- centralledger-handler-transfer-prepare
- centralledger-handler-transfer-position
- centralledger-handler-transfer-transfer
- centralledger-handler-transfer-fulfil
- centralledger-handler-transfer-timeout
- forensicloggingsidecar
  - centralkms
- mysql

Mojaloop Helm – General CLI Commands

List Helm deployments

```
$ helm list
NAME      REVISION  UPDATED                   STATUS    CHART            NAMESPACE
pi3       1         Sun Sep  2 20:00:18 2018  DEPLOYED  mojaloop-3.5.1   demo
```

Delete the Helm deployment

```
$ helm del --purge pi3
release "pi3" deleted
```

Mojaloop Helm – Installation (1 of 2)

Mojaloop Chart installation

```
helm install --namespace=demo --name=pi3 mojaloop/mojaloop
LAST DEPLOYED: Thu Sep  6 00:34:40 2018
NAMESPACE: demo
STATUS: DEPLOYED
==> v1beta1/Deployment

NAME                                          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
pi3-centralenduserregistry-postgresql         1        1        1           0          5s
pi3-centralenduserregistry                    1        1        1           0          5s
pi3-mockpathfinder-postgresql                 1        1        1           0          5s
pi3-mockpathfinder                            1        1        1           0          5s
pi3-centraldirectory-postgresql               1        1        1           0          5s
pi3-centraldirectory                          1        1        1           0          5s
pi3-centralledger-handler-timeout             1        1        1           0          5s
pi3-centralledger-handler-transfer-fulfil     1        1        1           0          5s
pi3-centralledger-handler-transfer-position   1        1        1           0          5s
pi3-centralledger-handler-transfer-prepare    1        1        1           0          5s
pi3-centralledger-handler-transfer-transfer   1        1        1           0          5s
pi3-centralledger-service                     1        1        1           0          5s
pi3-centralkms-postgresql                     1        1        1           0          5s
pi3-centralkms                                1        1        1           0          5s
pi3-forensicloggingsidecar-ledger-postgresql  1        1        1           0          5s
pi3-forensicloggingsidecar-ledger             1        1        1           0          5s
pi3-centralsettlement                         1        1        1           0          5s
pi3-interop-switch-postgresql                 1        1        1           0          5s
pi3-interop-switch                            1        1        1           0          5s
pi3-ml-api-adapter-handler-notification       1        1        1           0          5s
pi3-ml-api-adapter-service                    1        1        1           0          5s
```

Alternative: helm install --namespace=demo --name=pi3 --repo=http://mojaloop.io/helm/repo mojaloop

Mojaloop Helm – Installation (2 of 2)

Central-Ledger Chart installation

```
helm install --namespace=demo --name=pi3 mojaloop/centralledger
LAST DEPLOYED: Thu Sep  6 00:34:40 2018
NAMESPACE: demo
STATUS: DEPLOYED
==> v1beta1/Deployment

==> v1beta1/Deployment
NAME                                          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
pi3-centralledger-handler-timeout             1        1        1           0          1s
pi3-centralledger-handler-transfer-fulfil     1        1        1           0          1s
pi3-centralledger-handler-transfer-position   1        0        0           0          1s
pi3-centralledger-handler-transfer-prepare    1        1        1           0          1s
pi3-centralledger-handler-transfer-transfer   1        1        1           0          1s
pi3-centralledger-service                     1        0        0           0          1s
pi3-centralkms-postgresql                     1        0        0           0          1s
pi3-centralkms                                1        0        0           0          1s
pi3-forensicloggingsidecar-ledger-postgresql  1        0        0           0          1s
pi3-forensicloggingsidecar-ledger             1        0        0           0          1s
```

Alternative: helm install --namespace=demo --name=pi3 --repo=http://mojaloop.io/helm/repo centralledger

Mojaloop Helm - Upgrading

Mojaloop Chart Upgrade

```
helm upgrade pi3 --set central.centralledger.centralledger-service.containers.api.image.tag=v3.5.0-snapshot mojaloop

Release "pi3" has been upgraded. Happy Helming!
LAST DEPLOYED: Thu Sep  6 02:24:20 2018
NAMESPACE: demo
STATUS: DEPLOYED
...
```

Central-Ledger Chart Upgrade

```
helm upgrade pi3 --set centralledger-service.containers.api.image.tag=v3.5.0-snapshot mojaloop/centralledger

Release "pi3" has been upgraded. Happy Helming!
LAST DEPLOYED: Thu Sep  6 02:24:20 2018
NAMESPACE: demo
STATUS: DEPLOYED
...
```

Alternative: helm upgrade pi3 --set centralledger-service.containers.api.image.tag=v3.5.0-snapshot ./centralledger

Setup Kafka CLI for Debugging

Create Cli Test Client for Kafka

```
cat <<EOF | kubectl create -f -
apiVersion: v1
kind: Pod
metadata:
  name: testclient
  namespace: demo
spec:
  containers:
  - name: kafka
    image: solsson/kafka:0.11.0.0
    command:
      - sh
      - -c
      - "exec tail -f /dev/null"
EOF
pod "testclient" created
```

Command to list Topics

```
kubectl -n demo exec -ti testclient -- ./bin/kafka-topics.sh --zookeeper pi3-zookeeper:2181 --list
__consumer_offsets
topic-dfsp1-position-abort
topic-dfsp1-position-fulfil
topic-dfsp1-position-prepare
topic-dfsp1-transfer-prepare
topic-dfsp2-position-abort
topic-dfsp2-position-fulfil
topic-dfsp2-position-prepare
topic-dfsp2-transfer-prepare
topic-notification-event
topic-transfer-fulfil
topic-transfer-transfer
```

Command to describe Processing Topics

```
kubectl -n demo exec -ti testclient -- ./bin/kafka-consumer-groups.sh --bootstrap-server pi3-kafka:9092 --group central-ledger-kafka --describe
```

Command to describe Notification Topic

```
kubectl -n demo exec -ti testclient -- ./bin/kafka-consumer-groups.sh --bootstrap-server pi3-kafka:9092 --group kafka-ml-api-adapter --describe
```

Useful Kafka CLI Commands

List logs by a label

```
kubectl -n demo logs -f <pod-id>
```

List logs by a label for the Notifications Handler

```
kubectl -n demo logs -f $(kubectl get po -l app=pi3-ml-api-adapter-handler-notification -n demo -o jsonpath='{.items[*].metadata.name}')
```

List logs by a label for the Mock-Server

```
kubectl -n mockserver logs -f $(kubectl get po -l app=mockserver -n mockserver -o jsonpath='{.items[*].metadata.name}')
```

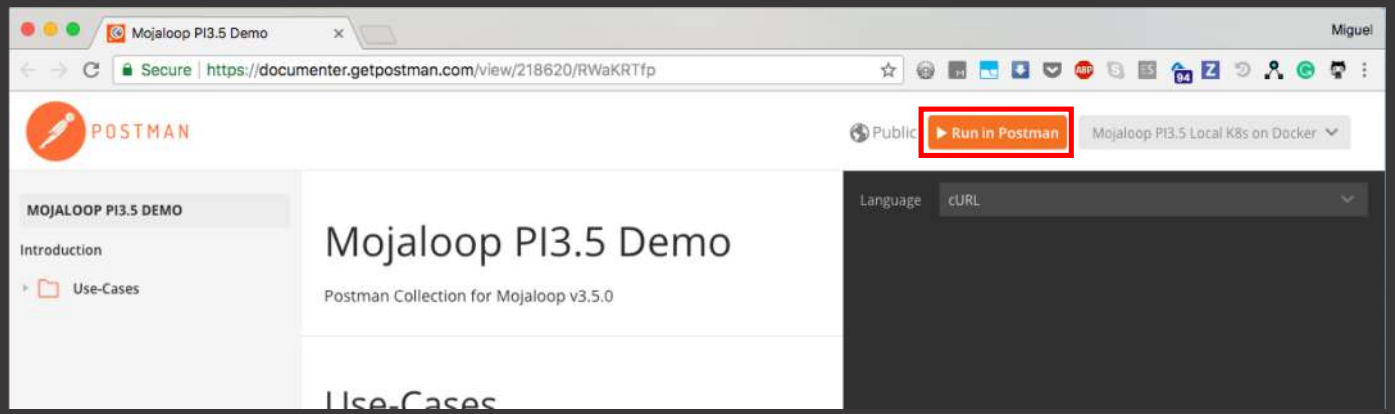
Mojaloop - Postman Collection

Postman Collection

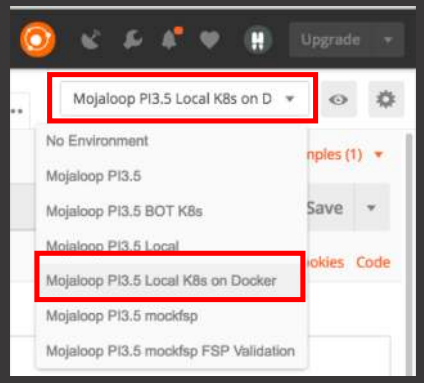
Import the following Collection:

- https://documenter.getpostman.com/view/218620/RWaKRTfp

1. Visit the URL and click the “Run in Postman” link to import the Postman Collection & Environment



2. Ensure that you select the “Mojaloop PI3.5 Local K8s on Docker” environment from the top right of your Postman



Deployment Information

Kubernetes versions

Mojaloop has been successfully deployed on the following Kubernetes version:

- v1.8.x
- v1.9.x
- v1.10.x

Kubernetes flavors for production deployments

Mojaloop should run on any flavor of Kubernetes. However we have deployed successfully on the following Kubernetes flavors:

- Rancher v2.0 (single-node and multi-node)
- Rancher v1.6 (multi-node)

Kubernetes flavours for local/dev/poc/demo deployments

Mojaloop community has used one of the following environments for local/dev/poc/demo deployments:

- Rancher v2.0 (single-node)
- Docker Kubernetes for Desktop 18.06.1-ce (single-node)
- Minikube v0.23.0+ (single-node)

Rancher v2.0 deployment environments tested

- Amazon EC2
- Azure
- Custom on-prem general purpose hardware

Rancher v2.0 deployment recommendations

- Ensure that you install Rancher Server with HA: https://rancher.com/docs/rancher/v2.x/en/installation/ha/

Deployment Recommendations

Resource Requirements:

- Control Plane (i.e. Master Node):
  https://kubernetes.io/docs/setup/cluster-large/#size-of-master-and-master-components

  3x Master Nodes for future node scaling and HA.

- ETCd Plane:
  https://coreos.com/etcd/docs/latest/op-guide/hardware.html

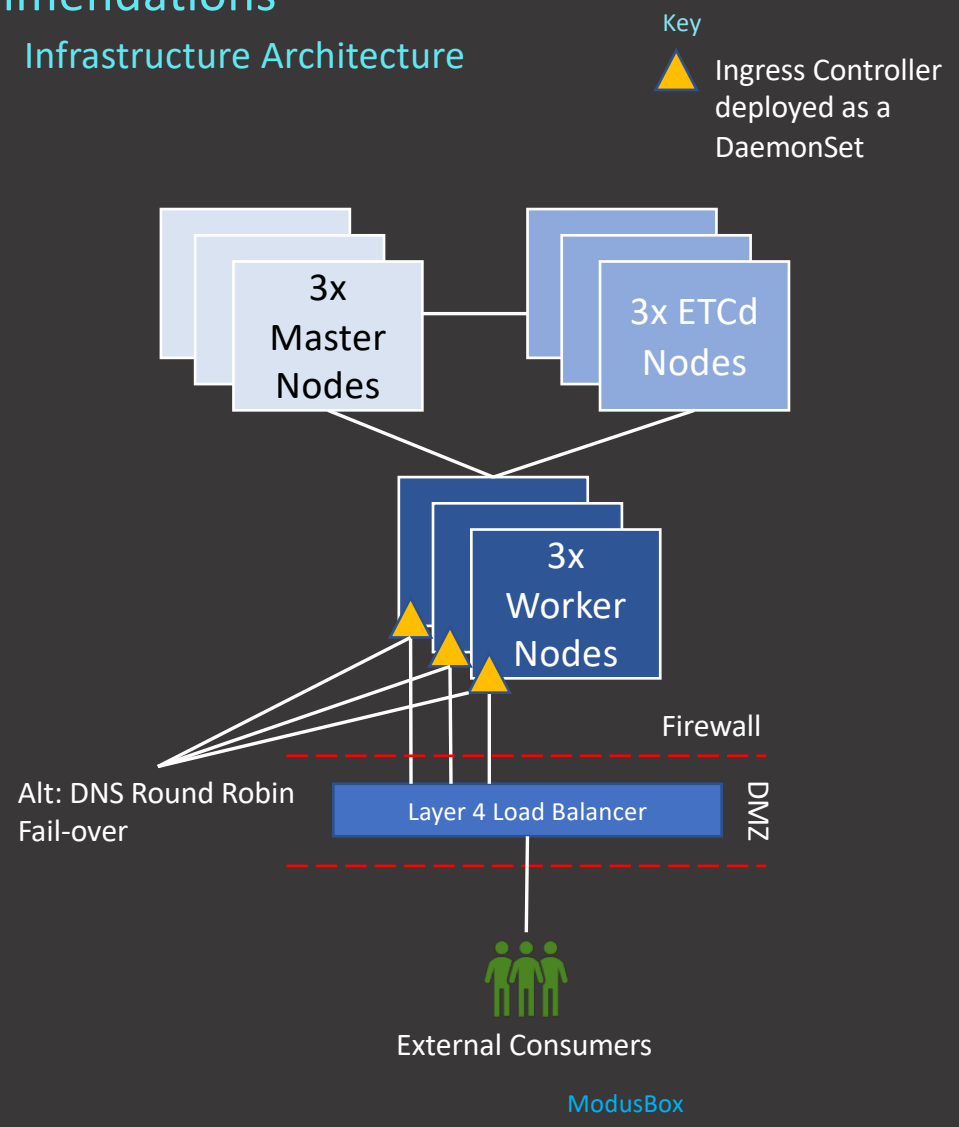
  3x ETCd nodes for HA

- Compute Plane (i.e. Worker Node):
  TBC once load testing has been concluded. However the current general *recommended size:

  3x Worker nodes, each being:
  - 4x vCPUs, 16gb of RAM, and 40gb storage

  *Note that this would also depend on your underlying infrastructure, and it does NOT include requirements for persistent volumes/storage.

Infrastructure Architecture



ModusBox

Support

Open Source Community
https://mojaloop.slack.com

Self-Invite link
https://mojaloop-slack.herokuapp.com

Contacts
Miguel de Barros (miguel.debarros@modusbox.com)

Kubernetes Training:
https://kubernetes.io/docs/tutorials/online-training/overview/

Rancher Training:
http://info.rancher.com/kubernetes-training
https://andrewlock.net/home-home-on-the-range-installing-kubernetes-using-rancher-2-0/

Rancher Installation Requirements (OS, Software, Hardware, Ports, etc):
https://rancher.com/docs/rancher/v2.x/en/installation/requirements/