Table of Contents

[Table of Contents 1](#_Toc496168948)

[1 Install dcos commands 1](#_Toc496168949)

[2 Config User 1](#_Toc496168950)

[3 Install Marathon-lb 2](#_Toc496168951)

[4 Install Private Registry 2](#_Toc496168952)

[4.1 Create TLS Certificate 2](#_Toc496168953)

[4.2 Setup Web Server Serving Certificate 3](#_Toc496168954)

[4.3 Create Certs on Nodes 3](#_Toc496168955)

[4.4 Create config.yml 3](#_Toc496168956)

[4.5 Deploy Registry 4](#_Toc496168957)

[4.6 Update marathon-lb 4](#_Toc496168958)

[4.7 Test registry 4](#_Toc496168959)

[5 Kafka 4](#_Toc496168960)

[5.1 Setup confluent-kafka 4](#_Toc496168961)

[5.2 Setup schema-registry 5](#_Toc496168962)

[5.3 Setup rest-proxy 5](#_Toc496168963)

[5.4 Setup control-center 5](#_Toc496168964)

# Install dcos commands

Copy dcos to 1st master node

Move dcos to /usr/local/bin/

dcos config set core.dcos\_url <https://10.165.105.151>

dcos config set core.ssl\_verify false

dcos package install dcos-enterprise-cli

# Config User

Superuser:

admin-sy

Readonly User:

readonly:readonly

dcos:adminrouter:ops:mesos | full (Task Tab)

dcos:adminrouter:ops:slave | full (Task Detail)

dcos:adminrouter:package | full (Universe Tab)

dcos:adminrouter:service:marathon | full (Service Tab)

dcos:adminrouter:ops:networking | full (Network Tab)

dcos:adminrouter:service:metronome | full (Jobs Tab)

dcos:adminrouter:ops:system-health | full (System Tab)

dcos:service:marathon:marathon:services:/ | read (services)

dcos:service:metronome:metronome:jobs | read

# Install Marathon-lb

Copy create-marathon-lb-accounts.sh, marathon-lb.json, and install-marathon-lb.sh to /home/mesossu/rcclapps/marathon-lb folder (create the folder if not present) of master node 1 (10.165.105.151.

Run the following command that will create the keys and secret for installation

./create-marathon-lb-accounts.sh

Run the following command that will install marathon lb

./install-marathon-lb.sh

To test the installation, browse <http://10.165.105.158:9090/haproxy?stats> that will show ha proxy page

# Install Private Registry

registry.allure.sh.rccl.com

## Create TLS Certificate

On 1st master node:

vi /home/mesossu/rcclapps/registry/gen-cert.sh

#!/bin/bash

#repo\_url length needs to be 64 or less

REPO\_URL=registry.allure.sh.rccl.com

echo "Generating key and crt"

openssl req -newkey rsa:4096 -nodes -sha256 \

-keyout domain.key -x509 -days 3650 \

-out domain.crt \

-subj "/C=US/ST=Florida/L=Miami/O=IT/CN=${REPO\_URL}"

echo "Generating pem"

cat domain.crt domain.key | tee registry.pem

chmod 700 /home/mesosadm/rcclapps/registry/gen-cert.sh

/home/mesosadm/rcclapps/registry/gen-cert.sh

## Setup Web Server Serving Certificate

On boot server:

mkdir /webserver

cd /webserver

place domain.crt and domain.key files

python -m SimpleHTTPServer 8404 &> /dev/null &

## Create Certs on Nodes

On all private and public agents:

vi copycrt.sh

#!/bin/bash

DOMAIN\_CRT=domain.crt

REPO\_URL=registry.allure.sh.rccl.com

REPO\_PORT=10104

rm -f domain.crt\* registry.pem\*

wget http://10.165.105.150:8404/domain.crt

wget http://10.165.105.150:8404/registry.pem

mkdir -p /etc/docker/certs.d/${REPO\_URL}:${REPO\_PORT}

cp ${DOMAIN\_CRT} /etc/docker/certs.d/${REPO\_URL}:${REPO\_PORT}/ca.crt

# RHEL

cp ${DOMAIN\_CRT} /etc/pki/ca-trust/source/anchors/${REPO\_URL}.crt

echo "Updating CA trust"

update-ca-trust

# This is for DCOS version 1.8 and lower only

CACERT=/opt/mesosphere/active/python-requests/lib/python3.5/site-packages/requests

echo "Hack for 1.8 cacerts"

cp ${CACERT}/{cacert.pem,cacert.pem.bup}

cat registry.pem >> ${CACERT}/cacert.pem

systemctl restart docker

chmod 700 copycrt.sh

./copycrt.sh

## Create config.yml

Login 4th private node alcmesagt934

mkdir -p /var/lib/docker/registry

cd /var/lib/docker/registry

vi config.yml

## Deploy Registry

Initial deploy through UI

Name: shared-services/registry

PIN to the Registry server

Environment Variable: "DOCKER\_OPTS": "--insecure-registry http://10.154.105.157:5000"

Volume Host Path: /var/lib/docker/registry/shared-services/registry

Optional URLs: http://10.154.105.150:8404/domain.crt, http://10.154.105.150:8404/domain.key

Optional Constraints: hostname:CLUSTER:10.154.105.157

"HAPROXY\_GROUP": "external",

"HAPROXY\_0\_SSL\_CERT": "/mnt/mesos/sandbox/registry.pem",

"HAPROXY\_0\_BACKEND\_REDIRECT\_HTTP\_TO\_HTTPS": "false",

"HAPROXY\_0\_VHOST": "10.154.105.158,10.154.105.159,registry.allure.sh.rccl.com"

Update service ports to 10104 (2 plces)

## Update marathon-lb

Update mlb-al-ext

Optional | URLs

<http://10.154.105.150:8404/registry.pem>

Restart mlb-al-ext

## Test registry

https://registry.symphony.sh.rccl.com:10104/v2/\_catalog

# Kafka

## Setup confluent-kafka

Copy **create-confluent-kafka-accounts.sh**, **install-confluent-kafka.sh**, and **confluent-kafka.json.sh** to /home/mesossu/rcclapps/confluent-kafka folder on master node 1

Run **create-confluent-kafka-accounts.sh** that will create service accounts for confluent kafka

Run **./install-confluent-kafka.sh** that will install confluent kafka

Using Dashbaord, add the following json properties

"KAFKA\_OVERRIDE\_CONFLUENT\_METRICS\_ENABLE": "true",

"KAFKA\_OVERRIDE\_KAFKA\_METRIC\_REPORTERS": "io.confluent.metrics.reporter.ConfluentMetricsReporter",

"KAFKA\_OVERRIDE\_CONFLUENT\_METRICS\_REPORTER\_MAX\_REQUEST\_SIZE": "10485760",

Using Dashbaord, update the following json properties:

"KAFKA\_OVERRIDE\_DEFAULT\_REPLICATION\_FACTOR": "3",

"KAFKA\_OVERRIDE\_COMPRESSION\_TYPE": "snappy",

"KAFKA\_OVERRIDE\_NUM\_PARTITIONS": "5",

++ set “phrase\_strategy” to ‘stage’

**Using kafka-client docker, run the following commands**:

./kafka-topics.sh --create --zookeeper 10.165.105.151:2181/dcos-service-confluent-kafka --topic dcos-connect-configs --replication-factor 3 --partitions 1 --config cleanup.policy=compact

./kafka-topics.sh --create --zookeeper 10.165.105.151:2181/dcos-service-confluent-kafka --topic dcos-connect-offsets --replication-factor 3 --partitions 50 --config cleanup.policy=compact

./kafka-topics.sh --create --zookeeper 10.165.105.151:2181/dcos-service-confluent-kafka --topic dcos-connect-status --replication-factor 3 --partitions 10 --config cleanup.policy=compact

./kafka-topics.sh --zookeeper 10.165.105.151:2181/dcos-service-confluent-kafka --topic dcos-connect-configs –describe

## Setup schema-registry

Copy **install-schema-registry.sh**, and **schema-registry.json** to /home/meosossu/rcclapps/confluent-components

++ DONOT forgot to modify in .json with new ipAddrs and Hostname:

**HAPROXY\_0\_VHOST ?**

Run ./install-schema-registry.sh

Test

<http://10.165.105.158:10131/subjects/>

<http://10.165.105.159:10131/subjects/>

## Setup rest-proxy

Copy **install-rest-proxy.sh**, and **rest-proxy.json** to /home/mesossu/rcclapps/confluent-components

++ DONOT forgot to modify in .json with new ipAddrs and Hostname:

**HAPROXY\_0\_VHOST ?**

Run ./install-rest-proxy.sh

Test:

<http://10.165.105.158/kafka-api/topics>

## Setup control-center (this one is currently removed from mesos because of multiple instance bug)

Copy **install-control-center.sh**, and **control-center.json** to /home/mesossu/rcclapps/confluent-components

++ DONOT forgot to modify in .json with new ipAddrs and Hostname:

**HAPROXY\_0\_VHOST ?**

Run ./install-control-center.sh

Test:

<http://10.165.105.158:10133>