



# **Deploying a Local Universe Containing Selected Packages on DC/OS**

## **Prerequisites:**

**Linux VM with these dependencies: (Ex. is CentOS 7.3)**

```
sudo yum -y install epel-release
sudo yum -y install python34
sudo yum -y install python34-pip
sudo pip3 install jsonschema
sudo yum -y install make
sudo yum -y install openssl
sudo pip3 install --upgrade pip
```

- Git installed
- Docker installed

## **Running DC/OS Cluster**

### **In Linux VM:**

#### **Step 1: Clone Universe Git repo**

```
git clone https://github.com/mesosphere/universe.git --branch
version-3.x
```

#### **Step 2: Build the universe-base image:**

```
cd universe/docker/local-universe/
sudo make base
```

#### **Step 3 - Build the Custom Local Universe:**

```
sudo make DCOS_VERSION=1.11.0
DCOS_PACKAGE_INCLUDE="alluxio-enterprise:2.1.1-1.7.1,arangodb3:3.2-1.
1.4,artifactory:5.1.4,artifactory-lb:5.1.4,bitbucket:4.5,calico:0.4.0
,cassandra:2.1.0-3.0.16,dcos-enterprise-cli:1.4.3,elastic:2.2.0-5.6.5
,elasticsearch:1.0.1-2,etcd:0.0.3,grafana:5.0.1-5.0.0,hdfs:2.1.0-2.6.
0-cdh5.11.0,influxdb:1.4,jenkins:3.4.0-2.89.2,kafka:2.1.0-1.0.0,kafka
-manager:1.1.0-1.3.3.16,kafka-zookeeper:2.1.0-3.4.11,kibana:2.2.0-5.6
.5,marathon:1.6.335,marathon-lb:1.11.3,mariadb:10.1.22,mongodb:3.2-0.
2,mongodb-admin:0.0.20-0.2,mongodb-replicaset:0.1.0,mysql:5.7.12-0.3,
mysql-admin:4.6.4-0.2,nginx:1.10.3,percona-mongo:0.1.0-3.4.10,portwor
x:1.1-1.2.22-24c81e4,postgresql:9.6-0.2,postgresql-admin:5.1-0.2,redi
s:3.2.9-0.0.1,spark:2.3.0-2.2.1-2,storm:0.1.0" local-universe
```

**\*Note:** This may take some time to complete, make sure there are no extra spaces or it will fail

#### **Step 4 - SCP (or manual transfer) to DC/OS Master:**

```
scp local-universe.tar.gz <user>@<master-IP>:~
scp dcos-local-universe-http.service <user>@<master-IP>:~
scp dcos-local-universe-registry.service <user>@<master-IP>:~
```

Exit

## **On Every Master Node:**

### **Step 5 - SSH into Master**

```
ssh -A <user>@<master-IP>
```

### **Step 6 - Move registry files into the /etc/systemd/system/ directory**

```
sudo mv dcos-local-universe-registry.service /etc/systemd/system/
```

```
sudo mv dcos-local-universe-http.service /etc/systemd/system/
```

### **Step 7 - Confirm that the files were successfully copied**

```
ls -la /etc/systemd/system/dcos-local-universe-*
```

### **Step 8 - Load the Universe into the local Docker instance**

```
docker load < local-universe.tar.gz
```

**\*Note:** This may take some time to complete

### **Step 9 - Restart the systemd daemon**

```
sudo systemctl daemon-reload
```

### **Step 10 - Enable the dcos-local-universe-http and dcos-local-universe-registry services:**

```
sudo systemctl enable dcos-local-universe-http
```

```
sudo systemctl enable dcos-local-universe-registry
```

### **Step 11 - Start the dcos-local-universe-http and dcos-local-universe-registry services:**

```
sudo systemctl start dcos-local-universe-http
```

```
sudo systemctl start dcos-local-universe-registry
```

### **Step 12 - Use the following commands to confirm that the services are now up and running:**

```
sudo systemctl status dcos-local-universe-http
```

```
sudo systemctl status dcos-local-universe-registry
```

```
exit
```

**Step 13 - Repeat steps 4-12 for all Master Nodes in the cluster**

## **On Every Agent Node:**

### **Step 14 - SSH into Agent Node:**

```
dcos node ssh --master-proxy --mesos-id=<mesos-id>
```

**\*Note:** In order to retrieve the node list containing <mesos-id> you can run `dcos node` in the CLI or access the Mesos GUI using [http://<master\\_url>/mesos/#](http://<master_url>/mesos/#) --> Agents

### **Step 15 - Use the following commands to download a copy of the DC/OS certificate locally and set it as trusted:**

```
sudo mkdir -p /etc/docker/certs.d/master.mesos:5000
```

```
sudo curl -o /etc/docker/certs.d/master.mesos:5000/ca.crt  
http://master.mesos:8082/certs/domain.crt
```

```
sudo systemctl restart docker
```

### **Step 16 - Configure the Apache Mesos fetcher to trust the downloaded Docker certificate:**

#### **16a. Copy the Certificate:**

```
sudo cp /etc/docker/certs.d/master.mesos:5000/ca.crt  
/var/lib/dcos/pki/tls/certs/docker-registry-ca.crt
```

#### **16b. Generate a hash:**

```
cd /var/lib/dcos/pki/tls/certs/  
openssl x509 -hash -noout -in docker-registry-ca.crt
```

#### **16c. Create a soft link:**

```
sudo ln -s /var/lib/dcos/pki/tls/certs/docker-registry-ca.crt  
/var/lib/dcos/pki/tls/certs/<hash_number>.0
```

```
exit
```

**Note:** You will need to create the `~/pki/tls/certs` directory on the public agent.

**Step 17 - Repeat steps 14-16c on each agent node in the DC/OS cluster**

## On Machine with DC/OS CLI Client:

**Step 18 - Remove default DC/OS Universe**

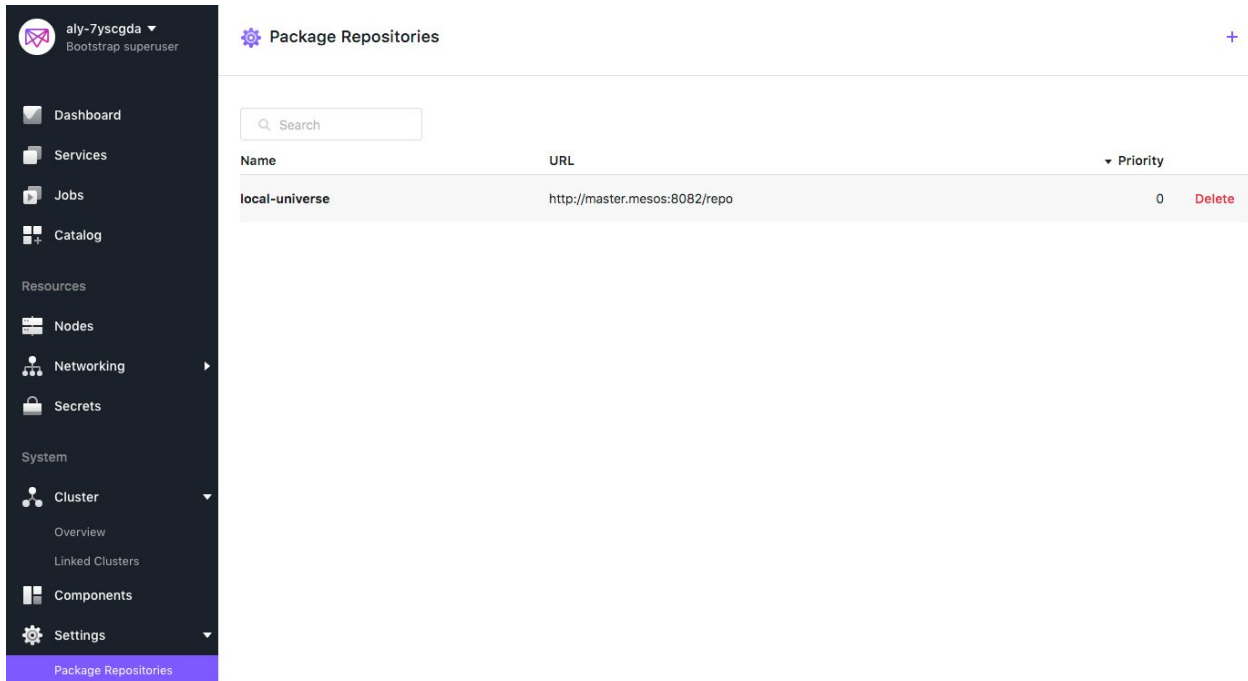
```
dcos package repo remove Universe
```

**Step 19 - Add reference to the local Universes that you added to each Master from your DC/OS CLI enabled Client:**

```
dcos package repo add local-universe http://master.mesos:8082/repo
```

**Step 20 - Validate in Local Universe in the DC/OS UI**

If successful we should see the local-universe as the only Package Repository listed



The screenshot shows the DC/OS UI interface. On the left is a dark sidebar with a navigation menu. The top of the sidebar shows the user 'aly-7yscgda' as 'Bootstrap superuser'. The menu items include Dashboard, Services, Jobs, Catalog, Resources (Nodes, Networking, Secrets), System (Cluster, Components, Settings), and Package Repositories (which is highlighted in purple). The main content area is titled 'Package Repositories' and features a search bar. Below the search bar is a table with the following data:

Name	URL	▼ Priority
local-universe	http://master.mesos:8082/repo	0 <a href="#">Delete</a>

Navigate to the Catalog tab and you should be able to only see the designated packages:

 Search catalog

## Certified

Certified packages are verified by Mesosphere for interoperability with DC/OS.



**cassandra**  
2.0.3-3.0.14

Certified



**elastic**  
2.1.0-5.6.2

Certified



**jenkins**  
3.4.0-2.89.2

Certified



**kibana**  
2.1.0-5.6.2

Certified



**marathon**  
1.5.3

Certified



**spark**  
2.1.0-2.2.0-1

Certified

## Community

Community packages are unverified and unreviewed content from the community.



**artifactory**  
5.1.4

Community



**artifactory-lb**  
5.1.4

Community



**beta-kubernetes**  
0.3.0-1.7.10-beta

Community



**bitbucket**  
4.5

Community

Congratulations, your local Universe is now complete!