

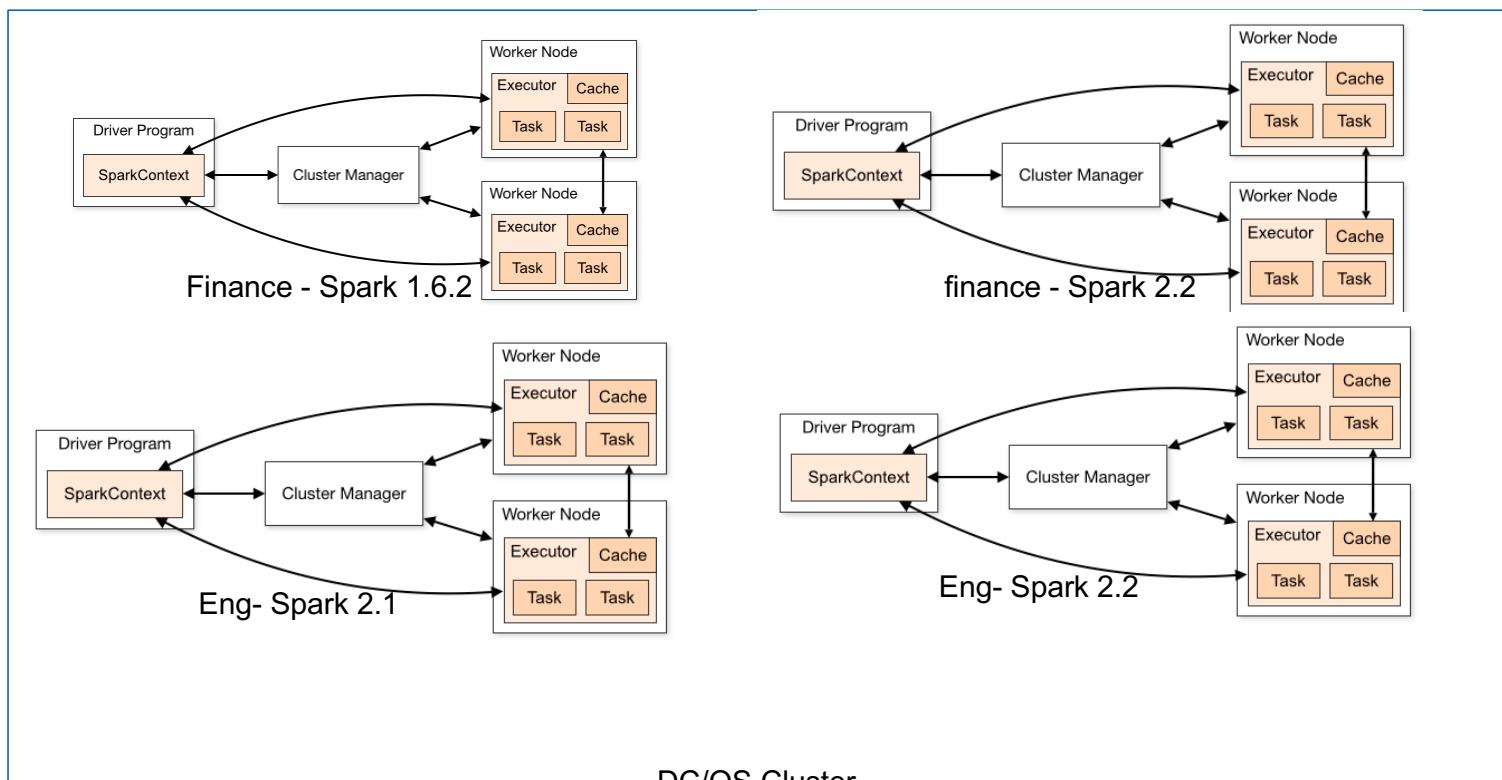
# Spark Data Processing and DC/OS

Part I: Getting started with Spark and exploring a sampling of data tools



Presented by Mark Johnson

# Spark on DC/OS Overview



- Support polygot database analytics
- In memory emphasis improves query performance
- Compatible with wide range of tools

---

# Agenda

1. Installing and configuring Spark on DC/OS
2. Running Spark Programs
3. Debugging and Monitoring
4. Spark Data Access
5. Spark Data Notebooks

---

# Part I: Spark Installation & Setup

© 2017 Mesosphere, Inc. All Rights Reserved.

# GUI Based Spark installation

The screenshot shows the Mesosphere DC/OS Catalog interface. On the left is a dark sidebar with navigation links: Dashboard, Services, Jobs, Catalog (which is highlighted in purple), Resources (Nodes, Networking, Secrets), System (Overview, Components, Settings, Organization). The main area is titled "Catalog" and has a search bar with the query "spark". Below the search bar, it says "6 services found". There are six service cards displayed in a grid:

- spark** (CERTIFIED): Icon is a white star in an orange square. Status: CERTIFIED.
- spark-history**: Icon is a purple cube. Status: COMMUNITY.
- spark-notebook**: Icon is a white star in an orange square. Status: COMMUNITY.
- spark-shuffle**: Icon is a white star in an orange square. Status: COMMUNITY.
- spark-thrift-server**: Icon is an orange square with a white star. Status: COMMUNITY.
- zeppelin**: Icon is a blue square with a white fan-like symbol. Status: COMMUNITY.



## spark

CERTIFIED

2.0.1-2.2.0-1

CONFIGURE

DEPLOY

By deploying you agree to the [terms and conditions](#)

### Description

Spark is a fast and general cluster computing system for Big Data. Documentation: <https://docs.mesosphere.com/service-docs/spark/>

### Information

SCM: <https://github.com/apache/spark.git>

Maintainer: [support@mesosphere.io](mailto:support@mesosphere.io)

### Licenses

Apache License Version 2.0: <https://raw.githubusercontent.com/apache/spark/master/LICENSE>

service

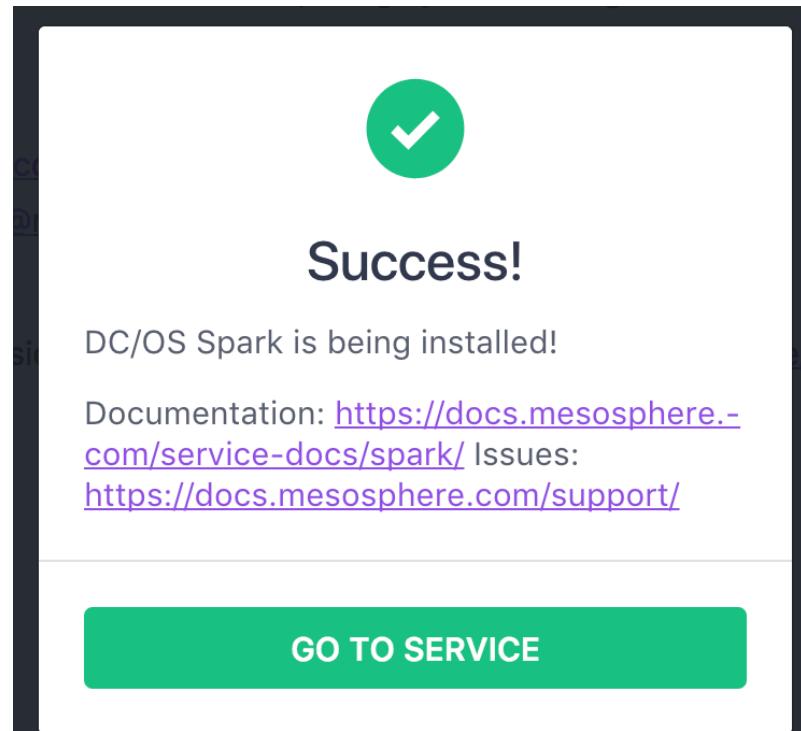
security

hdfs

## service

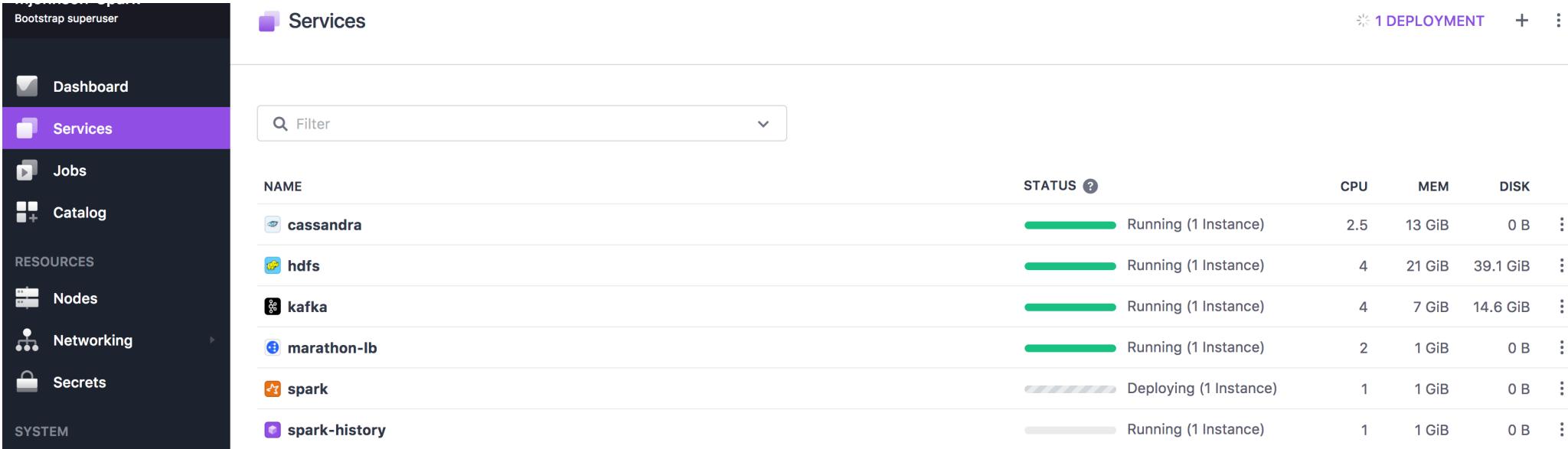
DCOS Spark configuration properties

NAME CPUS MEM ROLE SERVICE\_ACCOUNT SERVICE\_ACCOUNT\_SECRET [CANCEL](#)[REVIEW AND DEPLOY](#)



© 2017 Mesosphere, Inc. All Rights Reserved.

---



Bootstrap superuser

Services

1 DEPLOYMENT

+

...

Dashboard

Services

Jobs

Catalog

RESOURCES

Nodes

Networking

Secrets

SYSTEM

Filter

NAME	STATUS	CPU	MEM	DISK
cassandra	Running (1 Instance)	2.5	13 GiB	0 B
hdfs	Running (1 Instance)	4	21 GiB	39.1 GiB
kafka	Running (1 Instance)	4	7 GiB	14.6 GiB
marathon-lb	Running (1 Instance)	2	1 GiB	0 B
spark	Deploying (1 Instance)	1	1 GiB	0 B
spark-history	Running (1 Instance)	1	1 GiB	0 B



© 2017 Mesosphere, Inc. All Rights Reserved.

---

# Command Line Spark Installation

- `dcos package install spark --app-id=/production`
- `dcos package install spark --package-version=1.0.1-1.6.2 --app- id=/spark162`
- `dcos package install --yes spark --options=spark-dispatcher-options.json`

---

# DC/OS Spark Command Line Options

- dcos spark run ...
- dcos spark status {submission id}
- dcos spark log {submission id}
- dcos spark kill {submission id}
- dcos webui

---

# Part II:

# Running

# Spark

# Programs

---

## Run Spark from the command line

- dcos spark run --submit-args="--conf  
spark.mesos.executor.docker.image=mesosphere/spark:1.1.0-  
2.1.1-hadoop-2.6  
<https://downloads.mesosphere.com/spark/examples/pi.py> 30" --  
verbose

---

## Run Spark from the command line

- dcos spark run --submit-args="--conf  
spark.mesos.executor.docker.image=mesosphere/spark:1.1.0-  
2.1.1-hadoop-2.6  
**<https://downloads.mesosphere.com/spark/examples/pi.py>** 30" --  
verbose

# DC/OS Jobs

The screenshot shows the DC/OS Jobs interface. On the left is a sidebar with navigation links: Dashboard, Services, **Jobs** (selected), Catalog, Resources, Nodes, Networking, Secrets, System, Overview, Components, Settings, and Organization.

The main area displays a job configuration for a TeraGen task:

- ID:** tera.gen
- DESCRIPTION:** TeraGen with EventLog on HDFS
- CPUS:** 1
- MEMORY (MIB):** 1024
- DISK SPACE (MIB):** 0
- COMMAND:**

```
/opt/spark/dist/bin/spark-submit --name ${SPARK_NAME} --master ${SPARK_MASTER} --conf spark.driver.cores=${SPARK_DRIVER_CORES} --conf spark.driver.memory=${SPARK_DRIVER_MEM} --conf spark.executor.home=${SPARK_EXECUTOR_HOME} --conf spark.mesos.executor.docker.image=${SPARK_DOCKER_IMAGE} --conf spark.eventLog.enabled=${SPARK_EVENT_LOG_ENABLED} --conf spark.eventLog.dir=${SPARK_EVENT_LOG_DIR} --conf spark.mesos.uris=${SPARK_URIS} -class ${SPARK_CLASS} ${MESOS_SANDBOX}/${SPARK_JAR} ${SPARK_ARGS}
```

Below the configuration, there are sections for Docker Container and Labels.

**Docker Container**

IMAGE	mesosphere/spark:1.0.6-2.0.2-hadoop-2.6
-------	---

**Labels**

# Run every 15 minutes

Edit Job (gen) JSON MODE

General Schedule Docker Container Labels

**Schedule**  
Set time and date for the job to run

**RUN ON A SCHEDULE**

**CRON SCHEDULE \***

14 \* \* \* \*

Use cron format to set your schedule, e.g. 0 0 20 \*. [View documentation](#).

**TIME ZONE** ?

America/New\_York

**STARTING DEADLINE** ?

**ENABLED**

**CANCEL** **SAVE JOB**

# DC/OS Jobs

```
{  
  "description": "TeraGen with EventLog on HDFS",  
  "id": "tera.gen",  
  "run": {  
    "cpus": 1,  
    "mem": 1024,  
    "disk": 0,  
    "user": "root",  
    "cmd": "/opt/spark/dist/bin/spark-submit --name ${SPARK_NAME} --master ${SPARK_MASTER} --conf spark.driver.cores=${SPARK_DRIVER_CORES}  
--conf spark.driver.memory=${SPARK_DRIVER_MEM} --conf spark.executor.home=${SPARK_EXECUTOR_HOME} --conf spark.mesos.executor.docker  
.image=${SPARK_DOCKER_IMAGE} --conf spark.eventLog.enabled=${SPARK_EVENT_LOG_ENABLED} --conf spark.eventLog.dir=${SPARK_EVENT_LOG_DIR}  
--conf spark.mesos.uris=${SPARK_URIS} --class ${SPARK_CLASS} ${MESOS_SANDBOX}/${SPARK_JAR} ${SPARK_ARGS}",  
    "env": {  
      "SPARK_NAME": "TeraGen",  
      "SPARK_DOCKER_IMAGE": "mesosphere/spark:1.0.6-2.0.2-hadoop-2.6",  
      "SPARK_MASTER": "mesos://zk-1.zk:2181,zk-2.zk:2181,zk-3.zk:2181,zk-4.zk:2181,zk-5.zk:2181/mesos",  
      "SPARK_CORES_MAX": "10",  
      "SPARK_DRIVER_CORES": "1",  
      "SPARK_DRIVER_MEM": "4g",  
      "SPARK_EXECUTOR_CORES": "2",  
      "SPARK_EXECUTOR_MEM": "6g",  
      "SPARK_EXECUTOR_HOME": "/opt/spark/dist",  
      "SPARK_EVENT_LOG_ENABLED": "true",  
      "SPARK_EVENT_LOG_DIR": "hdfs://hdfs/history",  
      "SPARK_URIS": "http://api.hdfs.marathon.l4lb.thisdcos.directory/v1/endpoints/hdfs-site.xml,http://api.hdfs.marathon.l4lb.thisdcos  
.directory/v1/endpoints/core-site.xml,https://s3.amazonaws.com/vishnu-mohan/spark-terasort-2.0.2-2.11.8-jar-with-dependencies.jar",  
      "SPARK_CLASS": "com.github.ehiggs.spark.terasort.TeraGen",  
      "SPARK_JAR": "spark-terasort-2.0.2-2.11.8-jar-with-dependencies.jar",  
      "SPARK_ARGS": "10G hdfs://hdfs/terasort/terasort_in"  
    },  
    "docker": {  
      "image": "mesosphere/spark:1.0.6-2.0.2-hadoop-2.6"  
    },  
    "artifacts": [  
      {"uri": "http://api.hdfs.marathon.l4lb.thisdcos.directory/v1/endpoints/hdfs-site.xml"},  
      {"uri": "http://api.hdfs.marathon.l4lb.thisdcos.directory/v1/endpoints/core-site.xml"},  
      {"uri": "https://s3.amazonaws.com/vishnu-mohan/spark-terasort-2.0.2-2.11.8-jar-with-dependencies.jar"}  
    ]  
  },  
  "labels": {"location": "olympus"}  
  "labels": {  
    "location": "olympus",  
    "owner": "zeus"  
  }  
}
```

# Mesos Sandbox : Job Output

The screenshot shows the Apache Mesos UI interface. At the top, there's a navigation bar with tabs for Frameworks, Agents (which is selected), Roles, Offers, and Maintenance. Below the navigation bar, the URL path is /var/lib/mesos/slave/slaves/862b94df-3b0c-4182-ac6c-dfd4f26dec42-S2/frameworks/862b94df-3b0c-4182-ac6c-dfd4f26dec42-0000/executors/teragen\_20171023003140Wu73.899d4f19-b789-11e7-b9fe-428db85315b3/runs/7c86587e-f346-4f9a-80c0-1f0ad6e9b86b. The main content area displays a file listing for this executor run. The files listed are .ssl, core-site.xml, hdfs-site.xml, spark-terasort-2.0.2-2.11.8-jar-with-dependencies.jar, stderr, stderr.logrotate.conf, and stdout. Each file entry includes a 'Download' button. At the bottom of the page, there's a terminal window showing the execution of a 'teragen' task. The terminal output includes:

```
=====
Input size: 10GB
Total number of records: 100000000
Number of output partitions: 2
Number of records/output partition: 50000000
=====
Number of records written: 100000000
```

At the very bottom right, there's a copyright notice: 2017 Mesosphere, Inc. All Rights Reserved.

---

# Park III: Spark Debugging and Monitoring

© 2017 Mesosphere, Inc. All Rights Reserved.

# Mesos Task Screen

## Completed Tasks

<input type="text"/> Find...								
Framework ID	Task ID	Task Name	Role	State	Started ▼	Stopped	Host	
862b94df-3b0c-4182-ac6c-dfd4f26dec42-0005	driver-20171022161000-0001	Driver for kafka_producer_example.py	*	KILLED		5 minutes ago	10.0.1.138	Sandbox

# Mesos Sandbox

[Frameworks](#)[Agents](#)[Roles](#)[Offers](#)[Maintenance](#)[Master](#) / [Agent](#)/ [Browse](#)

/ var / lib / mesos / slave / slaves / 862b94df-3b0c-4182-ac6c-dfd4f26dec42-S7 / frameworks / 862b94df-3b0c-4182-ac6c-dfd4f26dec42-0005 / executors / driver-20171022161000-0001 / runs / d46639c6-d67a-4f7f-9be4-e370e2fcfd61

mode	nlink	uid	gid	size	mtime	
drwxr-xr-x	2	root	root	4 KB	Oct 22 12:10	.ssl
-rw-r--r--	1	root	root	517 B	Oct 22 12:10	<a href="#">kafka_producer_example.py</a>
-rw-r--r--	1	root	root	2 KB	Oct 22 12:10	<a href="#">stderr</a>
-rw-r--r--	1	root	root	0 B	Oct 22 12:10	<a href="#">stdout</a>

# Mesos Sandbox: Viewing Logs

size	mtime	
4 KB	Oct 22 12:10	 <a href="#">.ssl</a>
517 B	Oct 22 12:10	<a href="#">kafka_producer_example.py</a> <button>Download</button>
2 KB	Oct 22 12:10	<a href="#">stderr</a> <button>Download</button>
0 B	mjohnson-elastictl-1kgv9gj0yk46d-2119360242.us-west-2.elb.amazonaws.com/me...	   <pre>I1022 16:10:01.243775 6656 fetcher.cpp:533] Fetcher Info: {"cache_d I1022 16:10:01.248237 6656 fetcher.cpp:444] Fetching URI 'https://r I1022 16:10:01.248262 6656 fetcher.cpp:285] Fetching directly into I1022 16:10:01.248309 6656 fetcher.cpp:222] Fetching URI 'https://r I1022 16:10:01.248353 6656 fetcher.cpp:165] Downloading resource fr W1022 16:10:01.397732 6656 fetcher.cpp:324] Copying instead of extr I1022 16:10:01.397780 6656 fetcher.cpp:582] Fetched 'https://raw.gi</pre>

# View Sandbox on Host

```
~/demos/spark/spark_demo/kafka_spark >dcos node ssh --master-proxy --private-ip=10.0.1.138
Running `ssh -A -t core@35.165.198.240 ssh -A -t core@10.0.1.138 `
The authenticity of host '10.0.1.138 (10.0.1.138)' can't be established.
ECDSA key fingerprint is SHA256:imY1GxlD87E9Mu0aNGnF4u1kfWF1Fb74jRPLujOXlAg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.138' (ECDSA) to the list of known hosts.
Container Linux by CoreOS stable (1235.12.0)
Update Strategy: No Reboots
Failed Units: 1
  update-engine.service
core@ip-10-0-1-138 ~ $ ls -lt /var/lib/mesos/slave/slaves/862b94df-3b0c-4182-ac6c-dfd4f26dec42-S7/frameworks/862b94df-3b0c
-4182-ac6c-dfd4f26dec42-0005/executors	driver-20171022161000-0001/runs/d46639c6-d67a-4f7f-9be4-e370e2fcfd61/
total 8
-rw-r--r--. 1 root root  517 Oct 22 16:10 kafka_producer_example.py
-rw-r--r--. 1 root root 2050 Oct 22 16:10 stderr
-rw-r--r--. 1 root root     0 Oct 22 16:10 stdout
core@ip-10-0-1-138 ~ $ █
```

# SSH TUNNELS - Spark Program Debugging

- Ssh -N -L {desired port}:{DC/OS VIP or DNS}:{Endpoint Port} {user id}{dc/os url}
- Use 'dcos {framework} endpoints {desired endpoint}' to identify the VIP or DNS options.
- Change your source program to reference 'localhost:{desired port}'

```
~/dcos_scripts >dcos beta-kafka --name=kafka endpoints
[
  "broker",
  "zookeeper"
]
~/dcos_scripts >dcos beta-kafka --name=kafka endpoints broker
{
  "address": [
    "10.0.2.60:1025",
    "10.0.0.96:1025",
    "10.0.2.205:1025"
  ],
  "dns": [
    "kafka-0-broker.kafka.autoip.dcos.thisdcos.directory:1025",
    "kafka-1-broker.kafka.autoip.dcos.thisdcos.directory:1025",
    "kafka-2-broker.kafka.autoip.dcos.thisdcos.directory:1025"
  ],
  "vip": "broker.kafka.14lb.thisdcos.directory:9092"
}
```



```
~/demos/spark/spark_demo/kafka_spark >ssh -N -L 9092:broker.kafka.14lb.thisdcos.directory:9092 core@35.165.198.240
```



```
from kafka import KafkaConsumer, KafkaProducer
from kafka.errors import KafkaError

#kafka_url = "broker.kafka.14lb.thisdcos.directory:9092"
kafka_url = "localhost:9092"
topic_name = "example_topic"

def readTopic():
    consumer = KafkaConsumer(bootstrap_servers=kafka_url,
                            auto_offset_reset='earliest',
                            group_id="PrimaryGroup",
                            client_id="client-1",
                            enable_auto_commit=True, api_version=(0,10))
```

---

# Spark History

---

# Part IV: Data Access with Spark

# Spark & Cassandra

## Catalog

cassandr|

3 services found



cassandra

CERTIFIED



beta-cassandra

COMMUNITY



portworx-cassandra

COMMUNITY

All Rights Reserved.

# Cassandra DC/OS Addresses

```
~/demos/spark/spark_demo/cassandra_spark >dcos cassandra endpoints node
{
  "address": [
    "10.0.3.244:9042",
    "10.0.3.150:9042",
    "10.0.2.205:9042"
  ],
  "dns": [
    "node-0-server.cassandra.autoip.dcos.thisdcos.directory:9042",
    "node-1-server.cassandra.autoip.dcos.thisdcos.directory:9042",
    "node-2-server.cassandra.autoip.dcos.thisdcos.directory:9042"
  ],
  "vip": "node.cassandra.14lb.thisdcos.directory:9042"
}
```

# Cassandra SQL Execute Example

```
from cassandra.cluster import Cluster
from pyspark import SparkContext, SparkConf, SQLContext
from pyspark.sql import SparkSession

cassandra_url = 'node.cassandra.l4lb.thisdcos.directory:9042'
#cassandra_url = 'localhost'

spark = SparkSession.builder \
    .appName("PySpark Cassandra Read Write Example") \
    .config("spark.sql.crossJoin.enabled", "true") \
    .getOrCreate()

cluster = Cluster([cassandra_url])
session = cluster.connect()

session.execute("""CREATE KEYSPACE IF NOT EXISTS dcos_example WITH
    REPLICATION = { 'class': 'NetworkTopologyStrategy', 'datacenter1' : 3 }; """)

session.execute("USE dcos_example;")
session.execute("CREATE TABLE IF NOT EXISTS sample_table ( user_id timeuuid PRIMARY KEY,
    added_date timestamp, first_name text, last_name text, email text);")
```

# Spark tasks and driver

## Active Tasks

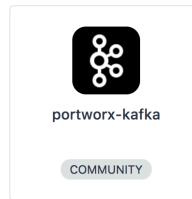
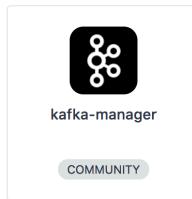
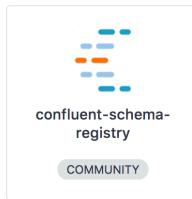
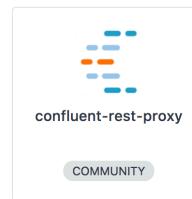
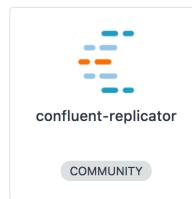
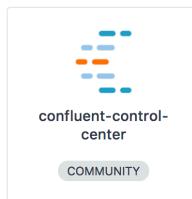
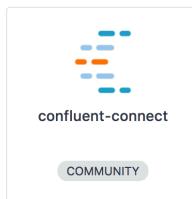
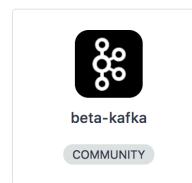
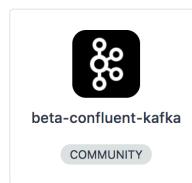
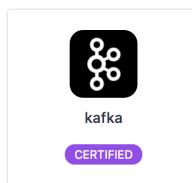
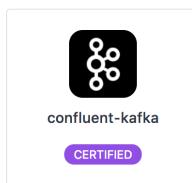
Framework ID	Task ID	Task Name	Role	State	Started ▼	Host	
862b94df-3b0c-4182-ac6c-dfd4f26dec42-0005-driver-20171023032008-0028	2	Task 2	*	STAGING		10.0.1.103	Sandbox
862b94df-3b0c-4182-ac6c-dfd4f26dec42-0005-driver-20171023032008-0028	1	Task 1	*	STAGING		10.0.1.138	Sandbox
862b94df-3b0c-4182-ac6c-dfd4f26dec42-0005	driver-20171023032008-0028	Driver for Cassandra_rw.py	*	RUNNING	just now	10.0.2.60	Sandbox

# Spark & Kafka

Catalog

kafka ×

11 services found



© 2017 Mesosphere, Inc. All Rights Reserved.

# Identify Kafka Broker Endpoints

```
~/demos/spark/spark_demo/kafka_spark >dcos beta-kafka --name=kafka endpoints broker
{
  "address": [
    "10.0.2.60:1025",
    "10.0.0.96:1025",
    "10.0.2.205:1025"
  ],
  "dns": [
    "kafka-0-broker.kafka.autoip.dcos.thisdcos.directory:1025",
    "kafka-1-broker.kafka.autoip.dcos.thisdcos.directory:1025",
    "kafka-2-broker.kafka.autoip.dcos.thisdcos.directory:1025"
  ],
  "vip": "broker.kafka.l4lb.thisdcos.directory:9092"
}

~/demos/spark/spark_demo/kafka_spark >
```

# Kafka Producer

```
from kafka import KafkaProducer
import time

kafka_url = ["kafka-0-
broker.kafka.autoip.dcos.thisdcos.directory:1025","kafka-1-
broker.kafka.autoip.dcos.thisdcos.directory:1025", "kafka-2-
broker.kafka.autoip.dcos.thisdcos.directory:1025"]
topic_name = "example_topic"

p = KafkaProducer(bootstrap_servers=kafka_url,api_version=(0,10))

iter_count = 0

while True:
    msg = b"test message - {}".format(iter_count)
    print msg
    p.send(topic_name, msg)
    time.sleep(1)
    iter_count = iter_count + 1
```

---

# Kafka Consumer

```
om kafka import KafkaConsumer

print("Start of Kafka consumer program")
kafka_url = "broker.kafka.l4lb.thisdcos.directory:9092"
#kafka_url = "localhost:9092"

topic_name = "example_topic"

print("Connecting to {}".format(kafka_url))
consumer = KafkaConsumer(topic_name, bootstrap_servers=kafka_url,
                         group_id="PrimaryGroup",
                         enable_auto_commit=True,
                         api_version=(0,10))
print ('Start consuming')
for message in consumer:
    print(message)
print("Done consuming")
```

# PostgresQL

mjohnson-Spark ▾

Bootstrap superuser

 Dashboard

 Services

 Jobs

 Catalog

RESOURCES

 Nodes

 Networking

 Secrets

SYSTEM

 Catalog

 postgresql 

3 services found



postgresql

COMMUNITY



postgresql-admin

COMMUNITY



tadpoledbhub

COMMUNITY



## postgresql

COMMUNITY 9.6-0.2

CONFIGURE

DEPLOY

By deploying you agree to the [terms and conditions](#)

### Description

PostgreSQL is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance.

### Pre-Install Notes

This DC/OS Service is currently in preview. There may be bugs, incomplete features, incorrect documentation, or other discrepancies.

[Advanced Installation options notes](#)

**storage / persistence:** create local persistent volumes for internal storage files to survive across restarts or failures.

**storage / persistence / external:** create external persistent volumes. This allows to use an external storage system such as Amazon EBS, OpenStack Cinder, EMC Isilon, EMC ScaleIO, EMC XtremIO, EMC VMAX and Google Compute Engine persistent storage. **NOTE:** To use external volumes with DC/OS, you MUST enable them during CLI or Advanced installation.

**storage / host\_volume:** if persistence is not selected, this package can use a local volume in the host for storage, like a local directory or an NFS mount. The parameter `host_volume` controls the path in the host in which these volumes will be created, which MUST be the same on all nodes of the cluster.

**NOTE:** If you didn't select persistence in the storage section, or provided a valid value for `host_volume` on installation, YOUR DATA WILL NOT BE SAVED IN ANY WAY.

**networking / port:** This DC/OS service can be accessed from any other application through a NAMED VIP in the format `service_name.marathon.l4lb.thisdcos.directory:port`. Check status of the VIP in the *Network* tab of the DC/OS Dashboard (Enterprise DC/OS only).

**networking / external\_access:** create an entry in Marathon-LB for accessing the service from outside of the cluster

# PostgresQL - Storage

The screenshot shows a configuration interface for a PostgreSQL service. On the left, a sidebar lists service, postgresql, database, storage (which is selected and highlighted in purple), and networking. The main panel has a header "storage" and a subtitle "PostgreSQL storage configuration properties". It contains two sections: "HOST\_VOLUME" with a value of "/tmp" and "PGDATA" with a value of "pgdata". Below these is a "persistence" section with a checkbox labeled "ENABLE" which is checked. A "VOLUME\_SIZE" input field shows "512". At the bottom are "CANCEL" and "REVIEW AND DEPLOY" buttons.

postgresql  
9.6-0.2

service  
postgresql  
database  
**storage**  
networking

**storage**  
PostgreSQL storage configuration properties

**HOST\_VOLUME** ?  
/tmp

**PGDATA** ?  
pgdata

**persistence**  
Enable persistent storage.  
 **ENABLE** ?

**VOLUME\_SIZE** ?  
512

**CANCEL** **REVIEW AND DEPLOY**

# PostgresQL – Networking – EXTERNAL ACCESS



service  
postgresql  
database  
storage  
networking

## networking

PostgreSQL networking configuration properties

### PORT ?

5432

### HOST\_MODE ?

## external\_access

Enable access from outside the cluster through Marathon-LB. NOTE: this connection is unencrypted.

### ENABLE ?

### EXTERNAL\_ACCESS\_PORT ?

15432

CANCEL

REVIEW AND DEPLOY

© 2017 Mesosphere, Inc. All Rights Reserved.

# External access PostgresOL

HAProxy version 1.7.6, released 2017/06/16

## Statistics Report for pid 27552

### > General process information

```
pid = 27552 (process #1, nbproc = 1)
uptime = 0d 0h0m00s
system limits: memmax = unlimited; ulimit-n = 100036
maxsock = 100036; maxconn = 50000; maxpipes = 0
current connns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/10; idle = 100 %
```

active UP  
 active UP, going down  
 active DOWN, going up  
 active or backup DOWN  
 active or backup DOWN for maintenance (MAINT)  
 active or backup SOFT STOPPED for maintenance  
 Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

- Display option:
- Scope :
  - [Hide 'DOWN' servers](#)
  - [Refresh now](#)
  - [CSV export](#)
- External resources:
- [Primary site](#)
  - [Updates \(v1.7\)](#)
  - [Online manual](#)

stats																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
Frontend			-	1	1	-	1	1	10 000	1	0	0	0	0	0	0	0	0	0	0	OPEN									
Backend	0	0		0	0		0	0	1 000	0	0	0s	0	0	0	0	0	0	0	0	0s UP					0	0	0	0	
marathon_http_in																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
Frontend			-	0	0		0	0	10 000	0		0	0	0	0	0	0	0			OPEN									
marathon_http_appid_in																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
Frontend			-	0	0		0	0	10 000	0		0	0	0	0	0	0	0			OPEN									
marathon_https_in																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
Frontend			-	0	0		0	0	10 000	0		0	0	0	0	0	0	0			OPEN									
postgresql_15432																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
Frontend			-	0	0		0	0	10 000	0		0	0	0	0	0	0	0			OPEN									
postgresql_15432																			Server											
	Queue			Session rate			Sessions						Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis									
10_0_0_96_13831	0	0	-	0	0		0	0	0	0	0	?	0	0	0	0	0	0	0	0	0s UP	L4OK in 0ms	1	Y	-	0	0	0s	-	
Backend	0	0		0	0		0	0	1 000	0	0	?	0	0	0	0	0	0	0	0	0s UP		1	1	0		0	0	0s	

# PostgresQL Example Program

```
import psycopg2
import sys
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

con = None
host = "34.215.42.192"
pgsql_port = "15432"

try:
    con = psycopg2.connect("host='{}' dbname='postgres' user='admin' password='password' port={}".format(host,pgsql_port))

    con.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT)
    cur = con.cursor()
    cur.execute("CREATE DATABASE test_db;")
    cur.execute("CREATE TABLE link (ID serial PRIMARY KEY, url VARCHAR (255) NOT NULL, name VARCHAR (255) NOT
NULL,"
               + "description VARCHAR (255),rel VARCHAR(50); ")
    cur.execute("INSERT INTO link (url, name) VALUES('http://www.postgresqltutorial.com','PostgreSQL Tutorial');")

    con.commit()
except psycopg2.DatabaseError, e:
    if con:
        con.rollback()

    print 'Error %s' % e
    sys.exit(1)

finally:
    if con:
        con.close()
```

```
/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7 /Users/markjohnson/demos/spark/spark_demo/postgresQL_Spark/postgresql_rw.py
```

```
Show me the Rows:
```

```
http://www.postgresqltutorial.com, PostgreSQL Tutorial
```

```
Process finished with exit code 0
```

---

# Spark & HDFS

# Spark & Elastic Search

The screenshot shows the Mesosphere Catalog interface. On the left is a dark sidebar with a purple header bar containing the user name "mjohnson-Spark" and the role "Bootstrap superuser". Below this are sections for Dashboard, Services, Jobs, and Catalog (which is highlighted with a purple background). Under Resources, there are links for Nodes, Networking, Secrets, and System Overview. Under Components, there are links for Components, Settings, and Organization.

The main area is titled "Catalog" and features a search bar with the query "kibana" and a dropdown suggestion "cabana". Below the search bar, it says "5 services found". There are five service cards displayed:

- elastic**: Certified
- kibana**: Certified
- beta-elastic**: Community
- beta-kibana**: Community
- portworx-elastic**: Community

© 2017 Mesosphere, Inc. All Rights Reserved.

# ElasticSearch identify coordinator

```
~/demos/spark/spark_demo/elasticsearch >dcos elastic endpoints
[
  "coordinator-http",
  "coordinator-transport",
  "data-http",
  "data-transport",
  "master-http",
  "master-transport"
]
~/demos/spark/spark_demo/elasticsearch >
```

```
~/demos/spark/spark_demo/elasticsearch >
~/demos/spark/spark_demo/elasticsearch >dcos elastic endpoints coordinator-http
{
  "address": ["10.0.1.138:1025"],
  "dns": ["coordinator-0-node.elastic.autoip.dcos.thisdcos.directory:1025"],
  "vip": "coordinator.elastic.14lb.thisdcos.directory:9200"
}
~/demos/spark/spark_demo/elasticsearch >
```

# ElasticSearch post example

```
import elasticsearch

elastic_url = 'coordinator-0-node.elastic.autoip.dcos.thisdcos.directory'
elastic_url = 'localhost'
elastic_port = 1025

print('Connecting to Elastic Search url={},'
port='}.format(elastic_url,elastic_port))
es = elasticsearch.Elasticsearch([{'host':elastic_url,'port':elastic_port}])
es.index(index='posts', doc_type='blog', id=1, body={
    'author': 'Santa Clause',
    'blog': 'Slave Based Shippers of the North',
    'title': 'Using Celery for distributing gift dispatch',
    'topics': ['slave labor', 'elves', 'python',
              'celery', 'antigravity reindeer'],
    'awesomeness': 0.2
})
```

- Use 'dcos elastic'
- Example writes to the index posts one entry

# Kibana Search Example

The screenshot shows the Kibana interface with a search results page. The left sidebar has icons for Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management, with 'Discover' selected. The main area shows a search bar with '1 hit' and the query 'Search... (e.g. status:200 AND extension:PHP)'. To the right is a 'New' button, 'Save', 'Open', and 'Share' buttons, along with a note 'Uses lucene query syntax' and a magnifying glass icon. The search results panel shows a single hit under the '\_source' field:

```
blog: Slave Based Shippers of the North awesomeness: 0.2 topics: slave labor, elves, python, celery, antigravity reindeer title: Using Celery for distributing gift dispatch author: Santa Clause _id: 1 _type: blog _index: posts _score: 1
```

---

# Part V: Spark Notebooks

© 2017 Mesosphere, Inc. All Rights Reserved.

# Zeppelin Notebooks

mjohnson-Spark ▾

Bootstrap superuser

Dashboard

Services

Jobs

Catalog

RESOURCES

Nodes

Networking

Secrets

SYSTEM

Catalog

zeppelin|



2 services found



spark-shuffle

COMMUNITY



zeppelin

COMMUNITY

sphere, Inc. All Rights Reserved.



zeppelin  
0.5.6-2

service

spark

## spark

URI ?

EXECUTOR\_DOCKER\_IMAGE ?

CORES\_MAX ?

EXECUTOR\_MEMORY ?

CANCEL

REVIEW AND DEPLOY

© 2017 Mesosphere, Inc. All Rights Reserved.



Zeppelin

Notebook ▾

Interpreter

Search in your notebooks



Connected

Note K1XEG2WM9



```
val rdd = sc.parallelize(1 to 5)
rdd.sum()

rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:29
res0: Double = 15.0
Took 98 seconds
```

FINISHED ▶ ✎ 📄 ⏪



MESOSPHERE

© 2017 Mesosphere, Inc. All Rights Reserved.



## Zeppelin Tutorial



```
// Zeppelin creates and injects sc (SparkContext) and sqlContext (HiveContext or SQLContext)
// So you don't need create them manually
```

```
// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("https://s3.amazonaws.com/apache-zeppelin/tutorial/bank/bank.csv"),
    Charset.forName("utf8")).split("\n"))

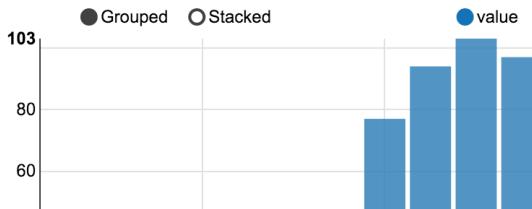
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)

val bank = bankText.map(s => s.split(",")).filter(s => s(0) != "\"age\"").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\"", ""),
    s(2).replaceAll("\"", ""),
    s(3).replaceAll("\"", ""),
    s(5).replaceAll("\"", "").toInt
  )
).toDF()
bank.registerTempTable("bank")

import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
bankText: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:37
defined class Bank
bank: org.apache.spark.sql.DataFrame = [age: int, job: string, marital: string, education: string, balance: int]
Took 15 seconds (outdated)
```



settings ▾ FINISHED ▶ ↻ ↺ ↻ ↻ ↻



maxAge settings ▾ FINISHED ▶ ↻ ↺ ↻ ↻ ↻



marital settings ▾ FINISHED ▶ ↻ ↺ ↻ ↻ ↻



---

# Linking Data sources

---

# Additional Resources

- Presentation Examples: [https://github.com/markfjohnson/dcos\\_spark\\_demo.git](https://github.com/markfjohnson/dcos_spark_demo.git)
- DC/OS Certified Services Documentation: <https://docs.mesosphere.com/service-docs/>
- Zeppelin Examples Doc: <https://github.com/dcos/examples/tree/master/zeppelin/1.9>
- Misc Examples: [www.dcosexamples.com](http://www.dcosexamples.com)
- Mesosphere Blog: <https://mesosphere.com/blog/>
- Developers Guide to the Universe: <https://mesosphere.com/blog/a-developers-guide-to-the-universe/>

# Spark and DC/OS

Next Session: Spark Security & HDFS Programming



Presented by Mark Johnson

# MESOSPHERE DC/OS ENTERPRISE FEATURES

New for 1.10

## Mesosphere DC/OS

Open source platform for modern apps

## Mesosphere DC/OS Enterprise

Production-grade solution for mission-critical apps

Features	Mesosphere DC/OS	Mesosphere DC/OS Enterprise
Platform Services	Kubernetes (Beta) container orchestration Marathon container orchestration with pods support <b>Production-certified data services (e.g., Spark, Kafka, Cassandra)</b>	✓ ✓ ✓
DC/OS Catalog	Distributed fault tolerant jobs scheduler (Cron) Universe of 100+ platform services from easily deployed with a single single-click or CLI command; Developed by Mesosphere, DC/OS community, and commercial partners	✓ ✓
Resource Management	Apache Mesos distributed systems kernel w/ Universal Container Runtime Management of persistent and external volumes Virtual networks with IP per container and Container Network Interface (CNI) Support Distributed load balancer, service discovery and name-based VIPs GPU-based scheduling	✓ ✓ ✓ ✓ ✓
Management & Monitoring	Guided on-premise and cloud installation templates (AWS, Azure) Powerful CLI and GUI Non-disruptive DC/OS upgrades	✓ ✓ ✓
Ops & troubleshoot	Platform monitoring & troubleshooting tools Application-level logging, metrics & debugging <b>In-Place Upgrade for Data services</b> <b>Validated DC/OS upgrades with automated pre and post upgrade health checks</b>	✓ ✓ ✓
Security	Role-Based Access Control for containers, jobs, and data services Identity management integration (Active Directory/LDAP/SAML 2.0/OpenID Connect) Secrets Management (Key/Value <b>and File-based</b> ) Public key infrastructure <b>w/ Custom certificate authority integration</b> Security audit logging	✓ ✓ ✓ ✓ ✓
Multi-tenancy	Fine-grained access control lists for containers and services <b>w/ folder integration</b> Service accounts	✓ ✓
Adv. Network	<b>High Performance L4/L7 Ingress load balancer (Edge-LB)</b>	✓
Price	Emergency patching Support for open-source Mesosphere-developed frameworks*	Cassandra, Kafka, Spark, Jenkins, Elastic, HDFS Cassandra, Kafka, Spark, Jenkins, Elastic, HDFS
Support	Support options	Premium Standard, Premium
	List price	Premium: \$3,000 Standard: \$3,250 Premium: \$3,950

\* Does not include support for baseline technology, e.g. Apache Spark