MESOSPHERE

A customer - Cassandra Operations

Updating Cassandra configuration to move C nodes between DC/OS Agents for maintenance*

Arthur Johnson,
Solutions Architect, Mesosphere, Inc.
ajohnson@mesosphere.io

Summary

A customer has a need to affect the configuration of Cassandra in order to move C* nodes to other DC/OS agents so that the DC/OS agents can be brought down and removed from the cluster for maintenance. This can be done by utilizing “Constraints” in the Cassandra configuration and “pod replace”. By doing a “pod replace” Cassandra will move the data pod to a different DC/OS agent. Pod replace should be used when a DC/OS agent is being removed, is permanently down, or pod placement constraints need to be updated. Once the DC/OS agent is no longer running C* nodes/pods, the agents can be safely removed from the cluster. Care must be taken in creating the constraints, ensuring that services are moved, and that all data within the Cassandra instance has been recovered to different nodes/DC/OS agents. Depending on the environment, this may take anywhere from an hour to several hours, allowing for all data to be synchronized to new agents.

Environment

DC/OS 1.11.4
5 Masters
8 Private agents
2 Public agents
Software/Framework: Cassandra 2.3.0-3.0.16

Pre-Reqs

1. DCOS-CLI
    a. Dcos sub package “Cassandra CLI”
        i. Dcos package install cassandra --cli
2. Access to DCOS-UI
    a. Ability to modify plan and container configuration for Cassandra nodes

Activity

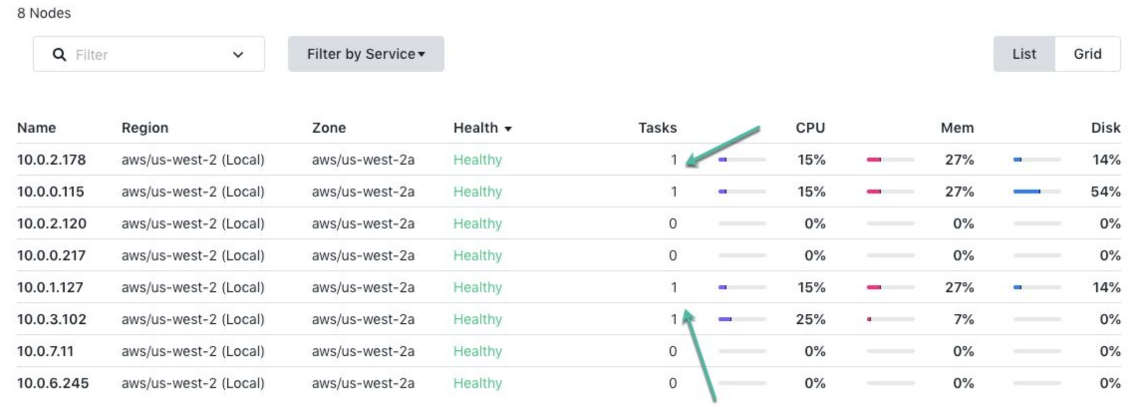
Reference documentation: Cassandra 2.3.0-3.0.16 →
https://docs.mesosphere.com/services/cassandra/2.3.0-3.0.16/operations/

Moving Cassandra Nodes to different DC/OS Agents

1 Using dcos-cli, view the current plan
---> `dcos cassandra --name=cassandra plan list`

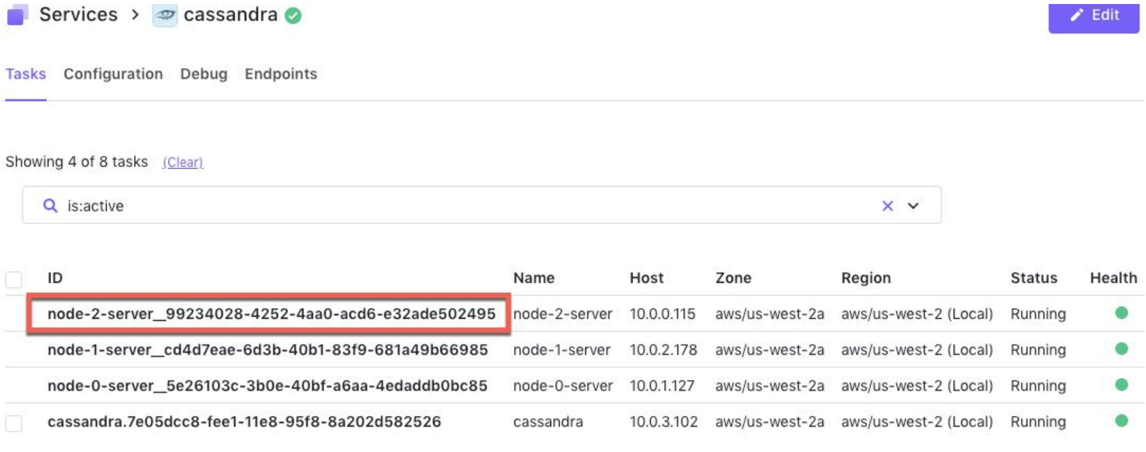
Make sure there is a “recovery” plan (as default, there should be)

2 Using the dcos-UI, review the agents in the cluster, take note which agents are *running* Cassandra nodes, and *which agents* are *not* running Cassandra.



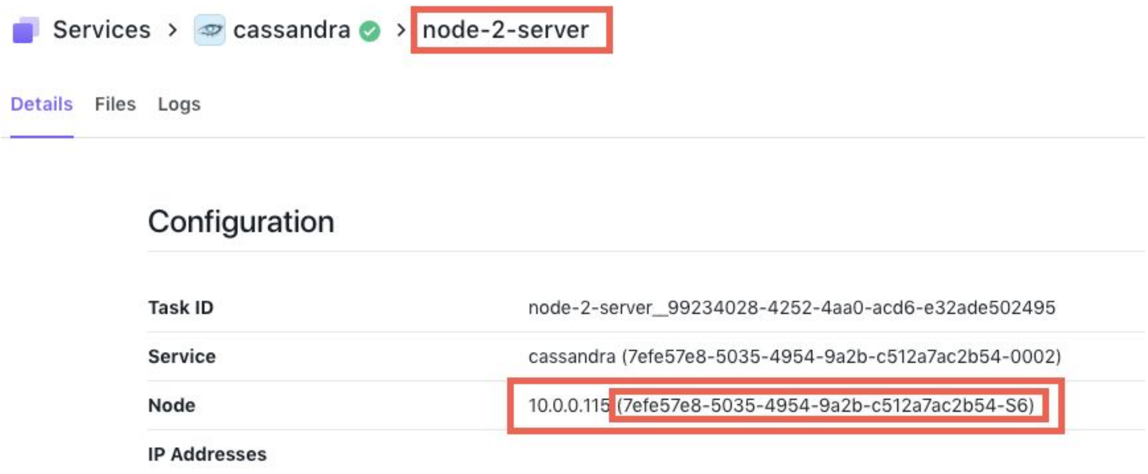
Name	Region	Zone	Health	Tasks	CPU	Mem	Disk
10.0.2.178	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	14%
10.0.0.115	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	54%
10.0.2.120	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.0.217	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.1.127	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	14%
10.0.3.102	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	25%	7%	0%
10.0.7.11	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.6.245	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%

2 Using dcos-UI, review the service, and then review each node, click on the link for a node:



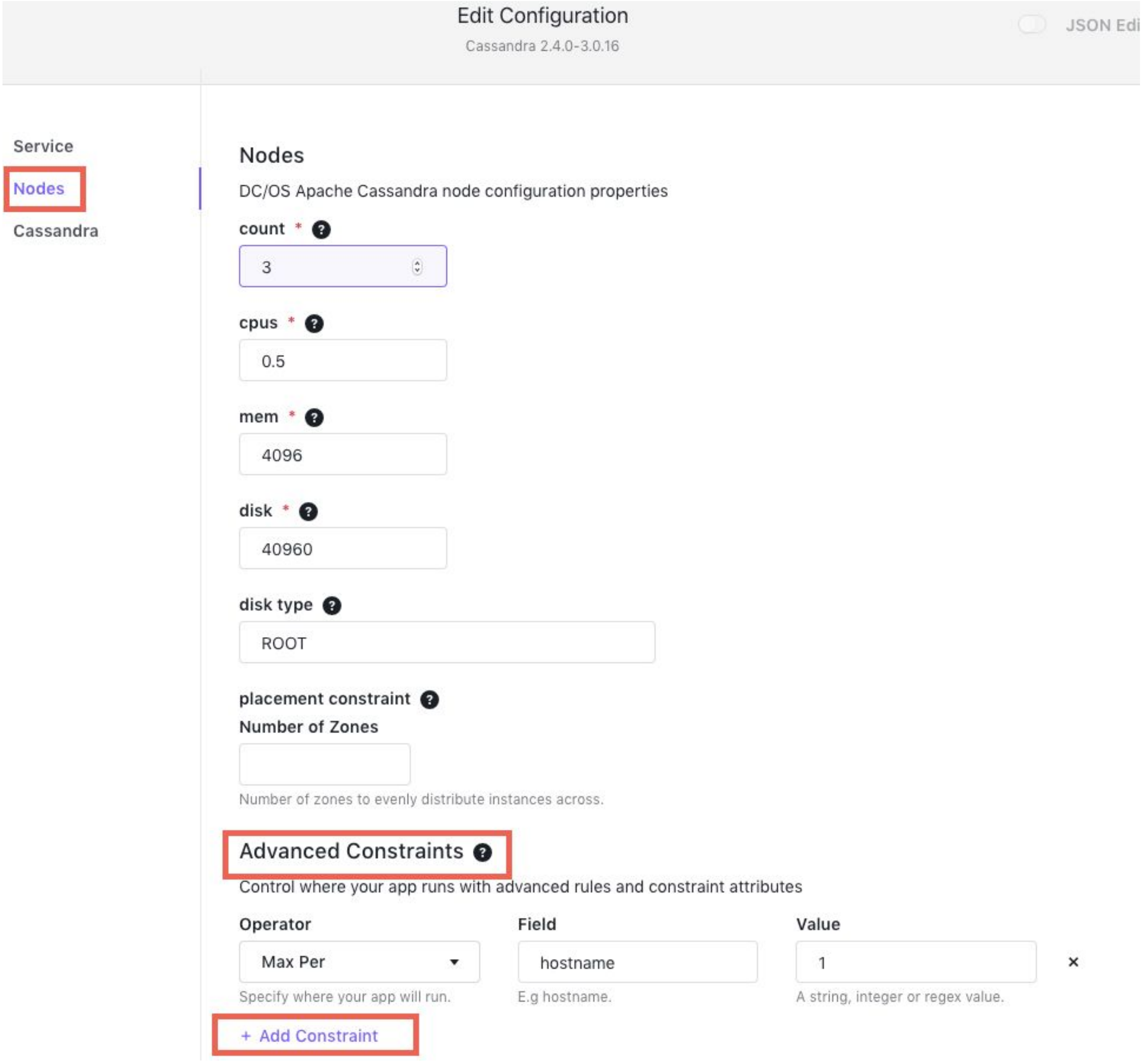
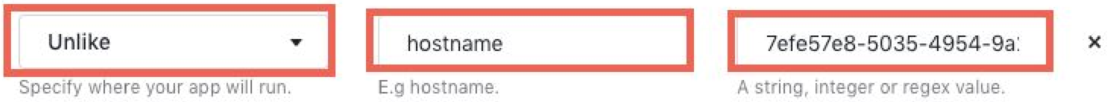
☐ ID	Name	Host	Zone	Region	Status	Health
node-2-server__99234028-4252-4aa0-acd6-e32ade502495	node-2-server	10.0.0.115	aws/us-west-2a	aws/us-west-2 (Local)	Running	●
node-1-server__cd4d7eae-6d3b-40b1-83f9-681a49b66985	node-1-server	10.0.2.178	aws/us-west-2a	aws/us-west-2 (Local)	Running	●
node-0-server__5e26103c-3b0e-40bf-a6aa-4edaddb0bc85	node-0-server	10.0.1.127	aws/us-west-2a	aws/us-west-2 (Local)	Running	●
☐ cassandra.7e05dcc8-fee1-11e8-95f8-8a202d582526	cassandra	10.0.3.102	aws/us-west-2a	aws/us-west-2 (Local)	Running	●

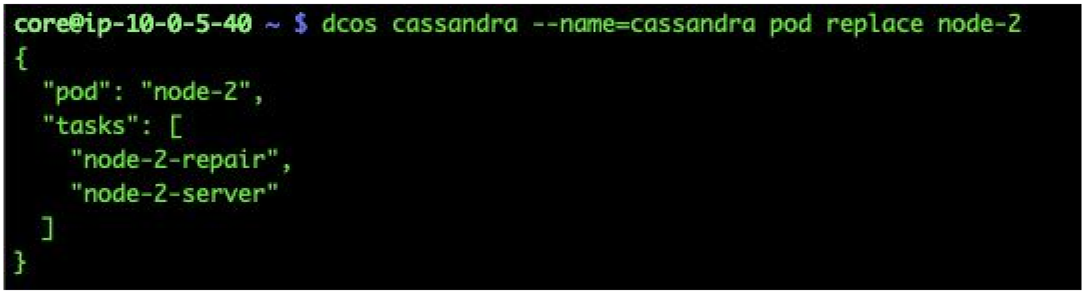
3 For each node, find the dcos agent id that the Cassandra node is running on:

	 Configuration Task ID node-2-server__99234028-4252-4aa0-acd6-e32ade502495 Service cassandra (7efe57e8-5035-4954-9a2b-c512a7ac2b54-0002) Node 10.0.0.115 (7efe57e8-5035-4954-9a2b-c512a7ac2b54-S6) IP Addresses

4	Create a list of nodes that need to be moved, e.g.:
	
	`node-2-server           10.0.0.115`
	`(7efe57e8-5035-4954-9a2b-c512a7ac2b54-S6)`
	
	`node-1-server           10.0.2.178`
	`(7efe57e8-5035-4954-9a2b-c512a7ac2b54-S7)`
	
	`node-0-server           10.0.1.127`
	`(7efe57e8-5035-4954-9a2b-c512a7ac2b54-S3)`

5	Create an exception for the Cassandra service to avoid nodes being located on an agent to be re-installed: `Cassandra-Service->Select “Edit”. Select “Nodes” → “Add Constraint”`

	 Edit Configuration Cassandra 2.4.0-3.0.16 JSON Edi Service Nodes Cassandra Nodes DC/OS Apache Cassandra node configuration properties count * 3 cpus * 0.5 mem * 4096 disk * 40960 disk type ROOT placement constraint Number of Zones Number of zones to evenly distribute instances across. Advanced Constraints Control where your app runs with advanced rules and constraint attributes Operator: Max Per — Specify where your app will run. Field: hostname — E.g hostname. Value: 1 — A string, integer or regex value. + Add Constraint
6	Add a constraint, such as: `Unlike->Hostname->UID of the agent (e.g. 7efe57e8-5035-4954-9a2b-c512a7ac2b54-S6)`  Unlike — Specify where your app will run. hostname — E.g hostname. 7efe57e8-5035-4954-9a — A string, integer or regex value. Save and run this new configuration.
7	From the dcos-CLI, perform a Pod Replace. -- *“Replace should be used only when the current instance of the pod should be completely destroyed. All persistent data (read: volumes) of the pod will be destroyed. Replace should be used when a DC/OS agent is being removed, is permanently down, or pod placement constraints need to be updated.”* ---> `dcos cassandra --name=cassandra pod list` ---> `dcos cassandra --name=cassandra pod replace node-2`

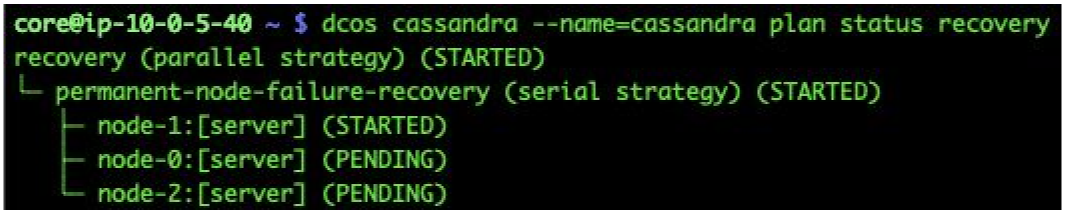


```
core@ip-10-0-5-40 ~ $ dcos cassandra --name=cassandra pod replace node-2
{
  "pod": "node-2",
  "tasks": [
    "node-2-repair",
    "node-2-server"
  ]
}
```

8 Check the status of the pod
---> `dcos cassandra --name=cassandra pod status node-2`

9 Check the status of the recovery plan
---> `dcos cassandra --name=cassandra plan status recovery`

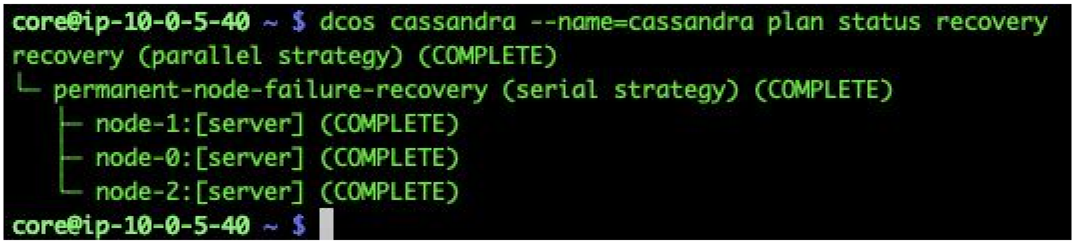
Showing here, node-1 is in a started state, others are in pending:



```
core@ip-10-0-5-40 ~ $ dcos cassandra --name=cassandra plan status recovery
recovery (parallel strategy) (STARTED)
└─ permanent-node-failure-recovery (serial strategy) (STARTED)
   ├─ node-1:[server] (STARTED)
   ├─ node-0:[server] (PENDING)
   └─ node-2:[server] (PENDING)
```

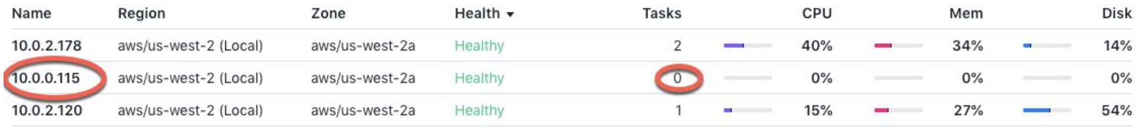
Continue monitoring. Until all show complete. You will see a “permanent-node-failure-recovery (serial strategy) (IN_PROGRESS)” shown while the recovery is ongoing.

When the pod replacement is complete, you will see that all nodes are in a complete state:



```
core@ip-10-0-5-40 ~ $ dcos cassandra --name=cassandra plan status recovery
recovery (parallel strategy) (COMPLETE)
└─ permanent-node-failure-recovery (serial strategy) (COMPLETE)
   ├─ node-1:[server] (COMPLETE)
   ├─ node-0:[server] (COMPLETE)
   └─ node-2:[server] (COMPLETE)
core@ip-10-0-5-40 ~ $
```

10 When it is “COMPLETE” Check in the DCOS-UI and verify that the agent (115) no longer has any tasks running.

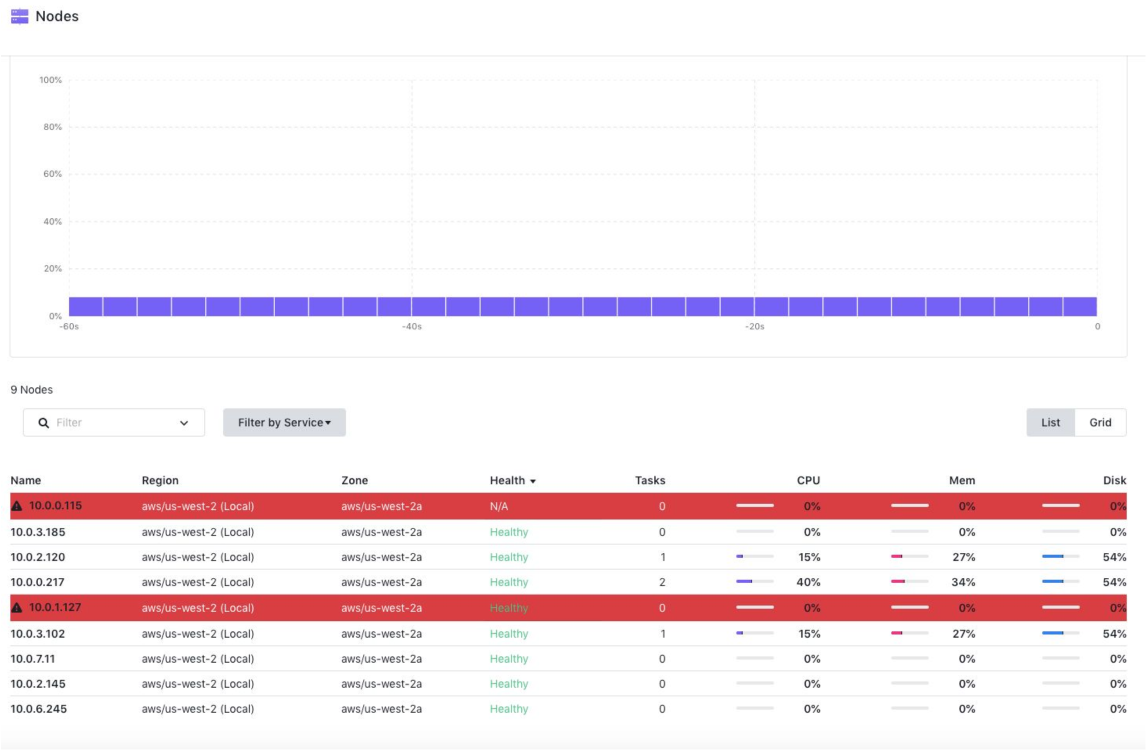


Name	Region	Zone	Health ▾	Tasks	CPU	Mem	Disk
10.0.2.178	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	2	40%	34%	14%
10.0.0.115	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.2.120	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	54%

11 Having verified that the Cassandra node has successfully moved from the DC/OS agent, go back and update the “Constraint” for Cassandra to include all agents you wish to evacuate Cassandra from. Do one at a time allowing the recovery plan to

sync/COMPLETE its “recovery”. Depending on the data set size, this could take some time…. Repeat steps 6-10 for each agent, updating the single Constraint with a list of the agent nodes (comma separated) to be free of Cassandra…

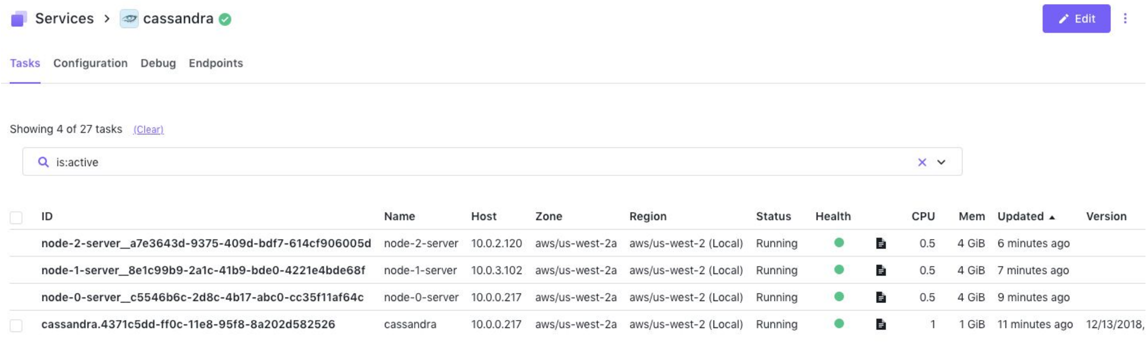
12 After completing the constraints and Cassandra node replacements, you can then remove the agents from DC/OS. I simply shut them down to have maintenance done.



9 Nodes

Name	Region	Zone	Health	Tasks	CPU	Mem	Disk
⚠ 10.0.0.115	aws/us-west-2 (Local)	aws/us-west-2a	N/A	0	0%	0%	0%
10.0.3.185	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.2.120	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	54%
10.0.0.217	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	2	40%	34%	54%
⚠ 10.0.1.127	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.3.102	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	1	15%	27%	54%
10.0.7.11	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.2.145	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%
10.0.6.245	aws/us-west-2 (Local)	aws/us-west-2a	Healthy	0	0%	0%	0%

13 Review the currently running Cassandra, and make sure the service is still successfully running:



Showing 4 of 27 tasks

is:active

ID	Name	Host	Zone	Region	Status	Health		CPU	Mem	Updated	Version
node-2-server__a7e3643d-9375-409d-bdf7-614cf906005d	node-2-server	10.0.2.120	aws/us-west-2a	aws/us-west-2 (Local)	Running	●	🗎	0.5	4 GiB	6 minutes ago	
node-1-server__8e1c99b9-2a1c-41b9-bde0-4221e4bde68f	node-1-server	10.0.3.102	aws/us-west-2a	aws/us-west-2 (Local)	Running	●	🗎	0.5	4 GiB	7 minutes ago	
node-0-server__c5546b6c-2d8c-4b17-abc0-cc35f11af64c	node-0-server	10.0.0.217	aws/us-west-2a	aws/us-west-2 (Local)	Running	●	🗎	0.5	4 GiB	9 minutes ago	
cassandra.4371c5dd-ff0c-11e8-95f8-8a202d582526	cassandra	10.0.0.217	aws/us-west-2a	aws/us-west-2 (Local)	Running	●	🗎	1	1 GiB	11 minutes ago	12/13/2018,

14 Return to service / normal operations ----
If the DC/OS agents are re-installed, restored to service within the cluster, they can once again have services scheduled on them. Edit the configuration for Cassandra again and remove any constraints used for maintenance, or update the constraints

	accordingly to how you want the environment configured. (e.g. constraints which match the service requirements)