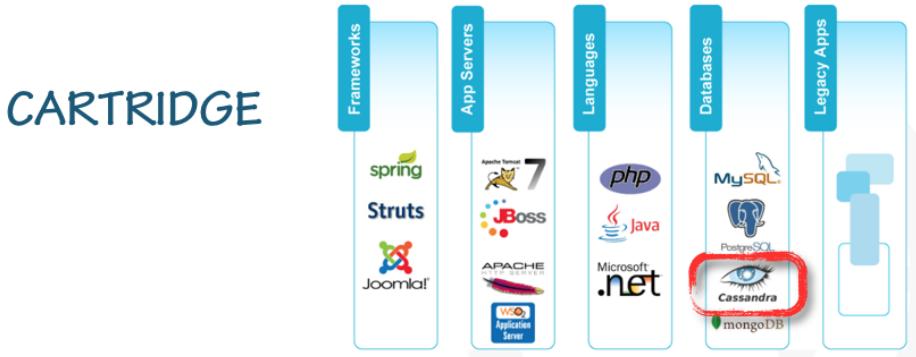


Docker based Cassandra Cartridge for Apache Stratos - User Guide

1. [Initial Design of the Cassandra Cartridge](#)
2. [Starting up the Virtual Machine](#)
3. [Virtual Machine Image Details](#)
[3.1 Locations](#)
[3.2 Credentials](#)
[3.3 URLs](#)
4. [Servers and setups in the Virtual Machine](#)
5. [Building the Docker](#)
6. [Apache Stratos Configuration](#)
[6.1 Partition](#)
[6.2 Autoscale Policy](#)
[6.3 Deployment Policy](#)
[6.4 Cassandra-Cartridges](#)
7. [Subscribing to cartridge](#)
8. [Cassandra Docker](#)
9. [Running an Application](#)
10. [Reference](#)

1. Initial Design of the Cassandra Cartridge



VIRTUAL
MACHINE

2. Starting up the Virtual Machine

The environment could be set up either in the same machine or a virtual machine. Setting up the environment in the same machine would require to install and configure openstack with devstack and Apache Stratos.

In this case, an image with openstack and apache stratos configured has been used to minimize the time taken for installation and configurations.

The following are the steps that should be followed to set up the environment in a Mac OS. As prerequisite, VirtualBox should be installed in the machine.

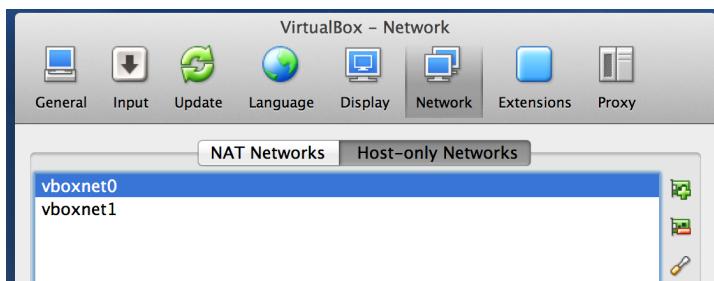
- Follow the link [Stratos-PaaS-Image.ova] to download the stratos pass image
- Import Appliance



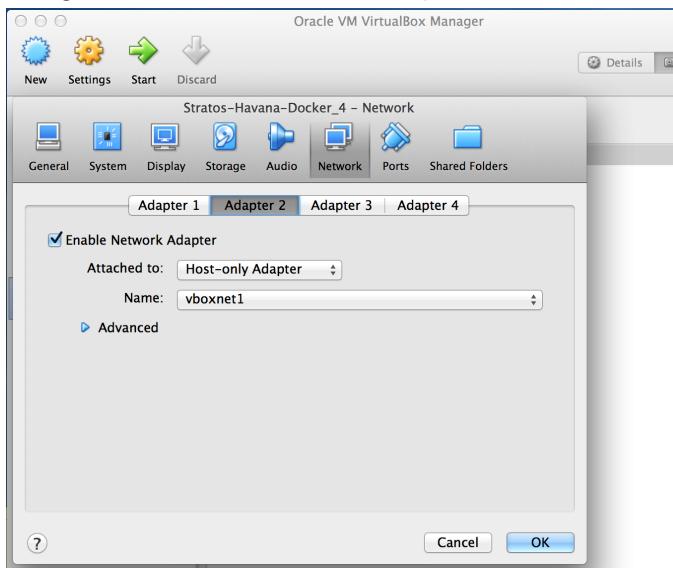
- Network Settings



Create 2 host only networks with the IP
192.168.57.30 and 192.168.92.3

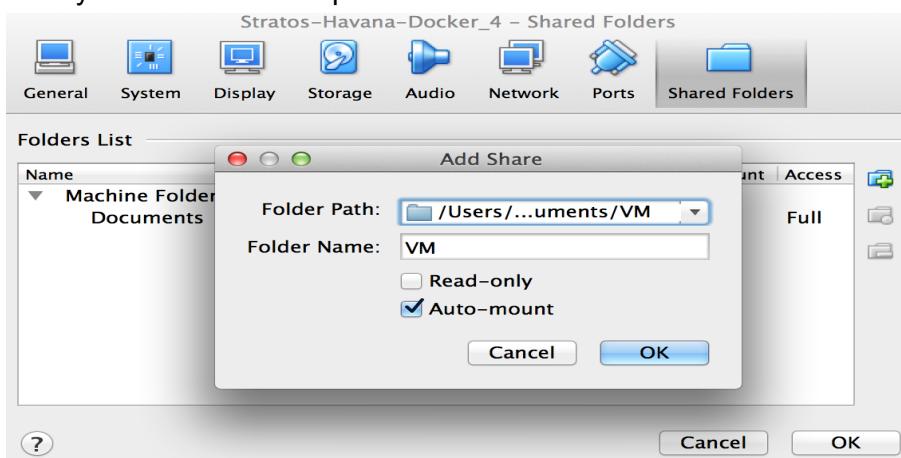


Assign them to the network adapters of the virtual machine as Host-only Adapter



- Shared Folder

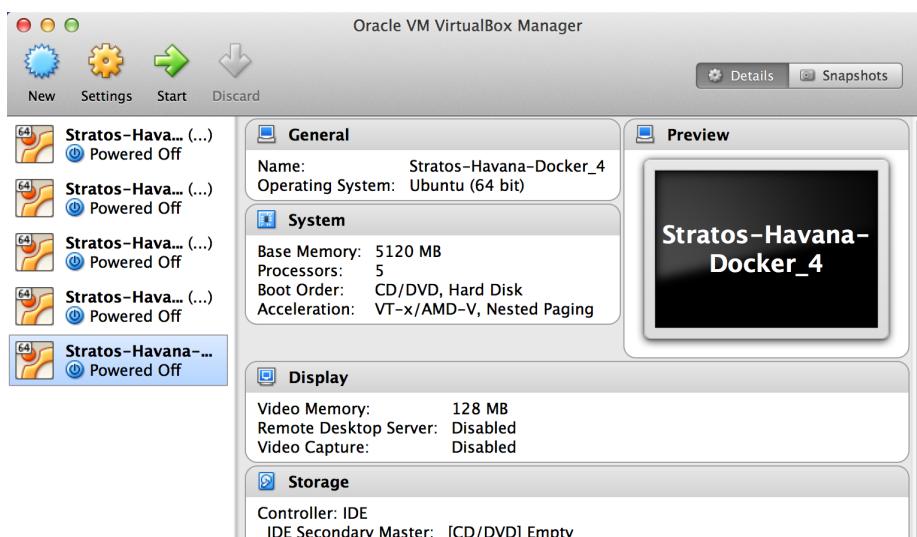
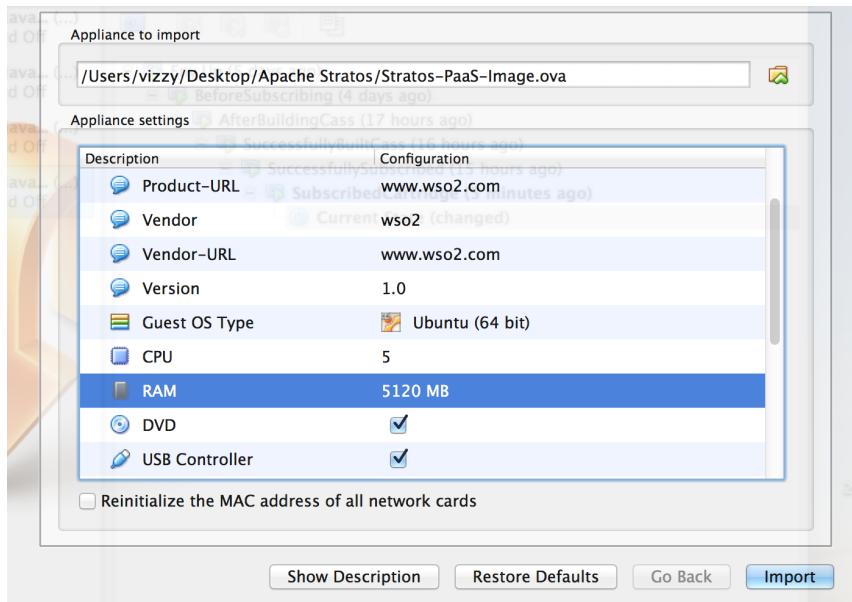
Modify the shared folder path to a folder in the local machine to share it with the virtual machine.



- VM Specifications

Next step would be specifying the specifications for the virtual machine. In this scenario the table below shows the allocations made for the virtual machine.

Specification	Local Machine	Allocation for the Virtual Machine
CPU	8 CPUs	5 CPUs
RAM	8 GB	5 GB



3. Virtual Machine Image Details

3.1 Locations

The following are the paths of all the servers, tools and components available inside the Image.

Devstack : /home/stratos/workspace/devstack

Stratos Source : /home/stratos/workspace/stratos

Java : /home/stratos/tools/java/jdk

Maven : /home/stratos/tools/build/maven

GitServer : /home/stratos/tools/gitserver/gitblit

Git Local Repo : /home/stratos/workbench/git-local-repo

Cartridge Builds : /home/stratos/workbench/cartridges

Cartridge Definition Files : /home/stratos/workbench/cartridge-definitions

Stratos Installation : /opt/stratos/apache-stratos-default

Stratos Installer : /opt/stratos-installer

Active MQ : /opt/stratos/apache-activemq-5.9.1

Load Balancer : /opt/apache-stratos-load-balancer-4.0.0

3.2 Credentials

Credentials of the tools and components inside the Image are given in the following table.

Tool/Component	UserId	Password
OS (Ubuntu 12.04)	root	password
	stratos	password
OpenStack	demo	password
	admin	password
Stratos	admin	admin
GitBlits	admin	admin
Load Balancer	admin	admin
Active MQ	admin	admin
Cartridge Instances	root	password

3.3 URLs

The following table shows all the URLs for the services.

Service	URL
OpenStack	http://dev.stratos.org http://192.168.57.30 http://localhost
Stratos	https://dev.stratos.org:9445/console https://192.168.57.30:9445/console https://localhost:9445/console
GitBlts	http://dev.stratos.org:7070 http://192.168.57.30:7070 http://localhost:7070
Active MQ	http://dev.stratos.org:8161/admin http://192.168.57.30:8161/admin http://localhost:8161/admin
Load Balancer	https://dev.stratos.org:9443/carbon https://192.168.57.30:9443/carbon https://localhost:9443/carbon

4. Servers and setups in the Virtual Machine

Once started the virtual machine, the following steps should be followed to start up the environment for stratos.

- GitBlit

GitBlit is available in the mentioned path in the following diagram. It could be used as the git repository if required for the cartridge in scenarios where it need to clone when subscribing. But for the cassandra cartridge scenario, it is not needed since it is just a database cartridge.

```
GitBlit stratos@Dev-PC:~/tools
stratos@Dev-PC:~$ stratos@Dev-PC:~$ cd tools/gitserver/gitblit/
stratos@Dev-PC:~/tools/gitserver/gitblit$ ./gitblit
```

- Devstack

The devstack is available in the path shown in the following diagram.
.rejoin-stack.sh command would invoke openstack over devstack.

```
GitBlit DevStack stratos@Dev-PC:~$ stratos@Dev-PC:~$ cd workspace/devstack/
stratos@Dev-PC:~/workspace/devstack$ ./rejoin-stack.sh
```

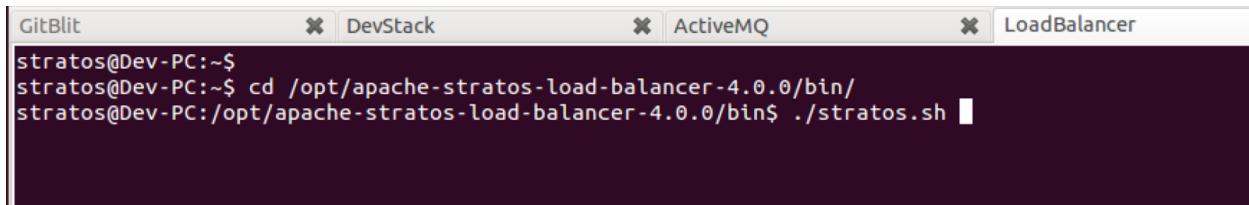
- ActiveMQ

ActiveMQ is available in the path shown in the following diagram.
.activemq start command would start ActiveMQ.

```
GitBlit DevStack ActiveMQ stratos@Dev-PC:~$ stratos@Dev-PC:~$ cd /opt/stratos/apache-activemq-5.9.1/bin/
stratos@Dev-PC:/opt/stratos/apache-activemq-5.9.1/bin$ ./activemq start
```

- Load Balancer

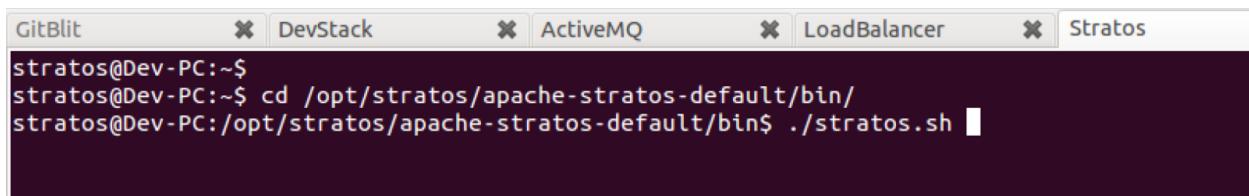
The load balancer could be started with from the path mentioned in the following diagram.
./stratos.sh command would start the load balancer.



```
stratos@Dev-PC:~$ stratos@Dev-PC:~$ cd /opt/apache-stratos-load-balancer-4.0.0/bin/ stratos@Dev-PC:/opt/apache-stratos-load-balancer-4.0.0/bin$ ./stratos.sh
```

- Apache Stratos

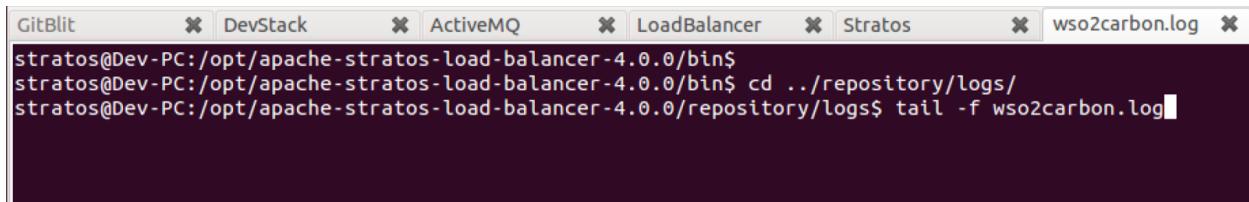
Stratos is available in the path mentioned in the following diagram.
./stratos.sh command would start Stratos.



```
stratos@Dev-PC:~$ stratos@Dev-PC:~$ cd /opt/stratos/apache-stratos-default/bin/ stratos@Dev-PC:/opt/stratos/apache-stratos-default/bin$ ./stratos.sh
```

- wso2carbon.log

To access the wso2carbon log, should follow to the path mentioned in the following diagram and type the command mentioned in the following diagram.



```
stratos@Dev-PC:/opt/apache-stratos-load-balancer-4.0.0/bin$ stratos@Dev-PC:/opt/apache-stratos-load-balancer-4.0.0/bin$ cd ../repository/logs/ stratos@Dev-PC:/opt/apache-stratos-load-balancer-4.0.0/repository/logs$ tail -f wso2carbon.log
```

After completing all the above steps, it is able to access openstack and apache stratos from the URLs mentioned in the URL Table.

GitBlit server is accessible through <http://192.168.57.30:7070/>

Openstack is accessible through <http://192.168.57.30>

Apache Stratos is accessible through <https://192.168.57.30:9445/console>

5. Building the Docker

The developed docker file for the cartridge and all its required packs and scripts should be available in the virtual machine to build. If the docker file is in the local machine, it could be placed in the shared folder mentioned when configuring the virtual machine.

If the folder named ‘SharedFolder’ has shared to the virtual machine. Then it could be accessible inside the virtual machine from the directory “/media/sf_SharedFolder”

Once it is inside the directory then the following commands should be used to build and push the cartridge.

- docker build -t name-cartridge .

once it is built, it will prompt with a message saying that it is build successfully.

- docker tag name-cartridge 192.168.57.30:5042/name-cartridge

then the command “docker images” can be used to view the built cartridges

- docker push 192.168.57.30:5042/name-cartridge

once the cartridge is successfully pushed, it is possible to view all the images by going to the “tools/devstack” directory and typing the following commands

```
tools/devstack$ . openrc admin admin
```

```
tools/devstack$ glance image-list
```

or it could be viewed at the openstack images.

	Image Name	Type	Status	Public	Protected	Format	Actions
<input type="checkbox"/>	cassandra-cartridge:latest	Image	Active	Yes	No	RAW	<button>Edit</button> <button>More</button>
<input type="checkbox"/>	-	Image	Active	Yes	No	RAW	<button>Edit</button> <button>More</button>

6. Apache Stratos Configuration

Once the cartridge is successfully pushed, then next step would be to do the configurations for the cartridge at Apache Stratos.

The screenshot shows a web browser window titled 'Configure Stratos - Partition Deployment'. The URL is https://dev.stratos.org:9445/console/configure_stratos.jag. The top navigation bar includes tabs for 'Instances - OpenSta...', 'Partition Deployment', 'File Edit View History Bookmarks Tools Help', and a status bar showing '1:06 PM' and 'Stratos Developer'. Below the navigation is a header with the Apache Stratos logo and links for 'My Cartridges', 'Configure Stratos' (which is selected), 'Tenant Mgt', and 'admin'. A secondary navigation bar below the header has tabs for 'Partitions' (selected), 'Auto scale Policies', 'Deployments Policies', 'LBs', 'Cartridges', and 'Multi-Tenant Services'. A red button labeled 'Configuration Wizard' is visible. On the left, there's a vertical toolbar with icons for various management functions. The main content area is titled 'Partition Configuration' and contains a large text input field for JSON configuration. An example configuration is provided in the field:

```
{  
    "id": "zone-1",  
    "provider": "ec2",  
    "property": [  
        {  
            "name": "region",  
            "value": "ap-southeast-1"  
        },  
        {  
            "name": "zone",  
            "value": "ap-southeast-1a"  
        }  
    ]  
}
```

Below the input field are two buttons: 'Deploy' and 'P1'.

6.1 Partition

```
{  
    "id": "P1",  
    "provider": "openstack",  
    "property": [  
        {  
            "name": "region",  
            "value": "RegionOne"  
        }  
    ]  
}
```

6.2 Autoscale Policy

```
{  
    "id": "economyPolicy",  
    "loadThresholds": {  
        "requestsInFlight": {  
            "average": "50",  
            "gradient": "0",  
            "secondDerivative": "0",  
            "scaleDownMarginOfGradient": "1.0",  
            "scaleDownMarginOfSecondDerivative": "0.2"  
        },  
        "memoryConsumption": {  
            "average": "100",  
            "gradient": "10",  
            "secondDerivative": "1",  
            "scaleDownMarginOfGradient": "0.5",  
            "scaleDownMarginOfSecondDerivative": "0.1"  
        }  
    }  
}
```

```

        "average": "70",
        "gradient": "0",
        "secondDerivative": "0",
        "scaleDownMarginOfGradient": "1.0",
        "scaleDownMarginOfSecondDerivative": "0.2"
    },
    "loadAverage": {
        "average": "70",
        "gradient": "0",
        "secondDerivative": "0",
        "scaleDownMarginOfGradient": "1.0",
        "scaleDownMarginOfSecondDerivative": "0.2"
    }
}
}
}

```

6.3 Deployment Policy

```
{
    "id": "cassandra-openstack",
    "partitionGroup": {
        "id": "openstack",
        "partitionAlgo": "one after another",
        "partition": [
            {
                "id": "P1",
                "partitionMax": "3",
                "partitionMin": "1"
            }
        ]
    }
}
```

6.4 Cassandra-Cartridges

ImageId could be found at the pushed image at openstack. In this scenario the maximum instances limit was made 2 because of the local memory in the virtual machine

```
{
    "type": "cassandra",
    "provider": "data",
    "host": "dev.stratos.org",
    "displayName": "Cassandra",
    "description": "Cassandra Cartridge",
    "version": "2.1.0",
    "defaultAutoscalingPolicy": "economyPolicy",
    "defaultDeploymentPolicy": "cassandra-openstack",
    "multiTenant": "false",
    "portMapping": [
        {

```

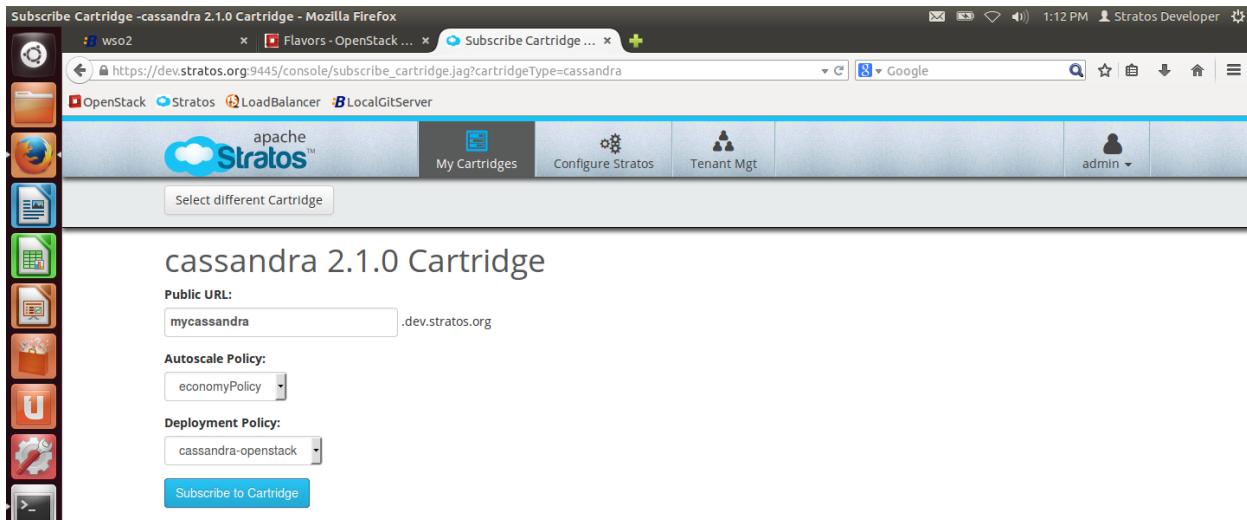
```

        "protocol":"http",
        "port":"9160",
        "proxyPort":"8280"
    }
],
"deployment": {},
"iaasProvider": [
    {
        "type":"openstack",
        "imageId":"RegionOne/1cb570a3-2c91-4cc4-8ae5-d6a1d7638304",
        "maxInstanceLimit":"2",
        "property": [
            {
                "name": "instanceType",
                "value": "RegionOne/3"
            },
            {
                "name": "keyPair",
                "value": "demo"
            },
            {
                "name": "securityGroups",
                "value": "open-all"
            }
        ]
    }
],
"property": [
    {
        "name": "payload_parameter.MB_IP",
        "value": "192.168.57.30"
    },
    {
        "name": "payload_parameter.MB_PORT",
        "value": "61616"
    },
    {
        "name": "payload_parameter.CEP_IP",
        "value": "192.168.57.30"
    },
    {
        "name": "payload_parameter.CEP_PORT",
        "value": "7613"
    }
]
}

```

7. Subscribing to cartridge

The following diagram shows the subscribing console in Apache Stratos, where a public URL, autoscale policy and deployment policy could be set before subscribing to the cartridge.



Once the cartridge is subscribed, the Stratos terminal shows the details of the process.

The image below shows that it is trying to spawn an instance via cloud.

```
[2014-10-20 11:41:36,285] INFO [org.apache.stratos.cloud.controller.impl.CloudControllerServiceImpl] - All partitions [P1] were validated successfully, against the Cartridge: cassandra
[2014-10-20 11:41:37,381] INFO [org.apache.stratos.autoscaler.util.AutoScaleUtil] - Partition context has been added: [partition] P1
[2014-10-20 11:41:37,381] INFO [org.apache.stratos.autoscaler.util.AutoScaleUtil] - Network partition context has been added: [network partition] openstack
[2014-10-20 11:41:37,381] INFO [org.apache.stratos.autoscaler.util.AutoScaleUtil] - Cluster monitor created: ClusterMonitor [clusterId=mycassandra.cassandra.domain, serviceId=cassandra, deploymentPolicy=Deployment Policy [id]cassandra-openstack [partitions] [org.apache.stratos.cloud.controller.stub.deployment.partition.Partition@458afa38], autoscalePolicy=ASPolicy [id=economyPolicy, displayName=null, description=null, lbReferenceType=null]
[2014-10-20 11:41:37,382] INFO [org.apache.stratos.autoscaler.message.receiver.topology.AutoScaleTopologyEventReceiver] - Cluster monitor has been added successfully: [cluster] mycassandra.cassandra.domain
[2014-10-20 11:41:37,384] WARN [org.apache.stratos.messaging.message.processor.topology.ClusterCreatedMessageProcessor] - Cluster already exists in service: [service] cassandra [cluster] mycassandra.cassandra.domain
[2014-10-20 11:41:37,385] INFO [org.apache.stratos.manager.topology.receiver.StratosManagerTopologyEventReceiver] - [ClusterCreatedEventListener] Received: class org.apache.stratos.messaging.event.topology.ClusterCreatedEvent
[2014-10-20 11:42:37,405] INFO [org.apache.stratos.autoscaler.client.cloud.controller.CloudControllerClient] - Trying to spawn an instance via cloud controller: [cluster] mycassandra.cassandra.domain [partition] P1 [lb-cluster] null [network-partition-id] openstack
```

If the stratos agent communication is successful, then it will prompt that the member activated as shown in the following diagram.

```

File Edit View Search Terminal Tabs Help
GB DevStack ActiveMQ LB Stratos log stratos@DevPC:/me...
memberId=mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554, nodeId=RegionOne/de4fe26d-384-a8ad-5366017258f3, clusterId=mycassandra.cassandra.domain, cartridgeType=cassandra, privateIpAddress=10.11.12.2, publicIpAddress=192.168.92.1, initTime=141378557406, lbClusterId=null, networkPartitionId=openstack
[2014-10-20 11:42:58,577] INFO {org.apache.stratos.cloud.controller.topology.TopologyEventPublisher} - Publishing instance spawned event: [service] cassandra [cluster] mycassandra.cassandra.domain [network-partition] openstack [partition] P1 [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:42:58,582] INFO {org.apache.stratos.messaging.message.processor.topology.InstanceSpawnedMessageProcessor} - Member created: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:42:58,587] INFO {org.apache.stratos.manager.topology.receiver.StratosManagerTopologyEventReceiver} - [InstanceSpawnedEventListener] Received: class org.apache.stratos.messaging.event.topology.InstanceSpawnedEvent
[2014-10-20 11:42:58,588] WARN {org.apache.stratos.messaging.message.processor.topology.InstanceSpawnedMessageProcessor} - Member already exists: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:05,450] INFO {org.apache.stratos.cloud.controller.topic.Instance.status.InstanceStatusEventMessageDelegator} - Instance status event message received from queue: org.apache.stratos.messaging.event.Instance.status.InstanceStartedEvent
[2014-10-20 11:43:05,451] INFO {org.apache.stratos.manager.listener.InstanceStatusListener} - Instance status message received
[2014-10-20 11:43:05,451] INFO {org.apache.stratos.manager.listener.InstanceStatusListener} - Event class name: org.apache.stratos.messaging.event.instance.status.InstanceStartedEvent
[2014-10-20 11:43:05,457] INFO {org.apache.stratos.manager.listener.InstanceStatusListener} - Cluster id: mycassandra.cassandra.domain
[2014-10-20 11:43:05,457] INFO {org.apache.stratos.cloud.controller.topology.TopologyBuilder} - member started event adding status started
[2014-10-20 11:43:05,497] INFO {org.apache.stratos.cloud.controller.topology.TopologyEventPublisher} - Publishing member started event: [service] cassandra [cluster] mycassandra.cassandra.domain [network-partition] openstack [partition] P1 [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:05,503] INFO {org.apache.stratos.messaging.message.processor.topology.MemberStartedMessageProcessor} - Member started: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:05,504] INFO {org.apache.stratos.manager.topology.receiver.StratosManagerTopologyEventReceiver} - [MemberStartedEventListener] Received: class org.apache.stratos.messaging.event.topology.MemberStartedEvent
[2014-10-20 11:43:05,505] WARN {org.apache.stratos.messaging.message.processor.topology.MemberStartedMessageProcessor} - Member already started: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:21,878] INFO {org.apache.stratos.manager.listener.InstanceStatusListener} - Instance status message received
[2014-10-20 11:43:21,878] INFO {org.apache.stratos.cloud.controller.topic.Instance.status.InstanceStatusEventMessageDelegator} - Instance status event message received from queue: org.apache.stratos.messaging.event.Instance.status.InstanceActivatedEvent
[2014-10-20 11:43:21,886] INFO {org.apache.stratos.manager.listener.InstanceStatusListener} - Event class name: org.apache.stratos.messaging.event.instance.status.InstanceActivatedEvent
[2014-10-20 11:43:21,900] INFO {org.apache.stratos.cloud.controller.topology.TopologyBuilder} - member started event adding status activated
[2014-10-20 11:43:21,941] INFO {org.apache.stratos.cloud.controller.topology.TopologyEventPublisher} - Publishing member activated event: [service] cassandra [cluster] mycassandra.cassandra.domain [network-partition] openstack [partition] P1 [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:21,947] INFO {org.apache.stratos.messaging.message.processor.topology.MemberActivatedMessageProcessor} - Member activated: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:21,954] INFO {org.apache.stratos.autoscaler.message.receiver.topology.AutoscalerTopologyEventReceiver} - Member stat context has been added successfully: [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:21,954] WARN {org.apache.stratos.messaging.message.processor.topology.MemberActivatedMessageProcessor} - Member already activated: [service] cassandra [cluster] mycassandra.cassandra.domain [member] mycassandra.cassandra.domaine4bd6017-5e44-4780-b4e7-682d5e919554
[2014-10-20 11:43:21,960] INFO {org.apache.stratos.manager.topology.receiver.StratosManagerTopologyEventReceiver} - [MemberActivatedEventListener] Received: class org.apache.stratos.messaging.event.topology.MemberActivatedEvent

```

My Cartridges - Mozilla Firefox

My Cartridges

Un-subscribe

mycassandra-(Cassandra - 2.1.0)

Cassandra Cartridge

When the member becomes active, it could be seen from openstack images as the instance would have an active status as shown in the diagram below.

Project	Host	Name	Image Name	IP Address	Size	Status	Task	Power State	Uptime	Actions
demo	Dev-PC	mycassandra-421	cassandra-cartridge:latest	10.11.12.2 192.168.92.1 192.168.92.2	m1.medium 4GB RAM 2 VCPU 40.0GB Disk	Active	None	Running	1 day, 6 hours	Edit Instance More

From the IP address (10.11.12.2) of the instance, It is possible to SSH into the instance from inside the virtual machine.

```
stratos@Dev-PC:/opt/stratos/apache-stratos-default/bin$ ssh root@10.11.12.2
The authenticity of host '10.11.12.2 (10.11.12.2)' can't be established.
ECDSA key fingerprint is b6:5c:31:2c:72:21:be:00:dd:eb:d9:1d:59:85:4d:6c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.11.12.2' (ECDSA) to the list of known hosts.
root@10.11.12.2's password: [REDACTED]
```

```
stratos@Dev-PC:/opt/stratos/apache-stratos-default/bin$ ssh root@10.11.12.2
The authenticity of host '10.11.12.2 (10.11.12.2)' can't be established.
ECDSA key fingerprint is b6:5c:31:2c:72:21:be:00:dd:eb:d9:1d:59:85:4d:6c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.11.12.2' (ECDSA) to the list of known hosts.
root@10.11.12.2's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.11.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

From the public IP (192.168.92.1) or the floating IP (192.168.92.2), it is possible to SSH into the instance from the local machine.

```
vizzy — root@instance-00000001:~ — ssh — 80x24
Last login: Thu Oct 23 16:31:19 on console
Vishanth-MacBook-Pro:~ vizzy$ ssh root@192.168.92.1
The authenticity of host '192.168.92.1 (192.168.92.1)' can't be established.
RSA key fingerprint is 22:f9:91:45:c7:ad:8e:8a:ba:ce:e5:68:09:c0:be:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.92.1' (RSA) to the list of known hosts.
root@192.168.92.1's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.11.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Wed Oct 22 10:34:58 2014 from 10.11.12.1
root@instance-00000001:~# [REDACTED]
```

8. Cassandra Docker

The cassandra docker could be found from the following git profile.

<https://github.com/Vishanth/Cassandra-Docker>

9. Running an Application

Create a maven project using eclipse.

Add the following dependency in the pom.xml file

```
<dependency>
    <groupId>com.datastax.cassandra</groupId>
    <artifactId>cassandra-driver-core</artifactId>
    <version>2.1.0</version>
</dependency>
```

A Simple java class which connects to the cassandra cluster which the public ip, in this scenario the instance has the IP 192.168.92.1.

```
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.ResultSet;
import com.datastax.driver.core.Row;
import com.datastax.driver.core.Session;

public class CassandraConnector {

    public static Cluster cluster;
    public static Session session;

    public static Cluster connect(String node){
        return Cluster.builder().addContactPoints(node).build();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        cluster = connect("192.168.92.1");
        session = cluster.connect();

        session.execute("CREATE KEYSPACE demo WITH REPLICATION = "
                + "{'class' : 'SimpleStrategy' , 'replication_factor' : 1};");

        session.execute("USE demo;");
        session.execute("CREATE TABLE IF NOT EXISTS testTable (id varchar PRIMARY KEY , name varchar);");

        session.execute("INSERT INTO testTable (id , name ) VALUES ( '1001' , 'Rooney' );");
        session.execute("INSERT INTO testTable (id , name ) VALUES ( '1002' , 'Scholes' );");

        String query_selectAll="SELECT * FROM testTable;";
        ResultSet result = session.execute(query_selectAll);
        for (Row rows: result){
            System.out.println(rows.toString());
        }
        session.close();
        cluster.close();
    }
}
```

10. Reference

Apache Cassandra - <http://cassandra.apache.org>

Download link - <http://cassandra.apache.org/download/>

Getting Started Documentation - <http://wiki.apache.org/cassandra/GettingStarted>

Apache Stratos - <http://stratos.apache.org>

Documentation - <https://cwiki.apache.org/confluence/display/STRATOS/Home>

Docker - <https://docs.docker.com>

Stratos-PaaS-Image Owner - Roshan Deniyage