4.1.0 Stratos M2 Developer Preview

Main Features

- Docker support using Google Kubernetes and CoreOS
- Auto scaling Docker Containers
- Manual scaling Docker Containers
- Git based artifact deployment for Docker
- CLI support for Docker deployments
- MQTT support (removal of JNDI)

Prerequisite

- Setting up the Kubernetes-CoreOS cluster in your local machine
  - Install Vagrant 1.6.5 and Oracle VM VirtualBox Manager 4.3.14 in your local machine.
  - Get a GIT clone of Vagrant Kubernetes Setup onto your machine.
  - Navigate to the cloned repository directory (**SETUP_HOME**).
  - Run `up.sh` script file - `{SETUP_HOME}$ ./up.sh` (You might have to use `sudo` in some cases.)
  - Above command will start-up 4 VMs, namely discovery, master, minion-1 and minion-2.
  - SSH to master node ; `{SETUP_HOME}$ vagrant ssh master`
  - Pull Stratos PHP Docker Image from DockerHub into master node or into the local machine.

```
docker pull apachestratos/php:4.1.0-m2
```

  - Import downloaded Stratos PHP Docker image as a tarball.

```
docker save -o stratos-php-latest.tar  apachestratos/php:4.1.0-m2
```

  - SCP the Stratos PHP Docker Image tarball to minion-1 and minion-2. You can find the private key file which you can use to SCP, in the **{SETUP_HOME}/ssh.config** file, against **IdentityFile** attribute.

```
scp -i ~/.vagrant.d/insecure_private_key stratos-php-latest.tar
core@172.17.8.101:.
```

  - SSH to minion-1 (`{SETUP_HOME}$ vagrant ssh minion-1`) and minion-2 (`{SETUP_HOME}$`

vagrant ssh minion-2)

- Load Stratos PHP Docker Image from tarball to minion-1 and minion-2

```
docker load -i stratos-php-latest.tar
```

- Verify that the image is properly loaded by issuing `docker images` in each node;

```
core@master ~ $ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
apachestratos/php   4.1.0-m2            434d3f9da2eb        13 hours ago
705.4 MB
```

- Download and extract Apache ActiveMQ 5.10.0 or later and start ActiveMQ - `{ACTIVEMQ_HOME}$ ./bin/activemq start` Please make sure mqtt transport connector is enabled in the ActiveMQ configuration file;**{ACTIVEMQ_HOME}/conf/activemq.xml**.
- Build Stratos master code, copy (from {**STRATOS_SOURCE}/products/stratos/modules/distribution/target/**) and extract the binary **apache-stratos-4.1.0-SNAPSHOT.zip** to a preferred directory (**STRATOS_HOME**).
- Change the **MgtHostName** and **HostName** elements' values in **{STRATOS_HOME}/repository/conf/carbon.xml**such that they point to the private IP address of your local machine.
- Change the **dataBridgeConfiguration.thriftDataReceiver.hostName** element's value in **{STRATOS_HOME}/repository/conf/data-bridge/data-bridge-config.xml** to the private IP address of your machine.
- Change the property named **"java.naming.provider.url"** value to **tcp://{MB_IP}:61616** in **{STRATOS_HOME}/repository/deployment/server/outputeventadaptors/JMSOutputAdaptor.xml** file.
- Change the **expiryTimeout** element's value in **{STRATOS_HOME}/repository/conf/autoscaler.xml** to **30000** (it is the maximum time a member can be in the pending state)
- Start Stratos using `{STRATOS_HOME}$ ./bin/stratos.sh start` command.

Testing M2

1. Register Kubernetes-CoreOS Host Cluster in Stratos

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d @"kub-register.json" -k  -u
admin:admin "https://localhost:9443/stratos/admin/kubernetes/deploy/group"
```

CLI Command

```
deploy-kubernetes-group -p kub-register.json
```

- Please refer support using Apache Stratos CLI Tool to configure CLI tool

kub-register.json

```
{
      "groupId": "KubGrp1",
```

```
  "description": "Kubernetes CoreOS cluster on EC2 ",
  "kubernetesMaster": {
             "hostId" : "KubHostMaster1",
             "hostname" : "master.dev.kubernetes.example.org",
             "hostIpAddress" : "172.17.8.100",
             "property" : [
                 {
                    "name": "prop1",
                    "value": "val1"
                 },
                 {
                    "name": "prop2",
                    "value": "val2"
                 }
             ]
  },

  "portRange" : {
     "upper": "5000",
     "lower": "4500"
  },

  "kubernetesHosts": [
        {
               "hostId" : "KubHostSlave1",
               "hostname" : "slave1.dev.kubernetes.example.org",
               "hostIpAddress" : "172.17.8.101",
               "property" : [
                   {
                       "name": "prop1",
                       "value": "val1"
                   },
                   {
                       "name": "prop2",
                       "value": "val2"
                   }
               ]
          },
          {
               "hostId" : "KubHostSlave2",
               "hostname" : "slave2.dev.kubernetes.example.org",
               "hostIpAddress" : "172.17.8.102",
               "property" : [
                   {
                       "name": "prop1",
                       "value": "val1"
                   },
                   {
                       "name": "prop2",
                       "value": "val2"
                   }
               ]
          }
],

"property": [
       {
              "name": "prop1",
              "value": "val1"
```

```
        },
        {
                "name": "prop2",
                "value": "val2"
```

```
            }
    ]
}
```

Verify Kubernetes-CoreOS Host Cluster Registration

Curl Command

```
curl  -k  -u admin:admin
"https://localhost:9443/stratos/admin/kubernetes/group/KubGrp1"
```

Response

```
{"kubernetesGroup":{"description":"Kubernetes CoreOS cluster on EC2
","groupId":"KubGrp1","kubernetesHosts":[{"hostId":"KubHostSlave1","hostIpAddress":"17
2.17.8.101","hostname":"slave1.dev.kubernetes.example.org","property":[{"name":"prop1"
,"value":"val1"},{"name":"prop2","value":"val2"}]},{"hostId":"KubHostSlave2","hostIpAd
dress":"172.17.8.102","hostname":"slave2.dev.kubernetes.example.org","property":[{"nam
e":"prop1","value":"val1"},{"name":"prop2","value":"val2"}]}],"kubernetesMaster":{"hos
tId":"KubHostMaster1","hostIpAddress":"172.17.8.100","hostname":"master.dev.kubernetes
.example.org","property":[{"name":"prop1","value":"val1"},{"name":"prop2","value":"val
2"}]},"portRange":{"lower":4500,"upper":5000},"property":[{"name":"prop1","value":"val
1"},{"name":"prop2","value":"val2"}]}}
```

CLI Command

```
list-kubernetes-groups
```

Response

```
Kubernetes groups found:
+----------+-----------------------------------+
| Group ID | Description                       |
+----------+-----------------------------------+
| KubGrp1  | Kubernetes CoreOS cluster on EC2  |
+----------+-----------------------------------+
```

CLI Command

```
list-kubernetes-hosts KubGrp1
```

Response

```
Kubernetes hosts found:
+---------------+-----------------------------------+--------------+
| Host ID       | Hostname                          | IP Address   |
+---------------+-----------------------------------+--------------+
| KubHostSlave1 | slave1.dev.kubernetes.example.org | 172.17.8.101 |
+---------------+-----------------------------------+--------------+
| KubHostSlave2 | slave2.dev.kubernetes.example.org | 172.17.8.102 |
+---------------+-----------------------------------+--------------+
```

2. Deploy a Docker Cartridge

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d @'php-docker-cartridge.json' -k -u
admin:admin "https://localhost:9443/stratos/admin/cartridge/definition"
```

CLI Command

```
deploy-cartridge -p php-docker-cartridge.json
```

php-docker-cartridge.json

```
{
      "type": "php",
      "provider": "apache",
      "host": "apachestratos.org",
      "displayName": "PHP",
      "description": "PHP Cartridge",
      "version": "5.0",
      "multiTenant": "false",
      "deployerType": "kubernetes",
      "portMapping": [
         {
            "protocol": "http",
            "port": "80",
            "proxyPort": "8280"
         }
       ],
       "container": [
        {
          "imageName": "apachestratos/php:4.1.0-m2",
          "property": [
            {
             "name": "prop-name",
             "value": "prop-value"
            }
          ]
        }
      ]
 }
```

3. Deploy the autoscale policy

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d @'autoscale-policy.json' -k -u
admin:admin “https://localhost:9443/stratos/admin/policy/autoscale”
```

CLI Command

```
deploy-autoscaling-policy -p autoscale-policy.json
```

autoscale-policy.json

```
{
    "id": "economy",
    "loadThresholds": {
      "requestsInFlight": {
         "upperLimit": 80,
         "lowerLimit": 5
      },
      "memoryConsumption": {
         "upperLimit": 80,
         "lowerLimit": 15
      },
      "loadAverage": {
         "upperLimit": 180,
         "lowerLimit": 20
      }
    }
}
```

4. Subscribe to a Docker Cartridge

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d @php-subscription.json -k -u
admin:admin "https://localhost:9443/stratos/admin/cartridge/subscribe"
```

CLI Command

```
subscribe-cartridge --autoscaling-policy economy -p php-subscription.json
```

php-subscription.json

- Replace **payload_parameter.MB_IP** and **payload_parameter.CEP_IP** by your local machine IP in the following json. Also, change the **repoURL** to the GIT repository of your PHP Application (please avoid using **.git** extension when specifying the URL). Add any additional payload parameters that are needed to get PHP cartridge running, such as the repository information etc.

```
{
    "cartridgeType": "php",
    "alias": "myphp",
    "commitsEnabled": "false",
    "autoscalePolicy": "economy",
    "repoURL": "https://github.com/sajhak/myphprepo",
    "property": [
            {
             "name": "KUBERNETES_CLUSTER_ID",
             "value": "KubGrp1"
            },
            {
             "name": "KUBERNETES_REPLICAS_MIN",
             "value": "3"
            },
            {
             "name": "KUBERNETES_REPLICAS_MAX",
             "value": "20"
            },
            {
             "name": "payload_parameter.MB_IP",
             "value": "10.100.5.84"
            },
            {
             "name": "payload_parameter.MB_PORT",
             "value": "1883"
            },
            {
             "name": "payload_parameter.CEP_IP",
             "value": "10.100.5.84"
            },
            {
             "name": "payload_parameter.CEP_PORT",
             "value": "7611"
            },
            {
             "name": "payload_parameter.APP_PATH",
             "value": "/var/www"
            }
          ]
}
```

5. Accessing PHP service

Currently accessing via Load Balancer is not supported. You could access the service via **http://{MemberPrivateIp}:{ALLOCATED_SERVICE_HOST_PORT}**.

In order to find the **MemberPrivateIp** and **ALLOCATED_SERVICE_HOST_PORT** issue following CLI command.

CLI Command

```
list-members --alias myphp --cartridge-type php
```

Response

```
List of members in the [cluster]: myphp

    ServiceName : php
    ClusterId : myphp.php.domain
    Status : Created
    MemberPrivateIp : 172.17.8.100
    MemberFloatingIp : 172.17.8.100
    Member Properties : [Property [name=ALLOCATED_SERVICE_HOST_PORT, value=4506]]
    -----------------------

    ServiceName : php
    ClusterId : myphp.php.domain
    Status : Created
    MemberPrivateIp : 172.17.8.101
    MemberFloatingIp : 172.17.8.101
    Member Properties : [Property [name=ALLOCATED_SERVICE_HOST_PORT, value=4506]]
    -----------------------
================================================
List of LB members for the [cluster]: myphp
================================================
```

Or else you can simply try one of the following URLs:

- http://172.17.8.100:{ALLOCATED_SERVICE_HOST_PORT}
- http://172.17.8.101:{ALLOCATED_SERVICE_HOST_PORT}
- http://172.17.8.102:{ALLOCATED_SERVICE_HOST_PORT}

6. Testing Autoscaling

Currently autoscaling works based on CPU and Memory usage. You can stress docker containers using stress tool. The given docker image alredy have this tool.

- ssh to the coreos node which is having containers (see trouble shoot guide at the end)
- ssh to one of the Docker containers (please refer to questions 9 and 10 of troubleshooting guide)
- stress the container

```
stress -c 4
```

- observe the stratos log, you will get drools logs with respect to scaling.
- check the growing member list.

CLI Command

```
list-members --alias myphp --cartridge-type php
```

Response

```
List of members in the [cluster]: myphp

    ServiceName : php
    ClusterId : myphp.php.domain
    Status : Created
    MemberPrivateIp : 172.17.8.100
    MemberFloatingIp : 172.17.8.100
    Member Properties : [Property [name=ALLOCATED_SERVICE_HOST_PORT, value=4506]]
    -----------------------

    ServiceName : php
    ClusterId : myphp.php.domain
    Status : Created
    MemberPrivateIp : 172.17.8.101
    MemberFloatingIp : 172.17.8.101
    Member Properties : [Property [name=ALLOCATED_SERVICE_HOST_PORT, value=4506]]
    -----------------------
================================================
List of LB members for the [cluster]: myphp
================================================
```

7. Testing Manual Scaling

Currently, manual scaling allows you to change the minimum container count dynamically for an existing subscription.

For an example, say you have subscribed to a **php** cartridge using **myphp** as alias and while you are subscribing, you have specified **KUBERNETES_REPLICAS_MIN** property to **3**. Now, you feel that you need to have **10** minimum docker instances for your php cluster, due to seasonal sales. With manual scaling feature, you are only one command away.

Curl Command

```
curl -X PUT -H "Content-Type: application/json" -d @manual-scaling.json -k -u
admin:admin https://localhost:9443/stratos/admin/subscriptions/myphp/properties
```

CLI Command

```
update-subscription myphp -p manual-scaling.json
```

manual-scaling.json

```
{
    "property": [
            {
             "name": "KUBERNETES_REPLICAS_MIN",
             "value": "10"
            }
          ]
}
```

Response

```
Successfully updated subscription alias: myphp
```

After few minutes, when you list members, you should see at least 10 members in the cluster.

CLI Command

```
list-members --alias myphp --cartridge-type php
```

Similarly, you could downgrade the minimum container count as well, using the same command.

8. Unsubscribe from a Cartridge

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d 'myphp' -k -v -u admin:admin
"https://localhost:9443/stratos/admin/cartridge/unsubscribe"
```

CLI Command

```
unsubscribe-cartridge -f myphp
```

Jira List

Bug

- [STRATOS-641] - LoadBalancer doesn't keep super-tenant subscriptions for a multi-tenant service
- [STRATOS-775] - Error when trying to login as a tenant from Carbon UI
- [STRATOS-802] - Partition deployment fails in EC2
- [STRATOS-814] - Tenant admin permissions are persisted in UM_PERMISSION incorrectly.
- [STRATOS-818] - Error log getting printed during the server start-up
- [STRATOS-858] - Error in publishing artifact updated event
- [STRATOS-882] - [CLI] Invalid Error Handling at Login
- [STRATOS-883] - CLI does not show error messages when unknown server errors occur
- [STRATOS-886] - List Kubernetets Hosts method does to work in REST API

Improvement

- [STRATOS-873] - [Sonar Findings] [Critical] Array is Stored Directly
- [STRATOS-875] - Abstracting out the Application and Group in the Topology
- [STRATOS-884] - CLI has duplicated command implementations in RestCommandLineService class
- [STRATOS-885] - CLI command classes are not using a proper naming convention

New Feature

- [STRATOS-716] - Support for manual scaling for Docker Clusters
- [STRATOS-785] - Autoscaling Containers in Stratos
- [STRATOS-890] - Users should be able to provide a Git repo url when subscribing

Troubleshooting Guide

1. How to ssh to master node?

- Navigate to the cloned repository directory (**SETUP_HOME**)

```
vagrant ssh master
```

2. How to ssh to minion-1 node?

- Navigate to the cloned repository directory (**SETUP_HOME**)

```
vagrant ssh minion-1
```

3. How to ssh to minion-2 node?

- Navigate to the cloned repository directory (**SETUP_HOME**)

```
vagrant ssh minion-2
```

4. How to list all the machines in CoreOS cluster?

- ssh to master node

```
fleetctl list-machines
```

- Output

```
core@master ~ $ fleetctl list-machines
MACHINE     IP      METADATA
07215782... 172.17.8.100    -
4b56425a... 172.17.8.102    -
bf39a4c4... 172.17.8.101    -
```

5. How to list the available replication controllers?

- ssh to master node

```
kubecfg list /replicationControllers
```

- Output

```
core@master ~ $ kubecfg list /replicationControllers
ID                  Image(s)                           Selector         Replicas
test2.php.domain    54.254.64.141:5000/stratos-php     name=php         2
```

6. How to list the available pods?

- ssh to master node

```
kubecfg list /pods
```

- Output

```
core@master ~ $ kubecfg list /pods
ID                                   Image(s)     Host                Labels
Status
115bbe15-49ff-11e4-91b7-08002794b041   stratos-php  172.17.8.100/
name=php,replicationController=php   Waiting
115cc7e9-49ff-11e4-91b7-08002794b041   stratos-php  172.17.8.101/
name=php,replicationController=php   Waiting
```

7. How to tail the kubernetes log?

- ssh to master node

```
journalctl -f
```

8. How to restart the kubernetes scheduler?

- ssh to master node

```
systemctl restart scheduler
```

9. How to list all the docker containers in a node?

- ssh to the node

```
docker ps
```

- Output

```
core@master ~ $ docker ps
CONTAINER ID        IMAGE                         COMMAND                CREATED
STATUS   PORTS
37e3303eb337        stratos-php:latest            "/bin/sh -c '/usr/lo   2 minutes
ago       Up
a3f787d0a7ae        kubernetes/pause:latest       "/pause"               2 minutes
ago       Up       0.0.0.0:80->80/tcp
```

10. How to get the IPAddress of a docker container?

- ssh to the node

```
docker inspect CONTAINER-ID | grep IPAddress
```

- Output

```
core@master ~ $ docker inspect a3f787d0a7ae | grep IPAddress
IPAddress": "10.100.56.3",
```

11. How to ssh to a docker container?

- ssh to the node

```
ssh root@CONTAINER-IPAddress
```

- enter g for password

12. How to kill a docker container?

- ssh to the node

```
docker kill CONTAINER-ID
```

- Output

```
core@minion-1 ~ $ docker kill 6f5ba525f9ab
6f5ba525f9ab
```

- Another conatiner will be created within few seconds

13. How to get info of a specific replicationController as a json?

- ssh to the master node

```
kubecfg -json get /replicationControllers/REPLICATION-CONTROLLER-ID
```

- Output

```
core@master ~ $  kubecfg -json get /replicationControllers/test2.php.domain
{"kind":"ReplicationController","id":"test2.php.domain","creationTimestamp":"2014
-10-02T07:44:58Z","resourceVersion":6701,"apiVersion":"v1beta1","desiredState":{"
replicas":2,"replicaSelector":{"name":"test2.php.domain"},"podTemplate":{"desired
State":{"manifest":{"version":"v1beta1","id":"","volumes":null,"containers":[{"na
me":"test2-apachestratos-org","image":"54.254.64.141:5000/stratos-php","ports":[{
"name":"tcp80","hostPort":80,"containerPort":80,"protocol":"tcp"}],"env":[{"name"
:"SERVICE_NAME","key":"SERVICE_NAME","value":"php"},{"name":"HOST_NAME","key":"HO
ST_NAME","value":"test2.apachestratos.org"},{"name":"MULTITENANT","key":"MULTITEN
ANT","value":"false"},{"name":"TENANT_ID","key":"TENANT_ID","value":"-1234"},{"na
me":"TENANT_RANGE","key":"TENANT_RANGE","value":"-1234"},{"name":"CARTRIDGE_ALIAS
","key":"CARTRIDGE_ALIAS","value":"test2"},{"name":"CLUSTER_ID","key":"CLUSTER_ID
","value":"test2.php.domain"},{"name":"CARTRIDGE_KEY","key":"CARTRIDGE_KEY","valu
e":"uLBSXhS3Kzos5xVe"},{"name":"REPO_URL","key":"REPO_URL","value":"null"},{"name
":"PORTS","key":"PORTS","value":"80"},{"name":"PROVIDER","key":"PROVIDER","value"
:"apache"},{"name":"PUPPET_IP","key":"PUPPET_IP","value":"127.0.0.1"},{"name":"PU
PPET_HOSTNAME","key":"PUPPET_HOSTNAME","value":"puppet.raj.org"},{"name":"PUPPET_
DNS_AVAILABLE","key":"PUPPET_DNS_AVAILABLE","value":"null"},{"name":"PUPPET_ENV",
"key":"PUPPET_ENV","value":"stratos"},{"name":"DEPLOYMENT","key":"DEPLOYMENT","va
lue":"default"},{"name":"CEP_PORT","key":"CEP_PORT","value":"7611"},{"name":"COMM
IT_ENABLED","key":"COMMIT_ENABLED","value":"false"},{"name":"MB_PORT","key":"MB_P
ORT","value":"1883"},{"name":"MB_IP","key":"MB_IP","value":"172.17.42.1"},{"name"
:"CEP_IP","key":"CEP_IP","value":"172.17.42.1"},{"name":"MEMBER_ID","key":"MEMBER
_ID","value":"test2.php.domaindb8e4a24-17e8-40a2-ad85-e0d0ca127887"},{"name":"LB_
CLUSTER_ID","key":"LB_CLUSTER_ID"},{"name":"NETWORK_PARTITION_ID","key":"NETWORK_
PARTITION_ID"},{"name":"KUBERNETES_CLUSTER_ID","key":"KUBERNETES_CLUSTER_ID","val
ue":"KubGrp1"},{"name":"KUBERNETES_MASTER_IP","key":"KUBERNETES_MASTER_IP","value
":"127.0.0.1"},{"name":"KUBERNETES_PORT_RANGE","key":"KUBERNETES_PORT_RANGE","val
ue":"4000-5000"}]}],"restartPolicy":{"always":{}}}},"labels":{"name":"test2.php.d
omain"}}},"currentState":{"replicas":2,"podTemplate":{"desiredState":{"manifest":
{"version":"","id":"","volumes":null,"containers":null,"restartPolicy":{}}}}},"la
bels":{"name":"test2.php.domain"}}
```

14. How to get update a specific replicationController?

- ssh to the master node
- get the replication controller info as a json as above and save it to a file

```
kubecfg -json get /replicationControllers/REPLICATION-CONTROLLER-ID >>
rep-controller.json
```

- edit the rep-controller.json (for example, set replicas to 0)
- update the replicationController

```
kubecfg -c rep-controller.json update
/replicationControllers/REPLICATION-CONTROLLER-ID
```

- Output

```
core@master ~ $ kubecfg -c raj.json update
/replicationControllers/test2.php.domain
ID                  Image(s)                          Selector
Replicas
test2.php.domain    54.254.64.141:5000/stratos-php    name=test2.php.domain   0
```

- it will delete all the pods imediately

15. How to delete a pod?

- ssh to the master node
- get the pod ID using `kubecfg list /pods`

`kubecfg delete pods/POD-ID`

- Output

```
core@master ~ $ kubecfg delete pods/3ed5c5a6-4a0a-11e4-91b7-08002794b041
I1002 08:01:25.351501 02112 request.go:292] Waiting for completion of
/operations/67
Status
success
```

16. How to delete a replicationController?

- ssh to the master node
- get the replicationController ID using `kubecfg list /replicationControllers`

`kubecfg delete replicationControllers/REPLICATION-CONTROLLER-ID`

- Output

```
core@master ~ $ kubecfg delete /replicationControllers/test2.php.domain
I1002 08:32:47.187198 02150 request.go:292] Waiting for completion of
/operations/86
Status
success
```

- If you unsubscribe to the cartridge, replicationControllers, pods and containers for that service cluster will be wiped out