FIWARE Docker Container Service Installation and Administration Guide

Deployment Steps

This section describes the procedure for manually deploying a FIWARE Docker Container Service (FDCS) on OpenStack. In brief, the FDCS is a Multi-Tenant Swarm cluster. We refer to the node where the Multi-Tenant Swarm manager is running as the Swarm Management Node and nodes where the docker engines are running as the Docker Nodes. The following steps are required:

Create a security group for the Swarm Management Node. It containers rules for allowing public access to the Swarm Manager Port, SSH port, and Ping. For example:

service	IP Protocol	From Port	To Port	Source
SSH	TCP	22	22	0.0.0.0/0 (CIDR)
Ping	ICMP	0	0	0.0.0.0/0 (CIDR)
Swarmp Manager	TCP	2376	2376	0.0.0.0/0 (CIDR)

Create a security group for the Docker nodes. It containers rules for allowing public access to the SSH port, Ping, and docker auto assigned ports. The docker auto assigned ports are those ports that docker automatically assigns to containers as there external ports when they are not specifically designated in the docker command. It also containers a rule for exclusive access to the Docker port from the Swarm Management Node. Use the Swam Management Node’s public IP.

service	IP Protocol	From Port	To Port	Source
SSH	TCP	22	22	0.0.0.0/0 (CIDR)
Ping	ICMP	0	0	0.0.0.0/0 (CIDR)
Docker Engine	TCP	2375	2375	Swarm Manager Public IP/32 (CIDR)
Docker Containers auto assigned by docker engine	TCP	32768	61000	0.0.0.0/0 (CIDR)

Create a security group for the NFS server. It contains rules for ssh access and for servicing the Docker Nodes.

service	IP Protocol	From Port	To Port	Source
SSH	TCP	22	22	0.0.0.0/0 (CIDR)
Ping	ICMP	0	0	0.0.0.0/0 (CIDR)
NFS Server	TCP	2049	2049	Docker Nodes IP/32 (CIDR)
NFS Server	UDP	2049	2049	Docker Nodes IP/32 (CIDR)

4.

>apt-get install nfs-kernel-server Create a directory 5.

that will be used to mount the docker volumes on the docker nodes: **>sudo mkdir /docker_volumes** In **/etc/hosts/** allow access to the docker volume directory to the docker nodes. For instance: **/docker_volumes /24(rw,sync,no_subtree_check,no_root_squash)**. Start the nfs server: **>sudo service nfs-kernel-server restart**

>sudo 6.

apt-get install common-nfs Mount the docker volumes directory to be backed by nfs: **>sudo mount -t nfs -o proto=tcp,port=2049 :/docker_volumes /var/lib/docker/volumes**

7.

/etc/default/docker with **DOCKER_OPTS="-H unix:///var/run/docker.sock -H tcp://0.0.0.0:2375 --icc=false"**. And then start docker as a service: **>sudo service docker restart**

8.

- **SWARM_ADMIN_TENANT_ID**: contains the id of the tenant that may run docker commands as admin. Admin is authorized to manage docker resources, including tenant containers, volumes, network, etc., and issue all docker requests without any filtering.
- **SWARM_APIFILTER_FILE**: may point to a json file that describes the docker commands an installation of the service wants to filter out. If no file is pointed to it defaults to apifilter.json in the directory where swarm is started. Currently there is only support for a "disableapi" array which contains a comma separated list of commands to disable. This is an example how an installation could disable network support:

```
{
  "disableapi": [ networkslist, networkinspect, networkconnect, networkdisconnect,
networkcreate, networkdelete ]
}
```

- **SWARM_AUTH_BACKEND:** if set to "Keystone" then Keystone is used to authenticate Tenants that docker requests based on the Authorization Token and Tenant ID in their request header.
- **SWARM_ENFORCE_QUOTA:** if set to "true" then the Multi-Tenant Swarm Quota feature is enabled otherwise it is disabled. See **SWARM_QUOTA_FILE** for how to specify quotas.
- **SWARM_FLAVORS_ENFORCED:** if set to "true" then the Multi-Tenant Swarm Flavors feature is enabled otherwise it is disabled. The flavors specification is embodied in a json file which contains a map describing the valid resource combinations that can appear in create container requests. Currently, Memory is the only resource that can be specified as a flavor. The Memory resource should be specified as a whole number which represents megabytes of memory. See **SWARM_FLAVORS_FILE** for how to specify flavors.
- **SWARM_FLAVORS_FILE:** if SWARM_FLAVORS_ENFORCED is set to "true" then SWARM_FLAVORS_FILE points to a file with the flavors specification. If there is no file pointed to then it defaults to flavors.json in the directory where swarm is started. The specification must contain a "default" flavor. When the create container parameters do not match any of the specified flavors, the default flavor is applied to the create container replacing its original parameters. This is an example of the json flavors specification that is shipped with Multi-Tenant Swarm:

```
{
    "default":{
     "Memory": 64
    },
    "medium":{
     "Memory": 128
    },
    "large":{
     "Memory": 256
    }

}
```

In the above flavors specification example there are three flavors default, medium, and large. *default* describes 64 megabytes of memory. *medium* describes 128 megabytes of memory. *large* describes 256 megabytes of memory. This means that a create container is limited to specifying its memory as 64MB, 128MB, or 256MB. If none is specified then the system will apply the default, i.e. 64MB.

- **SWARM_KEYSTONE_URL:** if SWARM_AUTH_BACKEND is set to "Keystone" then SWARM_KEYSTONE_URL must specify Keystone's URL, e.g. http://cloud.lab.fi-ware.org:4730/v2.0/.
- **SWARM_NETWORK_AUTHORIZATION:** if set to "false" then the Multi-Tenant Swarm Network Authorization feature is disabled otherwise it is enable.
- **SWARM_MEMBERS_TENANT_ID**: contains the tenant id whose members are eligible to use the service. If not set then any valid token tenant id may use the service. SWARM_MEMBERS_TENANT_ID is only valid when SWARM_AUTH_BACKEND is set to Keystone.
- **SWARM_MULTI_TENANT:** if set to "false" then the Multi-Tenant Swarm is disabled otherwise Multi-Tenant Swarm is enabled. When Multi-Tenant Swarm is disabled the result is that the service is launched as vanilla Swarm. Generally disabling Multi-Tenant Swarm is used for debugging purposes to discover if a bug is related to the swarm docker configuration or to a Multi-Tenant Swarm feature.
- **SWARM_QUOTA_FILE:** if SWARM_ENFORCE_QUOTA is set to "true" then SWARM_QUOTA_FILE may specify the quota specification. If there is no file pointed to it defaults to quota.json in the directory where swarm is started. Currently quota support is limited to tenant memory consumption and it is the same for all tenants. This is an example a json quota specification:

```
{
   "Memory": 300
}
```

Start Multi-Tenant Swarm Manager daemon (without TLS) on the Swarm Management Node. The 9. Multi-Tenant Swarm docker image resides in the FIWARE Docker Hub repository at **fiware/swarm_multi_tenant**(https://hub.docker.com/r/fiware/swarm_multi_tenant/) If token discovery is to be used then add the discovery flag, otherwise use the file flag to point to a file with a list of all the Docker Node public ips and docker ports. For instance:

```
   >docker run -t -p 2376:2375 -v /tmp/cluster.ipstmp/cluster.ips -e
SWARM_AUTH_BACKEND=Keystone -e
SWARM_KEYSTONE_URL=http://cloud.lab.fi-ware.org:4730/v2.0/ -t
fiware/swarm_multi_tenant:v0 --debug manage  file:///tmp/cluster.ips
```

Test the cluster’s remote connectivity by pinging and sshing to all the instances (including the 10.
Swarm Management Node).
Test whether the Multi-Tenant Swarm Cluster works as expected by using docker commands on 11.
your local docker client. The docker –H flag specifies the Swarm Manager Node and swarm port.
The docker –config specifies the directory where a config.json file is prepared with a valid token and
a valid tenantid. For instance: **>docker –H tcp://:2376 --config $HOME/dir docker command** See
the FIWARE Docker Container Service Users Guide for more details on how to use the service.

Getting Docker help

- FIWARE Docker Container Service Users Guide
- Docker homepage
- Docker Hub
- Docker blog
- Docker documentation
- Docker Getting Started Guide
- Docker code on GitHub
- Docker mailing list
- Docker on IRC: irc.freenode.net and channel #docker
- Docker on Twitter
- Get Docker help on StackOverflow
- Docker.com