AMS 2016: UniCloud

Julien Chastang

<2015-12-07 Mon>

Contents

Microsoft Azure VM with LDM, TDS, and RAMADDA

Preamble

The following instructions describe how to configure a Microsoft Azure VM serving data with the LDM, TDS, and RAMADDA. This document assumes you have access to Azure resources though these instructions should be fairly similar on other cloud providers. They also assume familiarity with Unix, Docker, and Unidata technology in general. We will be using Docker images defined here:

`https://github.com/Unidata/Unidata-Dockerfiles`

in addition to a configuration specifically planned for AMS 2016 demonstrations here:

`https://github.com/Unidata/Unidata-Dockerfiles/tree/master/ams2016`

Preliminary Setup on Azure (Mostly Docker)

For these instructions, we will decide on the name of an Azure VM; `unidata-server.cloudapp.net` abbreviated to `unidata-server`.

We will now create a VM on Azure.

On your local machine you will want to set up an Azure VM with a `docker-machine` command that will look something like the command below. See here for more information on using `docker-machine` with Azure.

The following command will take a while to run (between 5 and 10 minutes). You will have to supply `azure-subscription-id` and `azure-subscription-cert` path provided by Azure. Again see here if you have questions.

```
docker-machine -D create -d azure \
               --azure-subscription-id="3.141" \
               --azure-subscription-cert="/path/to/mycert.pem" \
               --azure-size="ExtraLarge" unidata-server
```

Set up your environment to interact with your new Azure VM.

```
eval "$(docker-machine env unidata-server)"
```

`ssh` into your new host with `docker-machine`

```
docker-machine ssh unidata-server
```

You will need to install one or more Unix utilities:

```
sudo apt-get -qq update
sudo apt-get -qq install unzip
```

Add the `ubuntu` user to the `docker` group.

```
sudo usermod -G docker ubuntu
sudo service docker restart
```

At this point, we want to restart the VM to get a fresh start. This may take a little while. . . .

```
docker-machine restart unidata-server
eval "$(docker-machine env unidata-server)"

docker-machine ssh unidata-server
```

Next install `docker-compose`. You may have to update version (currently at `1.5.2`).

```
 curl -L \
https://github.com/docker/compose/releases/download/1.5.2/docker-compose-‘uname -s‘-‘un
      > docker-compose
 sudo mv docker-compose /usr/local/bin/
 sudo chmod +x /usr/local/bin/docker-compose
```

LDM and TDS Configuration

Clone `Unidata-Dockerfiles` and `TdsConfig` repositories:

```
mkdir -p /home/ubuntu/git
git clone https://github.com/Unidata/Unidata-Dockerfiles /home/ubuntu/git/Unidata-Docke
git clone https://github.com/Unidata/TdsConfig /home/ubuntu/git/TdsConfig
```

Create some directories for the LDM, basically the familiar `etc`, `var`, `var/log`.

```
mkdir -p ~/var/logs
mkdir -p ~/etc/TDS
```

Now copy all files in `~/git/Unidata-Dockerfiles/ldm/etc/` into the `~/etc` directory from the `Unidata-Dockerfiles` repositories. Note that some of these files will be modified or overwritten shortly.

- `ldmd.conf`
- `registry.xml`
- `pqact.conf`
- `scour.conf`
- `netcheck.conf`

```
cp ~/git/Unidata-Dockerfiles/ldm/etc/* ~/etc
```

Now we are going to replace `ldmd.conf`, `registry.xml`, `scour.conf` from the `~/git/Unidata-Dockerfiles/ams2016` dir into `~/etc`.

Ask for Unidata (or Someone to Feed You Data Via the LDM)

The LDM operates on a push data model. You will have to find someone who will agree to push you the data. If you are part of the American academic community please send a support email to support-idd@unidata.ucar.edu.

ldmd.conf

```
cp ~/git/Unidata-Dockerfiles/ams2016/ldmd.conf ~/etc/
```

This `ldmd.conf` has been setup for the AMS 2016 demonstration serving the following data feeds:

- 13km Rapid Refresh

In addition, there is a `~/git/TdConfig/idd/pqacts/README.txt` file that may be helpful in writing a suitable `ldmd.conf` file.

registry.xml

```
cp ~/git/Unidata-Dockerfiles/ams2016/registry.xml ~/etc/
```

Make sure the `registry.xml` is edited correctly. The important element in this file is the `hostname` element. Work with support-idd@unidata.ucar.ed so that LDM stats get properly reported. It should be something like `unidata-server.azure.unidata.ucar.edu`.

scour.conf

Scouring configuration for the LDM. The crontab entry that runs scour is in the LDM Docker container. scour is invoked once per day.

```
cp ~/git/Unidata-Dockerfiles/ams2016/scour.conf ~/etc/
```

`pqact.conf` and TDS configuration

Next, explode `~/git/TdsConfig/idd/config.zip` into `~/tdsconfig` and `cp -r` the `pqacts` directory into `~/etc/TDS`. **Note** do NOT use soft links. Docker does not like them.

```
mkdir -p ~/tdsconfig/
cp ~/git/TdsConfig/idd/config.zip ~/tdsconfig/
unzip ~/tdsconfig/config.zip -d ~/tdsconfig/
cp -r ~/tdsconfig/pqacts/* ~/etc/TDS
```

netcheck.conf

This files remain unchanged.

Edit TDS catalog.xml Files

The `catalog.xml` files for TDS configuration are contained with the `~/Tdsconfig` directory. Search for all files terminating in `.xml` in that directory. Edit the xml files for what data you wish to server. See the TDS Documentation for more information on editing these XML files.

Let's see what is available in the `~/tdsconfig` directory.

```
find ~/tdsconfig -type f -name "*.xml"
```

```
/home/ubuntu/tdsconfig/idd/forecastModels.xml
/home/ubuntu/tdsconfig/idd/radars.xml
/home/ubuntu/tdsconfig/idd/obsData.xml
/home/ubuntu/tdsconfig/idd/forecastProdsAndAna.xml
/home/ubuntu/tdsconfig/idd/satellite.xml
/home/ubuntu/tdsconfig/radar/CS039_L2_stations.xml
/home/ubuntu/tdsconfig/radar/CS039_stations.xml
/home/ubuntu/tdsconfig/radar/RadarNexradStations.xml
/home/ubuntu/tdsconfig/radar/RadarTerminalStations.xml
/home/ubuntu/tdsconfig/radar/RadarL2Stations.xml
/home/ubuntu/tdsconfig/radar/radarCollections.xml
/home/ubuntu/tdsconfig/catalog.xml
/home/ubuntu/tdsconfig/threddsConfig.xml
/home/ubuntu/tdsconfig/wmsConfig.xml
```

Setting up the Data Volumes for the LDM and RAMADDA

The `/mnt` volume on Azure is a good place to store data. I do not know what kind of assurances Azure makes about the reliability of storing your data there for the long term. (I remember reading about this once, but I cannot remember where.) For the LDM this should not be too much of a problem, but for RAMADDA you may wish to be careful.

```
df -H
```

```
Filesystem  Size  Used  Avail  Use%  Mounted                            on
/dev/sda1   31G   1.8G  28G    6%    /
none        4.1k     0  4.1k   0%    /sys/fs/cgroup
udev        7.4G   13k  7.4G   1%    /dev
tmpfs       1.5G  394k  1.5G   1%    /run
none        5.3M     0  5.3M   0%    /run/lock
none        7.4G     0  7.4G   0%    /run/shm
none        105M     0  105M   0%    /run/user
none        66k      0  66k    0%    /etc/network/interfaces.dynamic.d
/dev/sdb1   640G   73M  607G   1%    /mnt
```

Create a `/data` directory where the LDM and RAMADDA data will live.

For the LDM:

```
sudo ln -s /mnt /data
sudo mkdir /mnt/ldm/
sudo chown -R ubuntu:docker /data/ldm
```

Create a `/repository` directory where the RAMADDA data will live.

```
sudo mkdir /mnt/repository/
sudo chown -R ubuntu:docker /data/repository
```

RAMADDA Configuration

When you fire up RAMADDA for the very first time, your will have to have a `password.properties` file in the RAMADDA home directory which is `/data/repository/`. See RAMADDA documentation for more details on setting up RAMADDA.

```
echo ramadda.install.password=changeme! > /data/repository/pw.properties
```

Ports

Make sure these ports are open on the VM where you are doing this work. Ask the cloud administrator for these ports to be open.

Service	External Port
HTTP	80
TDS	8080
RAMADDA	8081
SSL TDM	8443
LDM	388

Tomcat Logging for TDS and RAMADDA

It is a good idea to mount Tomcat logging directories outside the container so that they can be managed for both the TDS and RAMADDA.

```
mkdir -p ~/logs/ramadda-tomcat
mkdir -p ~/logs/tds-tomcat
```

Note that there is also a logging directory in `~/tdsconfig/logs`. That should be looked at periodically.

Fire Up the LDM TDS RAMADDA TDM

Edit the `docker-compose.yml` file and change the `TDM_PW` to `MeIndexer`.

At this point you are almost ready to run the whole kit and caboodle. But first pull the relevant docker images to make life easier for the subequent `docker-compose` command.

```
docker pull unidata/ldmtds:latest
docker pull unidata/tdm:latest
docker pull unidata/tds:latest
docker pull unidata/ramadda:latest
```

At this point you can run

```
docker-compose -f ~/git/Unidata-Dockerfiles/ams2016/docker-compose.yml up -d
```

Check What You Have Setup

At this point you have these services running:

- LDM
- TDS
- TDM
- RAMADDA

Verify you have the TDS running by navigating to:
`http://unidata-server.cloudapp.net/thredds/catalog.html`
Verify you have the RAMADDA running by navigating to:
`http://unidata-server.cloudapp.net:8081/repository`

If you are going to RAMADDA for the first time, you will have to do some RAMADDA set up.

Also RAMADDA has access to the `/data/ldm` directory so you may wish to set up server-side view of this part of the file system.