

UNICLOUD

Docker Use at Unidata

Julien Chastang (UCAR, Boulder, CO USA) , Ward Fisher, Sean Arms

2016-01-07

Contents

1	Introduction	1
1.1	Demonstration Servers	1
2	Quick Start	1
3	Long Form Instructions	2
4	Preliminary Setup on Azure	2
4.1	Install <code>docker-machine</code>	2
4.2	Create a VM on Azure	2
4.3	Configure Unix Shell to Interact with New Azure VM	3
4.4	Restart Azure VM	3
4.5	ssh into VM with <code>docker-machine</code>	3
4.6	Install Packages with <code>apt-get</code>	3
4.7	Add <code>ubuntu</code> User to <code>docker</code> Group and Restart Docker	3
4.8	Install <code>docker-compose</code> on VM	4
5	LDM and TDS Configuration	4
5.1	Background	4
5.1.1	Unidata-Dockerfiles Repository	4
5.1.2	TDSConfig Repository	4
5.2	<code>git clone</code> Repositories	4
5.3	Configuring the LDM	5
5.3.1	LDM Directories on Docker Host	5
5.3.2	LDM Configuration Files	5
5.3.3	Upstream Data Feed from Unidata or Elsewhere	7
5.4	Configuring the TDS	7
5.4.1	Edit TDS <code>catalog.xml</code> Files	7
6	Setting up Data Volumes	7
6.1	Check Free Disk Space	7
6.2	Create <code>/data</code> Directory	8
7	Opening Ports	8
7.1	More About the TDM	8
7.2	Note about port ADDE 112	8
8	Tomcat Logging for TDS and RAMADDA	9
9	Starting the LDM TDS RAMADDA TDM	9

9.0.1	RAMADDA Preconfiguration	9
9.0.2	Final Edit to <code>docker-compose.yml</code>	9
9.0.3	<code>chown</code> for Good Measure	9
9.0.4	Pull Down Images from the DockerHub Registry	9
9.0.5	Start the LDM, TDS, TDM, RAMADDA	10
10	Check What is Running	10
10.1	Docker Process Status	10
10.2	Checking Data Directory	10
10.3	TDS and RAMADDA URLs	11
10.4	Viewing Data with the IDV	11
10.4.1	Access TDS with the IDV	11
10.4.2	Access RAMADDA with the IDV	11
11	Appendix	12
11.1	Common Problems	12
11.1.1	Certificate Regeneration	12
11.1.2	Size of VM is not Large Enough	12
11.1.3	Where is my Data and the Finicky TDM	13
11.1.4	Cannot connect to the Docker daemon	13
11.2	Asking for Help	13
11.3	Acknowledgments	13

1 Introduction

This guide is a companion document (available in HTML, Markdown, text, PDF)¹ to a 2016 American Meteorological Society oral presentation, *UniCloud: Docker Use at Unidata*². It describes how to configure the LDM³, TDS⁴, and RAMADDA⁵ on a Microsoft Azure VM⁶. It assumes you have access to Azure resources though these instructions should be fairly similar on other cloud providers (e.g., Amazon). These instructions also require familiarity with Unix, Docker, and Unidata technology in general. You must have `sudo` privileges on the Azure host which will hopefully be available you. You must be comfortable entering commands at the Unix command line. We will be using Docker images defined at the `Unidata-Dockerfiles` repository⁷ in addition to a configuration specifically planned for an AMS 2016 demonstrations project⁸.

1.1 Demonstration Servers

An example of having followed these instructions can be found with these TDS⁹ and RAMADDA¹⁰ demonstration servers running on the Azure cloud.

2 Quick Start

In order to best understand this configuration process, it is recommended to read the complete contents of this document and follow the instructions starting in the next section. If there are problems you will be able

¹<https://github.com/Unidata/Unidata-Dockerfiles/tree/master/ams2016/README>

²<https://ams.confex.com/ams/96Annual/webprogram/Paper287336.html>

³<http://www.unidata.ucar.edu/software/ldm/>

⁴<http://www.unidata.ucar.edu/software/thredds/current/tds/>

⁵<http://sourceforge.net/projects/ramadda/>

⁶<https://azure.microsoft.com>

⁷<https://github.com/Unidata/Unidata-Dockerfiles>

⁸<https://github.com/Unidata/Unidata-Dockerfiles/tree/master/ams2016>

⁹<http://unidata-server.cloudapp.net/thredds/catalog.html>

¹⁰<http://unidata-server.cloudapp.net:8081/repository>

to reason about the errors. However, if you are eager to get started, you can follow this quick start section.

- `git clone https://github.com/Unidata/Unidata-Dockerfiles`
- Download and install¹¹ `docker-machine`
- Run the `Unidata-Dockerfiles/ams2016/unicloud-1.sh` script (this will take few minutes) to **create the Docker host on Azure**.

For example,

```
unicloud-1.sh \
  --azure-host <azure-host> \
  --azure-subscription-id "3.14" \
  --azure-subscription-cert \
  "/path/to/mycert.pem"
```

Now you are ready to do additional configuration on the new Docker host:

```
docker-machine \
  ssh <azure-host> "bash -s" < \
  Unidata-Dockerfiles/ams2016/unicloud-2.sh
```

Finally,

- `ssh` into new Docker host with `docker-machine ssh <azure-host>`
- Edit `registry.xml` with the correct hostname element
- Edit `~/git/Unidata-Dockerfiles/ams2016/docker-compose.yml` with the correct `TDM_PW` and `TDS_HOST`.
- Run `~/git/Unidata-Dockerfiles/ams2016/unicloud-3.sh`
- **Check** your setup

3 Long Form Instructions

If you are opting for the long form instructions instead of the quick start, begin here.

4 Preliminary Setup on Azure

The VM we are about to create will be our **Docker Host** from where we will run Docker containers for the LDM, TDS, and RAMADDA.

4.1 Install `docker-machine`

Install `docker-machine`¹² on your local computer. `docker-machine` is a command line tool that gives users the ability to create Docker VMs on your local computer or on a cloud provider such as Azure.

¹¹<https://docs.docker.com/machine/install-machine/>

¹²<https://docs.docker.com/machine/install-machine/>

4.2 Create a VM on Azure

The following `docker-machine` command will create a Docker VM on Azure for running various Unidata Docker containers. **Replace the environment variables with your choices.** This command will take a few minutes to run (between 5 and 10 minutes). You will have to supply `azure-subscription-id` and `azure-subscription-cert` path. See the Azure `docker-machine` instructions¹³, if you have questions about this process. Also set the size of the VM¹⁴ (e.g., `Small`, `ExtraLarge`) and supply the name of the Azure Docker host.

```
# Create Azure VM via docker-machine
docker-machine \
  -D create \
  -d azure \
  --azure-subscription-id=$AZURE_ID \
  --azure-subscription-cert=$AZURE_CERT \
  --azure-size=$AZURE_SIZE $AZURE_HOST
```

4.3 Configure Unix Shell to Interact with New Azure VM

Execute the following `eval` command on your local computer shell environment to ensure `docker` commands will be run with the newly created Docker host.

```
# Ensure docker commands will be run with new host
eval "$(docker-machine env $AZURE_HOST)"
```

If you see an error message pertaining to `docker-machine regenerate-certs`, see the [Certificate Regeneration section of the Appendix](#).

4.4 Restart Azure VM

Mysteriously, when you `ssh` (see next section) into the fresh VM, you are immediately told to restart it so let's preempt that message by doing that now.

```
# immediately restart VM, according to Azure
docker-machine restart $AZURE_HOST
# Again, ensure docker commands will be run with new host
eval "$(docker-machine env $AZURE_HOST)"
```

4.5 ssh into VM with docker-machine

```
docker-machine ssh $AZURE_HOST
```

4.6 Install Packages with apt-get

At the very least, we will need `unzip` on the Azure Docker host. The Unix `tree` command can also be handy. `docker` is already installed on Azure by default.

```
# update and install packages
sudo apt-get -qq update
sudo apt-get -qq upgrade
sudo apt-get -qq install unzip tree
```

¹³<https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-docker-machine/>

¹⁴<https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-size-specs/>

4.7 Add ubuntu User to docker Group and Restart Docker

```
# Add ubuntu to docker group
sudo usermod -G docker ubuntu
```

```
# Restart docker service
sudo service docker restart
```

In Unix, when adding a user to a group, it is simply easiest to logout and log back in for this change to be recognized. Do that by exiting the VM and logging back in with `docker-machine ssh` command.

4.8 Install docker-compose on VM

`docker-compose` is a tool for defining and running multi-container Docker applications. In our case, we will be running the LDM, TDS, TDM (THREDDS Data Manager) and RAMADDA so `docker-compose` is perfect for this scenario. Install `docker-compose` on the Azure Docker host.

You may have to update version (currently at 1.5.2).

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

5 LDM and TDS Configuration

5.1 Background

We have done the preliminary legwork to tackle the next step in this process. We will now want to clone two repositories that will allow us to configure and start running the the LDM, TDS, and RAMADDA. In particular, we will be cloning:

- github.com/Unidata/Unidata-Dockerfiles¹⁵
- github.com/Unidata/TdsConfig¹⁶

5.1.1 Unidata-Dockerfiles Repository

The `Unidata-Dockerfiles` repository contains a number of Dockerfiles that pertain to various Unidata technologies (e.g., the LDM) and also projects (e.g., `ams2016`). As a matter of background information, a `Dockerfile` is a text file that contains commands to build a Docker image containing, for example, a working LDM. These Docker images can subsequently be run by `docker` command line tools, or `docker-compose` commands that rely on a `docker-compose.yml` configuration file. A `docker-compose.yml` file is a text file that captures exactly how one or more containers run including directory mappings (from outside to within the container), port mappings (from outside to within the container), and other information.

5.1.2 TDSConfig Repository

The `TDSConfig` repository is a project that captures THREDDS and LDM configuration files (e.g., `catalog.xml`, `pqact.conf`) for the TDS at <http://thredds.ucar.edu>. Specifically, these TDS and LDM configurations were meant to work in harmony with one another. We can re-use this configuration with some minor adjustments for running the TDS on the Azure cloud.

¹⁵<https://github.com/Unidata/Unidata-Dockerfiles>

¹⁶<https://github.com/Unidata/TdsConfig>

5.2 git clone Repositories

With that background information out of the way, let's clone those repositories by creating `~/git` directory where our repositories will live and issuing some git commands.

```
# Get the git repositories we will want to work with
mkdir -p /home/ubuntu/git
git clone https://github.com/Unidata/Unidata-Dockerfiles \
/home/ubuntu/git/Unidata-Dockerfiles
git clone https://github.com/Unidata/TdsConfig /home/ubuntu/git/TdsConfig
```

5.3 Configuring the LDM

5.3.1 LDM Directories on Docker Host

For anyone who has worked with the LDM, you may be familiar with the following directories:

- `etc/`
- `var/data`
- `var/logs`
- `var/queue`

The LDM `etc` directory is where you will find configuration files related to the LDM including `ldmd.conf`, `pqact` files, `registry.xml`, and `scour.conf`. We will need the ability to easily observe and manipulate the files from **outside** the running LDM container. To that end, we need to find a home for `etc` on the Docker host. The same is true for the `var/data` and `var/logs` directories. Later, we will use Docker commands that have been written on your behalf to mount these directories from **outside** to **within** the container. The `var/queues` directory will remain inside the container.

```
# Create LDM directories
mkdir -p ~/var/logs
mkdir -p ~/etc/TDS
```

`var/data` is a bit different in that it needs to be mounted on data volume on the Docker host. We will be handling that step further on.

5.3.2 LDM Configuration Files

There is a generic set of LDM configuration files located here `~/git/Unidata-Dockerfiles/ldm/etc/`. However, we will just grab `netcheck.conf` which will remain unmodified.

```
# Copy various files for the LDM.
cp ~/git/Unidata-Dockerfiles/ldm/etc/netcheck.conf ~/etc
```

The rest of the LDM configuration files will come from our `ams2016` project directory. Also, remember that these files will be used **inside** the LDM container that we will set up shortly. We will now be working with these files:

- `ldmd.conf`
 - `registry.xml`
 - `scour.conf`
1. `ldmd.conf`

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

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

- 13km Rapid Refresh¹⁷
- GFS One Degree¹⁸
- NESDIS GOES Satellite Data¹⁹
- Unidata NEXRAD Composites

For your information, and for future reference, there is a `~/git/TdConfig/idd/pqacts/README.txt` file that may be helpful in writing a suitable `lmdm.conf` file.

2. registry.xml

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

This file has been set up for the AMS 2016 demonstration. Otherwise you will have to edit the `registry.xml` to ensure the `hostname` element is correct. For your own cloud VMs, and if you are part of the American academic community, work with `support-idd@unidata.ucar.edu` to devise a correct `hostname` element so that LDM statistics get properly reported. Here is an example `hostname` element:

```
unidata-server.azure.unidata.ucar.edu
```

3. scour.conf

You need to scour data or else your disk will full up. The crontab entry that runs scour is in the LDM Docker container²⁰. Scouring is invoked once per day.

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

4. pqact.conf and TDS configuration

In the `lmdm.conf` file we copied just a moment ago, there is a reference to a `pqact` file; `etc/TDS/pqact.forecastModels`. We need to ensure that file exists by doing the following instructions. Specifically, 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.

```
# Set up LDM and TDS configuration
mkdir -p ~/tdsconfig/
cp ~/git/TdsConfig/idd/config.zip ~/tdsconfig/
unzip ~/tdsconfig/config.zip -d ~/tdsconfig/
cp -r ~/tdsconfig/pqacts/* ~/etc/TDS
```

5. Edit ldmfile.sh

Examine the `etc/TDS/util/ldmfile.sh` file. As the top of this file indicates, you must change the `logfile` to suit your needs. Change the

```
logfile=logs/ldm-mcidas.log
```

¹⁷<http://rapidrefresh.noaa.gov/>

¹⁸<http://www.nco.ncep.noaa.gov/pmb/products/gfs/>

¹⁹http://www.nesdis.noaa.gov/imagery_data.html

²⁰<https://github.com/Unidata/Unidata-Dockerfiles/blob/master/ldm/crontab>

line to

```
logfile=var/logs/ldm-mcidas.log
```

This will ensure `ldmfile.sh` can properly invoked from the `pqact` files.

We can achieve this change with a bit of `sed`:

```
# in place change of logs dir w/ sed
sed -i s/logs\\ldm-mcidas.log/var\\logs\\ldm-mcidas\\.log/g \
    ~/etc/TDS/util/ldmfile.sh
```

5.3.3 Upstream Data Feed from Unidata or Elsewhere

The LDM operates on a push data model. You will have to find an institution 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` to discuss your LDM data requirements.

5.4 Configuring the TDS

5.4.1 Edit TDS `catalog.xml` Files

The `catalog.xml` files for TDS configuration are contained within 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 Documentation²¹ for more information on editing these XML files.

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

```
find ~/tdsconfig -name *.xml | awk 'BEGIN { FS = "/" } ; { print $NF }' | head
```

```
forecastModels.xml
radars.xml
obsData.xml
forecastProdsAndAna.xml
satellite.xml
CS039_L2_stations.xml
CS039_stations.xml
RadarNexradStations.xml
RadarTerminalStations.xml
RadarL2Stations.xml
```

For the purposes of the AMS demonstration, let's extract some `catalog.xml` files that are consistent with the rest of this configuration:

```
# use catalog xml files that are consistent with our data holdings
cp -r ~/git/Unidata-Dockerfiles/ams2016/catalogs/* ~/tdsconfig
```

6 Setting up Data Volumes

As alluded to earlier, we will have to set up data volumes so that the LDM can write data, and the TDS and RAMADDA can have access to that data. The `/mnt` has lots of space, but the storage is considered **ephemeral** so be careful! Azure makes no effort to backup data on `/mnt`. For the LDM this should not be too much of a problem because real time data is coming in and getting scoured continuously, but for for any other application you may wish to be careful as there is the potential to lose data. There is more information about this topic here²².

²¹<http://www.unidata.ucar.edu/software/thredds/current/tds/catalog/index.html>

²²<https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-how-to-attach-disk/>

6.1 Check Free Disk Space

Let's first display the free disk space with the `df` command.

```
df -H
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	7.4G	0	7.4G	0%	/dev
tmpfs	1.5G	27M	1.5G	2%	/run
/dev/sda1	31G	6.6G	24G	22%	/
tmpfs	7.4G	463k	7.4G	1%	/dev/shm
tmpfs	5.3M	0	5.3M	0%	/run/lock
tmpfs	7.4G	0	7.4G	0%	/sys/fs/cgroup
none	66k	0	66k	0%	/etc/network/interfaces.dynamic.d
/dev/sdb1	640G	187G	421G	31%	/mnt
cgmfs	103k	0	103k	0%	/run/cgmanager/fs
tmpfs	1.5G	0	1.5G	0%	/run/user/1000

6.2 Create /data Directory

Create a `/data` directory where the LDM can write data soft link to the `/mnt` directory. Also, create a `/repository` directory where RAMADDA data will reside.

```
# Set up data directories
sudo ln -s /mnt /data
sudo mkdir /mnt/l dm/
sudo chown -R ubuntu:docker /data/l dm
sudo mkdir /home/ubuntu/repository/
sudo chown -R ubuntu:docker /home/ubuntu/repository
```

These directories will be used by the LDM, TDS, and RAMADDA docker containers when we mount directories from the Docker host into these containers.

7 Opening Ports

Ensure these ports are open²³ on the VM where these containers will run.

Service	External Port
HTTP	80
TDS	8080
RAMADDA	8081
SSL TDM	8443
LDM	388
ADDE	112

7.1 More About the TDM

Note the TDM is an application that works in conjunction with the TDS. It creates indexes for GRIB data in the background, and notifies the TDS via port 8443 when data have been updated or changed. See here²⁴ to learn more about the TDM.

²³<https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-set-up-endpoints/>

²⁴<https://www.unidata.ucar.edu/software/thredds/current/tds/reference/collections/TDM.html>

7.2 Note about port ADDE 112

The ADDE port 112 is for future use since we have not dockerized ADDE, yet.

8 Tomcat Logging for TDS and RAMADDA

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

```
# Create Tomcat logging directories
mkdir -p ~/logs/ramadda-tomcat
mkdir -p ~/logs/tds-tomcat
```

Note there is also a logging directory in `~/tdsconfig/logs`. All these logging directories should be looked at periodically, not the least to ensure that `log` files are not filling up your system.

9 Starting the LDM TDS RAMADDA TDM

9.0.1 RAMADDA Preconfiguration

When you start RAMADDA for the very first time, you must have a `password.properties` file in the RAMADDA home directory which is `/home/ubuntu/repository/`. See RAMADDA documentation²⁵ for more details on setting up RAMADDA. Here is a `pw.properties` file to get you going. Change password below to something more secure!

```
# Create RAMADDA default password
echo ramadda.install.password=changeme! > \
  /home/ubuntu/repository/pw.properties
```

9.0.2 Final Edit to `docker-compose.yml`

When the TDM communicates to the TDS concerning changes in data it observes with data supplied by the LDM, it will communicate via the `tdm tomcat` user. Edit the `docker-compose.yml` file and change the `TDM_PW` to `MeIndexer`. This is not as insecure as it would seem since the `tdm` user has few privileges. Optimally, one could change the password hash for the TDM user in the `tomcat-users.xml` file. Also endure `TDS_HOST` is pointing to the correct Azure Docker host (e.g., `http://unidata-server.cloudapp.net`).

9.0.3 `chown` for Good Measure

As we are approaching completion, let's ensure all files in `/home/ubuntu` are owned by the `ubuntu` user in the `docker` group.

```
sudo chown -R ubuntu:docker /home/ubuntu
```

9.0.4 Pull Down Images from the DockerHub Registry

You are almost ready to run the whole kit and caboodle. But first pull the relevant docker images to make this easier for the subsequent `docker-compose` command.

```
# Docker pull all relevant images
docker pull unidata/ldmtds:latest
docker pull unidata/tdm:latest
docker pull unidata/tds:latest
docker pull unidata/ramadda:latest
```

²⁵<http://ramadda.org/repository/userguide/toc.html>

9.0.5 Start the LDM, TDS, TDM, RAMADDA

We are now finally ready to start the LDM, TDS, TDM, RAMADDA with the following `docker-compose` command.

```
# Start up all images
docker-compose -f ~/git/Unidata-Dockerfiles/ams2016/docker-compose.yml up -d
```

10 Check What is Running

In this section, we will assume you have created a VM called `unidata-server`. You should have these services running:

- LDM
- TDS
- TDM
- RAMADDA

Next, we will check our work through various means.

10.1 Docker Process Status

From the shell where you started `docker-machine` earlier you can execute the following `docker ps` command to list the containers on your docker host. It should look something like the output below.

```
docker ps --format "table {{.ID}}\t{{.Image}}\t{{.Status}}"
```

CONTAINER ID	IMAGE	STATUS			
d192f310d3a6	unidata/mcidas	Up	10	days	
7615f31c5bf8	unidata/ramadda:latest	Up	13	days	
9dbd4e83f15b	unidata/ldmtds:latest	Up	13	days	
dbcd457abe75	unidata/tdm:latest	Up	13	days	
5a4a13b5671e	unidata/tds:latest	Up	12	days	

10.2 Checking Data Directory

If you used the configuration described herein, you will have a `/data/ldm` directory tree that looks something like this created by the LDM:

```
tree --charset=ASCII -L 3 /data/ldm -d -I '*2015*|*2016*|current'
```

```
/data/ldm
'-- pub
  '-- native
    |-- grid
    |-- radar
    '-- satellite
```

5 directories

Poke around for GRIB2 data.

```
find /data/ldm -name *.grib2 | awk 'BEGIN { FS = "/" } ; { print $NF }' | head
```

```
RR_CONUS_13km_20151216_2200.grib2
RR_CONUS_13km_20151216_2100.grib2
RR_CONUS_13km_20151216_2000.grib2
RR_CONUS_13km_20151216_2300.grib2
GFS_Global_onedeg_20151216_1800.grib2
Level3_Composite_NOR_20151217_0000.grib2
Level3_Composite_NOR_20151217_0005.grib2
Level3_Composite_NOR_20151217_0010.grib2
Level3_Composite_NOR_20151216_2155.grib2
Level3_Composite_NOR_20151216_2315.grib2
```

Search for GRIB index files (grib2). If you do not see them, see the section about a **finicky TDM** in the in the Appendix.

```
find /data/ldm -name *.grib2 | awk 'BEGIN { FS = "/" } ; { print $NF }' | head
```

```
RR_CONUS_13km_20151216_2200.grib2.grib2
RR_CONUS_13km_20151216_2300.grib2.grib2
RR_CONUS_13km_20151216_2100.grib2.grib2
RR_CONUS_13km_20151216_2000.grib2.grib2
GFS_Global_onedeg_20151216_1800.grib2.grib2
Level3_Composite_NOR_20151217_0005.grib2.grib2
Level3_Composite_NOR_20151217_0000.grib2.grib2
Level3_Composite_NOR_20151216_2205.grib2.grib2
Level3_Composite_NOR_20151216_2315.grib2.grib2
Level3_Composite_NOR_20151216_2330.grib2.grib2
```

10.3 TDS and RAMADDA URLs

Verify what you have the TDS and RAMADDA running by, for example, navigating to: <http://unidata-server.cloudapp.net/thredds/catalog.html> and <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 up²⁶.

10.4 Viewing Data with the IDV

Another way to verify your work is run the Unidata Integrated Data Viewer²⁷.

10.4.1 Access TDS with the IDV

In the IDV Dashboard²⁸, you should be able to enter the catalog XML URL: <http://unidata-server.cloudapp.net/thredds/catalog.xml>.

10.4.2 Access RAMADDA with the IDV

RAMADDA has good integration with the IDV and the two technologies work well together. You may wish to install the RAMADDA IDV plugin²⁹ to publish IDV bundles to RAMADDA. RAMADDA also has access to the /data/ldm directory so you may want to set up server-side view of this part of the file system³⁰. Finally, you can enter this catalog URL in the IDV dashboard to examine data holdings shared bundles, etc. on RAMADDA <http://unidata-server.cloudapp.net:8081/repository?output=thredds.catalog>.

²⁶<http://ramadda.org/repository/userguide/toc.html>

²⁷<https://www.unidata.ucar.edu/software/idv/>

²⁸<https://www.unidata.ucar.edu/software/idv/docs/userguide/data/choosers/CatalogChooser.html>

²⁹<http://www.unidata.ucar.edu/software/idv/docs/workshop/savingstate/Ramadda.html>

³⁰<http://ramadda.org/repository/userguide/developer/filesystem.html>

1. RAMADDA IDV Plugin

You may wish to install the RAMADDA IDV plugin³¹ to publish IDV bundles to RAMADDA from directly within the IDV.

2. RAMADDA Server Side Views

RAMADDA also has access to the `/data/ldm` directory so you may want to set up server side view of this part of the file system³². It is a two step process where administrators go to the Admin, Access, File Access menu item and lists the allowed directories they potentially wish to expose via RAMADDA. Second, the users are now capable of creating a "Server Side" Files with the usual RAMADDA entry creation mechanisms.

3. RAMADDA Catalog Views from the IDV

Finally, you can enter this catalog URL in the IDV dashboard to examine data holdings shared bundles, etc. on RAMADDA. For example, <http://unidata-server.cloudapp.net:8081/repository?output=thredds.catalog>.

11 Appendix

11.1 Common Problems

11.1.1 Certificate Regeneration

When using `docker-machine` may see an error message pertaining to regenerating certificates.

```
Error running connection boilerplate: Error checking and/or regenerating the
certs: There was an error validating certificates for host
"host.cloudapp.net:2376": dial tcp 104.40.58.160:2376: i/o timeout You can
attempt to regenerate them using 'docker-machine regenerate-certs name'. Be
advised that this will trigger a Docker daemon restart which will stop running
containers.
```

In this case:

```
docker-machine regenerate-certs <azure-host>
eval "$(docker-machine env <azure-host>)"
```

Like the error message says, you may need to restart your Docker containers with `docker-compose`, for example.

11.1.2 Size of VM is not Large Enough

If you see your containers not starting on Azure or error messages like this:

```
ERROR: Cannot start container
ef229d1753b24b484687ac4d6b8a9f3b961f2981057c59266c45b9d548df4e24: [8] System
error: fork/exec /proc/self/exe: cannot allocate memory
```

it is possible you did not create a sufficiently large VM. Try increasing the size of the VM³³.

³¹<http://www.unidata.ucar.edu/software/idv/docs/workshop/savingstate/Ramadda.html>

³²<http://ramadda.org//repository/userguide/developer/filesystem.html>

³³<https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-size-specs/>

11.1.3 Where is my Data and the Finicky TDM

If you are not finding the data you expect to see via the THREDDS `catalog.xml` tree, check the TDM logs in `~/tdsconfig/logs/`. Also try restarting the containers on the Azure Docker host as directories may have been added by the LDM after TDS/TDM start up which the TDS/TDM apparently does not like:

```
cd ~/git/Unidata-Dockerfiles/ams2016
docker-compose stop
# remove stopped containers
docker-compose rm -f
# ensure containers are no longer running with
docker-compose ps -a
docker-compose up -d
```

You may also just have to **wait**. It can take a few hours for the TDM to catch up to what is going on in the `/data/lm` directory.

11.1.4 Cannot connect to the Docker daemon

Cannot connect to the Docker daemon. Is the docker daemon running on this host?

You may have simply forgotten to logout/login.

11.2 Asking for Help

For problems, help, questions, please submit an issue on github³⁴.

11.3 Acknowledgments

- National Science Foundation (Grant NSF-1344155)
- Microsoft "Azure for Research" program
- Tom Yoksas for Unidata operations expertise

³⁴<https://github.com/Unidata/Unidata-Dockerfiles/issues>