Geoscientific Data Distribution in the XSEDE Jetstream Cloud

Unidata on the Jetstream Cloud

Julien Chastang (UCAR, Boulder, CO USA), Mohan Ramamurthy, Tom Yoksas

2017-01-25 Wed

Contents

1 Introduction

This guide is a companion document (available in HTML, Markdown, text, PDF)[1] to a 2017 American Meteorological Society oral presentation, *Geoscientific Data Distribution in the XSEDE Jetstream Cloud*[2]. It describes how to configure the LDM[3], TDS[4], RAMADDA[5], and McIDAS ADDE[6] on XSEDE Jetstream VMs[7]. It assumes you have access to Jetstream 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. We will also be making use of the Jetstream API[8]. Obtain permission from the XSEDE Jetstream team to use Jetstream API. You must be comfortable entering commands at the Unix command line. We will be using Docker images available at the Unidata Github account[9] in addition to a configuration specifically planned for an AMS 2017 demonstration[10].

2 Obtain Jetstream Resources

Apply for cloud resource allocations on Jetstream[11].

3 Configure Jetstream to Run Unidata Docker Containers

3.1 Clone the Unidata-Dockerfiles Repository

We will be making heavy use of the `Unidata/Unidata-Dockerfiles` git repository. Install git[12] and clone that repository first:

```
1 git clone https://github.com/Unidata/Unidata-Dockerfiles
```

3.2 Build and Start the Jetstream API Docker Container

We will be using the Jetstream API directly and via convenience scripts. Install Docker (e.g., docker-machine[13]) on your local computing environment because we will be interacting with the Jetstream API in a Docker container.

```
1 cd Unidata-Dockerfiles/jetstream/openstack
```

3.2.1 Create ssh Keys

Create an `.ssh` directory for your ssh keys:

```
1 mkdir -p .ssh && ssh-keygen -b 2048 -t rsa -f .ssh/id_rsa -P ""
```

[1] https://github.com/Unidata/Unidata-Dockerfiles/tree/master/jetstream/readme
[2] https://ams.confex.com/ams/97Annual/webprogram/Paper315508.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://www.ssec.wisc.edu/mcidas/
[7] https://www.xsede.org/jump-on-jetstream
[8] https://iujetstream.atlassian.net/wiki/display/JWT/Using+the+Jetstream+API
[9] https://github.com/Unidata
[10] http://jetstream.unidata.ucar.edu
[11] https://www.xsede.org/jump-on-jetstream
[12] https://www.git-scm.com/book/en/v2/Getting-Started-Installing-Git
[13] https://docs.docker.com/machine/

3.2.2 Download openrc.sh

Download the `openrc.sh` file into the `Unidata-Dockerfiles/jetstream/openstack` directory according to the Jetstream API instructions[14]. In the Jetstream Dashboard, navigate to `Access & Security`, `API Access` to download `openrc.sh`.
Edit the `openrc.sh` file and the supply the TACC resource `OS_PASSWORD`:

```
export OS_PASSWORD="changeme!"
```

Comment out

```
# echo "Please enter your OpenStack Password: "
# read -sr OS_PASSWORD_INPUT
```

3.2.3 Build the openstack-client Container

Build the `openstack-client` container, here done via `docker-machine`.

```
docker-machine create --driver virtualbox openstack
eval "$(docker-machine env openstack)"
docker build -t openstack-client .
```

3.3 Set Up Jetstream API to Create VMs

Start the `openstack-client` container with

```
sh os.sh
```

You should be inside the container which has been configured to run openstack `nova` and `neutron` commands. Go though the following Jetstream API sections[15]:

- Create security group
- Upload SSH key
- Setup the network

At this point, you should be able to run `glance image-list` which should yield something like:

ID	Name
fd4bf587-39e6-4640-b459-96471c9edb5c	AutoDock Vina Launch at Boot
02217ab0-3ee0-444e-b16e-8fbdae4ed33f	AutoDock Vina with ChemBridge Data
b40b2ef5-23e9-4305-8372-35e891e55fc5	BioLinux 8

If not, check your setup.

3.4 Working with Jetstream API to Create VMs

3.4.1 IP Numbers

We are ready to fire up VMs. First create an IP number which we will be using shortly:

```
nova floating-ip-create public
nova floating-ip-list
```

or you can just `nova floating-ip-list` if you have IP numbers left around from previous VMs.

[14] https://iujetstream.atlassian.net/wiki/display/JWT/Setting+up+openrc.sh
[15] https://iujetstream.atlassian.net/wiki/display/JWT/OpenStack+command+line

3.4.2 Boot VM

Now you can boot up a VM with something like the following command:

```
boot.sh -n unicloud -s m1.medium -ip 149.165.157.137
```

The `boot.sh` command takes a VM name, size, and IP number created earlier. See `boot.sh -h` and `nova flavor-list` for more information.

3.4.3 Create and Attach Data Volumes

Also, create and attach `/data` and `/repository` volumes which we will be using shortly via the openstack API:

```
cinder create 750 --display-name data
cinder create 100 --display-name repository

cinder list && nova list

nova volume-attach <vm-uid-number> <volume-uid-number> auto
nova volume-attach <vm-uid-number> <volume-uid-number> auto
```

3.4.4 ssh Into New VM

`ssh` into that newly minted VM:

```
ssh ubuntu@149.165.157.137
```

If you are having trouble logging in, you may try to delete the `~/.ssh/known_hosts` file. If you still have trouble, try `nova stop <vm-uid-number>` followed by `nova stop <vm-uid-number>`.

3.5 Set up VM to Run LDM, TDS, RAMADDA, ADDE

3.5.1 VM Maintenance and Install git

As `root` (`sudo su -`), update, upgrade and install `git`:

```
apt-get update && apt-get -y upgrade && apt-get -y dist-upgrade && \
    apt-get -y install git ntp
```

Create a `git` directory for the `Unidata-Dockerfiles` project.

```
mkdir -p ~/git
```

3.5.2 Clone Unidata-Dockerfiles

Clone the the `Unidata-Dockerfiles` project.

```
git clone https://github.com/Unidata/Unidata-Dockerfiles ~/git/Unidata-Dockerfiles
```

3.5.3 Run the VM Set Up Script and Reboot

Install `Docker` and `docker-compose` and get the `ubuntu` user set up to run docker.

```
bash ~/git/Unidata-Dockerfiles/docker-vm-setup/ubuntu/setup-ubuntu.sh -u ubuntu \
     -dc 1.8.1
```

Reboot

```
reboot now
```

3.5.4 Check Docker Installation

Log back in to the VM as user `ubuntu`. Test `docker` with

```
docker run hello-world
```

If docker gives an error

```
docker: An error occurred trying to connect: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/cre
See 'docker run --help'.
```

Try as `root`

```
service docker stop
rm -rf /var/lib/docker/aufs #always think hard before rm -rf
service docker start
```

If the `hello-world` container runs smoothly, continue.

3.5.5 Mount Data Volumes

As `root`, run some convenience scripts to mount the data volumes for data being delivered via the LDM (`/data`) and RAMADDA (`/repository`).

```
bash ~/git/Unidata-Dockerfiles/jetstream/openstack/mount.sh -m /dev/sdb \
     -d /data
bash ~/git/Unidata-Dockerfiles/jetstream/openstack/mount.sh -m /dev/sdc \
     -d /repository

# ensure disks reappear on startup
echo /dev/sdb   /data    ext4  rw      0 0 | tee --append /etc/fstab > /dev/null
echo /dev/sdc   /repository   ext4  rw      0 0 | tee --append /etc/fstab > /dev/null
```

3.5.6 Clone Unidata-Dockerfiles and TdsConfig Repositories

We will again be cloning the `Unidata-Dockerfiles` repository, this time as user `ubuntu`.

```
mkdir -p ~/git
git clone https://github.com/Unidata/Unidata-Dockerfiles \
    ~/git/Unidata-Dockerfiles
git clone https://github.com/Unidata/TdsConfig ~/git/TdsConfig
```

3.5.7 Create Log Directories

Create all log directories

```
mkdir -p ~/logs/ldm/ ~/logs/ramadda-tomcat/ ~/logs/ramadda/ ~/logs/tds-tomcat/ \
      ~/logs/tds/ ~/logs/traefik/ ~/logs/tdm/
```

3.5.8 Configure the LDM

Grab the ldm `etc` directory

```
mkdir -p ~/etc
cp -r ~/git/Unidata-Dockerfiles/jetstream/etc/* ~/etc/
```

In the `~/etc` you will find the usual LDM configuration files (e.g., `ldmd.conf`, `registry.xml`). Configure them to your liking.

1. NTP As root, you also want to ensure the network time protocol configuration file accesses `timeserver.unidata.ucar.edu`.

```
sed -i \
    s/server\ 0.ubuntu.pool.ntp.org/server\ timeserver.unidata.ucar.edu\\nserver\ 0.ubuntu.pool.ntp.org/g \
    /etc/ntp.conf
```

3.5.9 Configure the TDS

In the `ldmd.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. Be sure to edit `~/tdsconfig/threddsConfig.xml` for contact information in the `serverInformation` element.

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

1. 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
   ```

   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
```

   Also ensure that `ldmfile.sh` is executable.

```
chmod +x ~/etc/TDS/util/ldmfile.sh
```

3.5.10 Configure RAMADDA

When you start RAMADDA for the very first time, you must have a `password.properties` file in the RAMADDA home directory which is `/repository/`. See RAMADDA documentation[16] 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! | tee --append \
  /repository/pw.properties > /dev/null
```

[16] http://ramadda.org//repository/userguide/toc.html

3.5.11 Configure McIDAS ADDE

```
cp ~/git/Unidata-Dockerfiles/jetstream/mcidas/pqact.conf_mcidasA ~/etc
mkdir -p ~/mcidas/upcworkdata/ ~/mcidas/decoders/ ~/mcidas/util/
cp ~/git/Unidata-Dockerfiles/mcidas/RESOLV.SRV ~/mcidas/upcworkdata/
```

3.5.12 Create a Self-Signed Certificates

In the `~/git/Unidata-Dockerfiles/jetstream/files/` directory, generate a self-signed certificate with `openssl` (or better yet, obtain a real certificate from a certificate authority).

```
openssl req -new -newkey rsa:4096 -days 3650 -nodes -x509 -subj \
  "/C=US/ST=Colorado/L=Boulder/O=Unidata/CN=tomcat.example.com" \
  -keyout ~/git/Unidata-Dockerfiles/jetstream/files/ssl.key \
  -out ~/git/Unidata-Dockerfiles/jetstream/files/ssl.crt
```

3.5.13 TDS Host and TDM User

Ensure the `TDS_HOST` URL (with a publicly accessible IP number of the docker host or DNS name) is correct in `/git/Unidata-Dockerfiles/jetstream/docker-compose.yml`.
In the same `docker-compose.yml` file, ensure the `TDM_PW` corresponds to the SHA digested password of the `tdm` user `/git/Unidata-Dockerfiles/jetstream/files/tomcat-users.xml`

```
docker run tomcat /usr/local/tomcat/bin/digest.sh -a "SHA" mysupersecretpassword
```

3.5.14 Configure TDM

TDM logging will not be configurable until TDS 5.0[17]. Until then:

```
curl -SL \
     https://artifacts.unidata.ucar.edu/content/repositories/unidata-releases/edu/ucar/tdmFat/4.6.8/tdmFat-4.6.8.
     -o ~/logs/tdm/tdm.jar
curl -SL https://raw.githubusercontent.com/Unidata/thredds-docker/master/tdm/tdm.sh \
     -o ~/logs/tdm/tdm.sh
chmod +x ~/logs/tdm/tdm.sh
```

3.6 chown for Good Measure

As `root` ensure that permissions are as they should be:

```
chown -R ubuntu:docker /data /repository ~ubuntu
```

4 Start Everything

Fire up the whole kit and caboodle with `docker-compose.yml` which will start:

- LDM
- Traefik[18], a reverse proxy that will channel ramadda and tds http request to the right container
- NGINX web server

[17] https://github.com/Unidata/thredds-docker#capturing-tdm-log-files-outside-the-container
[18] https://traefik.io/

- RAMADDA
- THREDDS
- TDM
- McIDAS ADDE

As user `ubuntu`:

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

4.1 Bootstrapping

The problem at this point is that it will take a little while for the LDM to fill the `/data` directory up with data. I don't believe the TDS/TDM can "see" directories created after start up. Therefore, you may have to bootstrap this set up a few times as the `/data` directory fills up with:

```
cd ~/git/Unidata-Dockerfiles/jetstream/
docker-compose stop && docker-compose up -d
```

5 References

Stewart, C.A., Cockerill, T.M., Foster, I., Hancock, D., Merchant, N., Skidmore, E., Stanzione, D., Taylor, J., Tuecke, S., Turner, G., Vaughn, M., and Gaffney, N.I., Jetstream: a self-provisioned, scalable science and engineering cloud environment. 2015, In Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure. St. Louis, Missouri. ACM: 2792774. p. 1-8. http://dx.doi.org/10.1145/2792745.2792774

John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, Nancy Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", Computing in Science & Engineering, vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80

6 Acknowledgments

We thank Jeremy Fischer, Marlon Pierce, Suresh Marru, George Wm Turner, Brian Beck, Craig Alan Stewart, Victor Hazlewood and Peg Lindenlaub for their assistance with this effort, which was made possible through the XSEDE Extended Collaborative Support Service (ECSS) program.