DRAFT COMBINE-DOCKER v0.11.1 Installation Instructions DRAFT

INTRODUCTION

Combine is an application to facilitate the harvesting, transformation, analysis, and publishing of metadata records by Service Hubs for inclusion in the Digital Public Library of America (DPLA).

These technical instructions are for metadata experts or system administrators who wish to install Combine on their institution's servers to process and publish metadata records to DPLA. This version (v0.11) includes a number of bug fixes and improvements. You can find the version change documentation at nnnnnnnnnn.

Help is available for combine installation at combine-support@umich.edu. You can also add issues to the combine github at https://github.com/MI-DPLA/combine.

ASSUMPTIONS

1. A linux server on which you have superuser privileges (can run **sudo** commands).
    a. Instructions tested on Ubuntu 18.04; not guaranteed to work on other Linuxes, but likely to work on later Ubuntus
2. These instructions are for installing v0.11.1 of Combine which requires docker and includes steps for migrating from an earlier version of Combine. The Vagrant/Ansible installation options are no longer available.
3. Enough space to run the old and new Combines while migrating data, plus the space needed for the exported state data files from the older version. Also, some of the docker images are large!
4. There are conditional instructions for various errors you may encounter. If you find additional errors or have suggested improvements please let us know so we can improve these instructions at at combine-support@umich.edu.

INSTALLING COMBINE-DOCKER

1. Clone the git repository https://github.com/MI-DPLA/combine-docker.git somewhere. We chose /opt/combine-docker for symmetry with the old method
2. Make sure that dependencies are installed:
    a. Docker
        i. `sudo apt update`
        ii. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
        iii. `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"`
        iv. `sudo apt update`
        v. `sudo apt-get install docker-ce`
        vi. Add the relevant user account(s) to the docker group: **`sudo usermod -aG docker ${USER}`** (this is necessary to run docker commands without sudo, which we need for docker-compose basically).
        vii.
    b. Docker-compose
        i. `sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
        ii. `sudo chmod +x /usr/local/bin/docker-compose`
    c. Subversion, which is needed to pull down the spark/livy static files without pulling down the entire repository.
        i. `sudo apt-get install subversion`
3. Log out and log back in so that the group change sticks
4. **cd** to your combine-docker directory and make sure you have permission to edit it.
    a. Maybe change the group to the docker group?
    b. Maybe consider having a service account?
    c. You may also need to change the owner to yourself? *Argh, we had to do this at Michigan.*

5. **`git checkout master`**
6. Run the **`build.sh`** script
    a. You might get a lot of warnings. Most of these are fine to ignore (for instance, Livy wants to build off a Zinc server, but nothing bad happens when it doesn't find one)
7. IF you get dns failures:
    a. One way to resolve this might be to add the config file at the path `/etc/docker/daemon.json` with the following contents:

    ```
    {
          "dns": ["{your dns server address}"]
    }
    ```

    and then `sudo systemctl reload docker` / `sudo systemctl restart docker`
        i. If the daemon.json file does not exist on your machine already, you will have to create it at the path /etc/docker/daemon.json
    b. You can test if dns is even kind of working with `docker run busybox /bin/sh -c "nslookup github.com"` -- if it times out, nothing at all is working
    c. If step (a) doesn't fix the problem, look up something to do with `docker` and `resolved`
8. Run the **`build.sh`** script again if you had to resolve dns failures.
9. IF it is endlessly waiting for MySQL container to be ready...
    a. The script that happens in is `combine-docker/combine/combine_db_prepare.sh`
    b. Try running each of those commands in that shell script in order using the command template **"`docker-compose run combine-django /bin/bash -c "${command}"`"**
    c. Once you've successfully done all the db-preparing, proceed to the next step.
10. Run the **`buildstatic.sh`** script.
11. Change nginx's/docker-compose's exposed ports to ones not in use by your server's existing applications (e.g., change '127.0.0.1:80:80' to '28080:80' in docker-compose.yml to expose host port 28080, or pick some other port not already in use). Note that you do not need to change the listen port in nginx.conf, because

docker is taking the traffic sent to port 28080 on the host and redirecting it to the nginx docker image's port 80, where nginx is actually listening.

12. Try **`docker-compose up`**. Repeat the port-changing process for anything that errors out as conflicting (most other ports live in docker-compose.yml).
13. Try going to **${url}:28080/combine** (or whatever port) in a browser to confirm that your docker combine is working; **log in** with user ***combine*** and password ***combine***

MIGRATING DATA FROM PREVIOUS VERSIONS OF COMBINE

14. **Background:** Exporting and Importing of "States" in Combine is the ability to select various level of hierarchy (Organizations, Record Groups, and/or Jobs), and optionally Configuration Scenarios (OAI Endpoints, Transformations, Validations, etc.), and export to a fully serialized, downloadable, archive file. This file can then be imported into the same, or another, instance of Combine and reconstitute all the pieces that would support those Jobs and Configurations.
15. IF your old Combine has more than one user account, make sure you create the *SAME NUMBER* OF USERS for your new Combine. This is an unfortunate required workaround for the moment to import your existing Combine data. To add users to the new Combine:
    a. Use the "Configuration" menu item in the top navigation bar then the "Django Admin Console" button at the top of the page. Then click "Home" in the breadcrumbs bar to get to the Django administration page and click the **Users** link. **OR**
    b. Enter **${url}:28080/admin** into your browser's address bar to get to the Django administration page and click the **Users** link.
16. If everything's working, try exporting everything from your old Combine:
    a. You can get to State Export/Import using the "Configuration" menu item in the top navigation bar then scroll down to the bottom of the page to find the "State Export/Import" button. **OR**
    b. At **${url}/combine/stateio/export** (type this into your browser address bar).
17. Click the **Export State** button and on the next page select everything you want to export and again click the **Export State** button. The export process may take a while or get stuck if it runs out of space.

18. This is a point at which you might run out of disk space and may need to work with your system support to have more allocated.
19. Now you will need to find the export on disk and put it somewhere that the new Combine can see. In theory, with all settings at default, the following should work:
    a. docker exec -it combine-docker_combine-django_1 /bin/bash
    b. cd /home/combine/data/combine/stateio/exports (or the STATEIO_EXPORT_DIR from localsettings.py)
    c. Find the .zip file you just created
    d. cp ${blah}.zip /opt/combine (/opt/combine in both combine-django and combine-celery should be the same mapped docker volume)
    e. Go to import page and import from filesystem with path "/opt/combine/${blah}.zip"
20. IF your version of Combine is old enough that it still has the bug where jobs with no upstream or downstream job can't be exported, you can work around that by creating a meaningless Analysis job that takes all the singletons as inputs.
21. Each imported job has a button that allows you to rebuild the elasticsearch index for that job. (We at Michigan are looking into a more global way to do this.)

TROUBLESHOOTING

ElasticSearch container dies because of vm.max_map_count

Depending on the machine and the OS (Linux, Mac, Windows), you might need to bump the vm.max_map_count on the Docker host machine (seems to be particularly true on older ones)

https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html#docker-cli-run-prod-mode

Port collision error: port is already allocated

By default, nearly all relevant ports are exposed from the containers that conspire to run Combine, but these can turned off selectively (or changed) if you have services running on your host that conflict. Look for the ports section for each service in the docker-compose.yml to enable or disable them.

java.lang.ClassNotFoundException: org.elasticsearch.hadoop.mr.LinkedMapWritable

Make sure that the elasticsearch-hadoop-x.y.z.jar in combinelib matches the version specified in the ELASTICSEARCH_HADOOP_CONNECTOR_VERSION environment variable configured in your .env.

Other issues?

Please don't hesitate to submit an issue!