

Setting up DSA Instance at UF

The Resource is setup via PubApps; <https://athena.rc.ufl.edu>, and it is migrated from the instance at Buffalo; <https://athena.ccr.buffalo.edu>.

Digital slide archive (DSA) runs five docker containers via a docker compose. This system is for large histology image data visualization as well as running plugins for conducting image segmentation. This will be a prototype only, with long term goal will be to use for general audience. For now we want to demonstrate the use to our funders.

We have three domain names: athena.rc.ufl.edu, devathena.rc.ufl.edu, sandboxathena.rc.ufl.edu and they are mapped to <http://pubcontainer:8101>, <http://pubcontainer:8102>, <http://pubcontainer:8103>. The sandbox instance can only be accessed via UF network or UF-VPN.

Access restrictions:

Public Open - Visible outside of the UF network. There are no access restrictions or authentication is used only for administrative paths.

We are using rootless docker for running all the docker images and special configuration is needed for that which is documented [here](#)

The userID configured by the IT with all the required permissions to run applications in a rootless mode can be found under the directory: /var/run/user. The specified path can be helpfull in debugging user related erros.

To ssh into pubcontainer:

Login to HiperGator

```
ssh pubcontainer
```

See <https://docs.docker.com/engine/security/rootless/> for reference regarding rootless docker.

Computational, storage, and database resources:

Our purchased PubAPPS has 16TB storage, we also have 28 :14 core CPUs and 2 NVIDIA 1080Ti GPUs with 12GB memory each for computation. The compute node memory on a quote is 128GB. The OS is RHEL CentOS 8.

NOTE FOR THE DEVELOPER:

We need to set up CI (continuous integration) and CD (continuous deployment) pipeline, so code changes in an external git repository get applied to the application and docker images are automatically built and deployed.

The DSA instance codes are available at:

https://github.com/DigitalSlideArchive/digital_slide_archive

<https://hub.docker.com/r/sarderlab/histo-cloud/tags>

<https://github.com/SarderLab/Histo-cloud> The above github pages have all the dependencies.

The DSA instance docker images are available on Docker Hub under sarderlab Note: Temporarily a couple of docker plugin images are available under suhaskc repo in Dockerhub.

Documentation for migration of data between two DSA instances can be accessed [here](#).

The /pubapps/athena directory tree on pubcompute is using an NFS-mounted directory tree.

Adding a plugin to DSA

To add a plugin to DSA from a docker image navigate to Admin Console -> Plugins -> Slicer CI Web (Click on the Settings Icon) Under 'Import Docker Image CLI Tasks': Give image:tag in the 'Docker Image' field and click on Import Image.

Note: We need to expand the system next to use multiple GPUs and CPUs in a backend HPCC to run several worker nodes.(Currently working with 2GPUs now)

Accessing and configuring users on pubcontainer: ssh key should be added. Ask ahmed.

```
~/.ssh/config
Host pubapps
  User athena
  HostName pubcontainer
  KeepAlive yes
```

GIVING ACCESS TO A NEW USER TO PUBAPPS:

```
nano ~/.ssh/config

Host pubweb
  User athena
  HostName pubweb
  KeepAlive yes

Host pubcontainer
  User athena
  HostName pubcontainer1
  KeepAlive yes
```

Add the ssh key generated by the user on HiperGator to the authorized users file

```
~/.ssh/authorized_keys
```

The host that is going to be running all the backend containers is pubcontainer1, but it's not accessible directly, yet. You only have access to it via ssh to pubcontainer.

```
$ ssh pubapps
$ ssh pubcontainer
```

Note for Sayat and Alex: There are dsa files on pubweb and we can ssh into it. But they are not being used. We might need to delete and cut off the entire access.

dev and prod directory trees are seperated. ~/dsa/prod and ~/dsa/dev as example.

Purchase of long term storage with backup

<https://www.rc.ufl.edu/get-started/purchase-allocation/> - purchase either hardware or service purchase of Orange storage and specify the need for 'TSM Backup' in the Note when filling out the form.

(https://help.rc.ufl.edu/doc/Getting_Started#Purchasing_Resources)

File transfer from another institute

As we already have ssh access to the Buffalo machine. For this case we can do it as:

```
[athena@pubweb1 prod]$ pwd
/pubapps/athena/dsa/prod
[athena@pubweb1 prod]$ rsync -av user@mymachine.buffalo.edu:/data/directory/ data/
```

This will do the job. Someone from the team can start a screen or tmux. (https://help.rc.ufl.edu/doc/Persistent_Terminal_Sessions) on a login node, ssh into pubweb in that session and run the rsync.

Setting up Winscp for Pubapps

If we have to do file transfer via a windows based machine, say via WinSCP: Check this link <https://support.rc.ufl.edu/attachment.cgi?id=3455> Using WinSCP to sftp data to athena webserver

From a quick look the https://winscp.net/eng/docs/ui_login_tunnel appears to be the key to get it working. See the attached screenshot of a WinSCP SFTP session from a local Windows desktop to athena@pubapps.

Login settings:

File protocol: SFTP Host name: pubweb.ufhpc Port number 22 User name athena

Advanced > Connection > Tunnel settings:

CONNECT THROUGH SSH TUNNEL

Host name: hpg.rc.ufl.edu Port number: 22 User name: GATORLINK_USERNAME Password: REDACTED

Since there is no 'password' for athena user there's one more setting: Advanced > SSH > Authentication Private key file: putty_id_rsa.ppk

Note: Convert the text of the public putty key file to a linux ssh format and add it to ~/.ssh/authorized_keys for athena@pubweb user.

Example: the putty public key generated on the windows desktop looks like

```
Comment: "rsa-key-putty"
AAAAB3NzaC1yc2EAAAABJQAAAQEAh+U+5ZHfDBqdwvyaORPLf1khpjHJ1F9uHQVY
GzfKuxoLz2EQrdL5ob4jY2Ao9Hr0Q114djm/pkwnAD30k2Pc9lRx/jTT2SanCbHf
Ldr5pMWJJDmoAH5+qWT3E3LmMGjwnQ1x3ld06q66iW9GBWGTQVeGv7x6BXRIeBS
nRGTv7C4zMLb0ZZH0c+dJSBpsiFQ4DHxhFcirEN754cHgapBB6h0AGMhHjm7SY0B
F/Karn18gIj6yqlwVxbR+a44d7cRGXsAAS+5MXeUteKkmUIzqlddxAb709mcY0gj
PXjP+Qsl/H/zXI2h3Bce5SpUxinxP6xKIyLAN/qYJrWCJd9iDw==
```

and here's how it looks when (user) joined the ssh lines and added the ssh-rsa header and a comment all as a single line of text:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAh+U+5ZHfDBqdwvyaORPLf1khpjHJ1F9uHQVYGzfKuxoLz2EQrdL5ob4jY2Acrsa-key-putty
```

Note: In Long-term plan, we're planning to switch pubcontainer from the rootless docker-ce runtime to RHEL podman runtime with a docker compatibility layer as a more stable long-term environment.

How to start the docker containers:

```
[athena@c36a-s5 ~]$ cd dsa/prod
$ source env.sh
$ cd digital_slide_archive/devops/dsa
$ docker compose up -d
```

To stop:

```
$ cd dsa/prod/digital_slide_archive/devops/dsa
$ docker compose down
```

Rootless configuration of docker applications

```
[athena@c36a-s5 digital_slide_archive]$ git diff
diff --git a/devops/dsa/docker-compose.yml b/devops/dsa/docker-compose.yml
index 3de38ca..1f93258 100644
--- a/devops/dsa/docker-compose.yml
+++ b/devops/dsa/docker-compose.yml
@@ -5,18 +5,18 @@ services:
   image: dsarchive/dsa_common
   build: ../..
   # Instead of privileged mode, fuse can use:
   # devices:
   #   - /dev/fuse:/dev/fuse
+  devices:
+    - /dev/fuse:/dev/fuse
   # security_opt:
   #   - apparmor:unconfined
-  # cap_add:
-  #   - SYS_ADMIN
+  cap_add:
+    - SYS_ADMIN
```

But these may be somewhat host specific, so we default to privileged. If the docker daemon is being run with `--no-new-privileges`, fuse may not work. See also <https://github.com/docker/for-linux/issues/321> for possible methods to avoid both privileged mode and `cap_add SYS_ADMIN`. - privileged: true + privileged: false Set `DSA_USER` to a user id that is part of the docker group (Example: `DSA_USER=$(id -u):$(id -g)`). This makes files in assetstores and logs owned by that user and provides permissions to manage docker.

Here's the contents of the env.sh for the athena.rc.ufl.edu:

```
[athena@c36a-s5 prod]$ cat env.sh
```

Girder logs are in `~/dsa/prod/digital_slide_archive/devops/dsa/logs`

Errors:

Issue: In the docker-compose yaml file, we expect docker.sock to be mounted to `/var/run/docker.sock` (e.g., volume of `/run/user/1276600062/docker.sock:/var/run/docker.sock` or `${DOCKER_SOCKET}:/var/run/docker.sock`)

Solution: On the host system each rootless user has their own socket at \$XDG_RUNTIME_DIR/docker.sock which in case of athena is indeed /run/user/1276600062/docker.sock

Suhas error :

Following error on mongodb container. - If you get the below error first check the permission for db folder and give the necessary permissions.

```
{"t":{"$date":"2023-02-02T14:16:15.177+00:00"},"s":"E", "c":"CONTROL", "id":20557, "ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"IllegalOperation: Attempted to create a lock file on a read-only directory: /data/db } }
```

solution:

```
[athena@c36a-s5 dsa]$ chmod 777 db
```

Permission issues with rootless docker:

The previous path became invalid, because now the docker is running with non-root privileges and the path was changed from /var/run/docker.sock to /var/run/user/{non_root_user_id}/docker.sock

```
In dev/env.sh + export DOCKER_SOCKET=/var/run/user/${id -u}/docker.sock In /dev/digital_slide_archive/devops/dsa/docker-compose.yaml volumes: (-) - /var/run/docker.sock:/var/run/docker.sock (+) - ${DOCKER_SOCKET}:/var/run/docker.sock
```

(-) - \${TMPDIR}:/tmp}:\${TMPDIR}:/tmp} TmpDIR was causing some error and we temporarily disabled it. Solution:

1) We can either change the permission of the folders created during runtime by changing the permission codes (766 or 777 as required) or change default permissions of that folder through startup scripts. 2) Set the same user id (1001) by modifying the running image (dsaarchive/histomicstk), as we are running in rootless docker. This is a temporary fix for long term this is not suitable.

The issue is resolved by the above solution and they were able to pull images successfully from Plugins.

This changes are running on <https://devathena.rc.ufl.edu/> currently.

NOTE: Both db and assestore should be located upstream of the digital_slide_archive/devops/dsa e.g. in

/pubapps/athena/dsa/prod/db and /pubapps/athena/dsa/prod/assetstore respectively.