**DEPLOYING GALAXY VIA DOCKER**

**PUSHING CODE CHANGES TO GALAXY**

1. Make code changes in your normal development environment.
2. Execute **build\_galaxymapgen.xml** ANT script in the top level project directory (using Run feature in eclipse)
3. Copy GalaxyMapGen.JAR from your NGCHM project directory (**c:\workspace\Projects\NGCHM**) to the Galaxy project directory (**c:\users\stucky\git\NGCHM\_Galaxy**)on your local machine.
4. Execute build\_ngchmApp.xml in the top level project directory (using Run feature in eclipse) to create the Javascript and Downloadable NGCHM App
5. Copy ngchmWidget-min.js from NGCHM project to **NGCHM\_Galaxy\mda\_heatmap.viz\static\javascript**
6. Zip up ngchmApp.html (as ngchmApp.zip) and copy from NGCHM\_project webContents directory to **NGCHM\_Galaxy\mda\_heatmap.viz\static**
7. Stop/Remove/Build/Run docker galaxy (using steps above)

**DEPLOYING CODE CHANGES TO GALAXY PRODUCTION**

**(these steps are executed from CMD window. Same dir as DEV instructions)**

1. Build production version: **docker build -t ngchm/ngchm-galaxy .**
2. Login: **docker login**
3. Username: **rbrown1422**
4. Password: **Dooddood%5**
5. Push Docker: **docker push ngchm/ngchm-galaxy:latest**
6. Get latest image id (eg **13dc6215cd85**)
7. Tag Docker version: **docker tag <image id> ngchm/ngchm-galaxy:2.7.0**
8. Push tagged version: **docker push ngchm/ngchm-galaxy:2.7.0**

**DEPLOYING CODE CHANGES TO GALAXY TOOLSHED**

1. In Eclipse Galaxy Project run: **build\_heatmap\_gen.xml**
2. Goto Galaxy toolshed website: **https://toolshed.g2.bx.psu.edu/**
3. login as: [**rbrown@insilico.us.com/Iooiiooi%5**](mailto:rbrown@insilico.us.com/Iooiiooi%255)
4. Find the tool Search Repositories: **NG-CHM**
5. Click on the tool in question (e.g. **heat\_map\_creation**)
6. Click **Repo action** button
7. Select **Upload Files to Repository**
8. Browse to GIT location and select tool (eg **mda\_heatmap\_gen.tar**)
9. Press **Upload** (to do another you **MUST** go to that tool **BEFORE** upload)
10. Look for green success message

**DEPLOYING A GENERIC DOCKER INSTANCE TO TEST TOOLSHED DOWNLOAD**

1. Clean up and remove any existing galaxy instances.
2. Run generic Galaxy version: **docker run --name="NG-CHM" -d -p 8888:80 -p 7021:21 -e "**[**GALAXY\_CONFIG\_ADMIN\_USERS=mark@mark.com**](mailto:GALAXY_CONFIG_ADMIN_USERS=mark@mark.com)**" bgruening/galaxy-stable**
3. Register and login as: [**mark@mark.com**](mailto:mark@mark.com) (this will make you administrator)
4. Go to **Admin** (on top menu bar)
5. Select **Install New Tools** (on left side menu bar)
6. Click little arrow on **Galaxy Main Toolshed** button and pick **search for valid tools** from dropdown
7. Enter *NG-CHM* into **Tool Name** and press **Search Repositories** button
8. You will be presented with a list of std and advanced versions.
9. Find the one (or both) that you want, click the check box next to each, and press **Install To Galaxy** button.
10. Enter a name to install the tools under (e.g. *NG-CHM Tools*) and press **Install** button.
11. If you go to Analyze Data (on the top menu bar) you should now see a subsection in the left-side menu called **NG-CHM Tools**. HOWEVER you are not done yet because the visualizations must still be installed.
12. Bash into container: **docker exec -it NG-CHM /bin/bash**
13. Copy the ng-chm plugin: **mv /galaxy-central/../shed\_tools/toolshed.g2.bx.psu.edu/repos/md-anderson-bioinformatics/heat\_map\_creation/\*/heat\_map\_creation/mda\_heatmap\_viz.zip  /galaxy-central/config/plugins/visualizations/**
14. Here we need to do a few things to configure the environment:
    1. Run command: **apt-get update** (this will get all updates for zip commands and java)
    2. Run command: **apt-get install -y r-base**
    3. Run command: **apt-get install -y software-properties-common python-software-properties**
    4. Run command: **add-apt-repository -y ppa:openjdk-r/ppa**
    5. Run command: **apt-get install -y openjdk-8-jdk**
    6. Run command: **update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java**
15. Change directory to: **cd /galaxy-central/config/plugins/visualizations/**
16. Run command: **apt-get update**
17. Run command: **sudo apt-get install unzip**
18. Unzip the visualization: **unzip mda\_heatmap\_viz.zip** (note if unzip command is not found something went wrong installing unzip)
19. **Exit** the shell
20. Stop the container: **docker stop NG-CHM**
21. Start the container: **docker start NG-CHM** (do not use the RUN command again or you will have to start all over)

**EXTRA NOTES:**

OK. I did not write those instructions Bob did so they are a bit wack. First you need to get into the galaxy virtual machine so you do the docker exec -it <container name or id> /bin/bash. Then you need to copy and unzip the ng-chm plugin. Something like: 1) mv /galaxy-central/../shed\_tools/toolshed.g2.bx.psu.edu/repos/md-anderson-bioinformatics/heat\_map\_creation/\*/heat\_map\_creation/mda\_heatmap\_viz.zip  /galaxy-central/config/plugins/visualizations/  
2) cd /galaxy-central/config/plugins/visualizations/  3) unzip mda\_heatmap\_viz.zip  Then you must restart Galaxy for the visualization portion to take effect.

**Build incident from 3/31/2021**

Overview: Everytime that I tried to run the build it failed in an early step of the dockerfile to do the command ***apt-get update***. There were 2 issues:

1. We received the following errors in step 37 of apt-get: *GPG error: http://dl.yarnpkg.com stable InRelease: The following signatures were invalid: KEYEXPIRED 1507181400 KEYEXPIRED 1546376218 KEYEXPIRED 1546372003 KEYEXPIRED 1580619281 KEYEXPIRED 1580607983*
2. We received gnutls handshake errors trying to reach [*http://research.cs.wisc.edu/htcondor/ubuntu/stable/dists/trusty/contrib/binary-amd64/Packages*](http://research.cs.wisc.edu/htcondor/ubuntu/stable/dists/trusty/contrib/binary-amd64/Packages)during step 39 of the *apt-get update* process.

Issue #1: This was resolve by searching for the error and finding the following link: <https://askubuntu.com/questions/1306111/why-yarn-key-update-failes-all-the-time-gpg-error-https-dl-yarnpkg-com-debia>. That link suggested adding the following line to the dockerfile before *apt-get update*: RUN apt-key adv --refresh-keys --keyserver keyserver.ubuntu.com Adding the line resolved the issue.

Issue #2: This one was harder to figure out. The first thing we needed to do was make sure that it was not our code causing the issue. So, we did a run of just the base docker image (bgruening/galaxystable) on your local machine… then run this command: **docker run --name="galaxy\_test" -d -p 8888:80 -p 7021:21 -e "GALAXY\_CONFIG\_ADMIN\_USERS=mstucky@insilico.us.com"** bgruening/galaxy-stable

Running the above command allowed us to then bash into the base bgruening image and run the *apt-get* update command within that image. YES… the error was still there. Since our image is updated from the base in each build, we cannot edit files there, run the script, and expect to see results.

Since the error was there, our next step was to find references to <http://research.cs.wisc.edu> :

**Steps:**

1. Cd to etc/apt
2. Recursive Grep for wisc: grep wis \* -r
3. Found reference at: sources.list.d/htcondor.list:deb http://research.cs.wisc.edu/htcondor/ubuntu/stable/ trusty contrib

The next thing to do would be to see that if we removed that reference from htcondor.list.d, apt-get update would work.

**Steps:**

1. Delete the offending line from the file htcondor.list.d
2. echo “” > /etc/apt/sources.list.d/htcondor.list
3. Run *apt-get update*
4. Worked
5. So restore the line to htcondor.list and get out of the base docker image.

The final thing to do was to add the line ***RUN echo “” > /etc/apt/sources.list.d/htcondor.list*** to our Dockerfile before the *apt-get update* command. After doing this the build works. It is important to note that we probably were not updating any of the files in the docker image with stuff to be found at wisc.edu BUT it did not affect the application.