ACT Install from Docker

Running this image will require a recent supported installation of the Docker Engine

You will also need root or superuser access to run this install.

If you are not familiar with Docker

Getting started with Docker

Operating systems checked

Our testing has shown that this ACT Docker container will run on: CentOS 7, Windows 10 Professional, and Mac OS. Also, this container will run on Windows 7 with the Docker Toolbox for Windows installed.

Dockerhub

Our public Dockerhub repository page is here: ACT On this page you will find abridged instructions on how to run this image into a container on your local host.

1

Download this script: launcher This script is a helper that will prompt for parameters that are unique to your site. Then the script will execute a Docker run with these parameters.

2

Pull the Docker image by:

```
docker pull i2b2/act-web
```

3

Run the launcher script that you downloaded in Step 1.

```
./launcher.sh
```

4

The script will prompt your for your SHRINE URL. Enter in your SHRINE URL.

```
Enter in your SHRINE URL: http://[your SHRINE domain]:9094/shrine-act-test/
```

5

The script will now prompt you for your i2b2 domain. Enter it here.

```
Enter in your domain: [your i2b2 domain]
```

6

The script will now prompt you for your PM cell URL. Enter it here.

```
Enter in your PM cell URL: http://[your PM domain]:9090/i2b2/serv
```

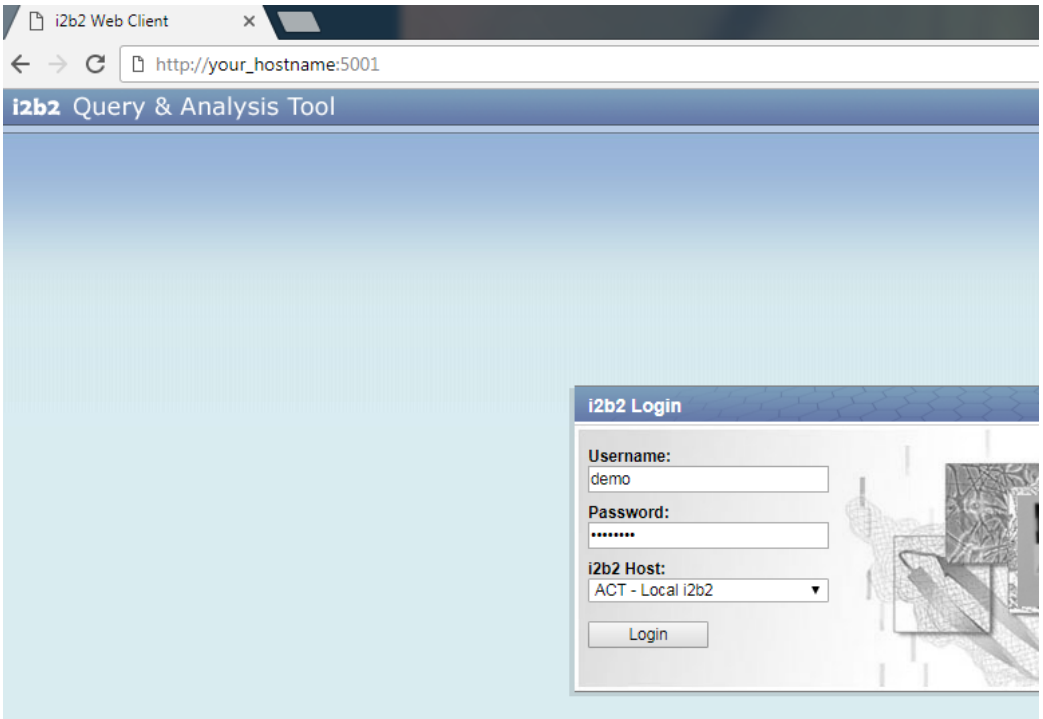
7

The script will now build the ACT image, and run the ACT container. All outputs of these commands can be found in ACTStart.log

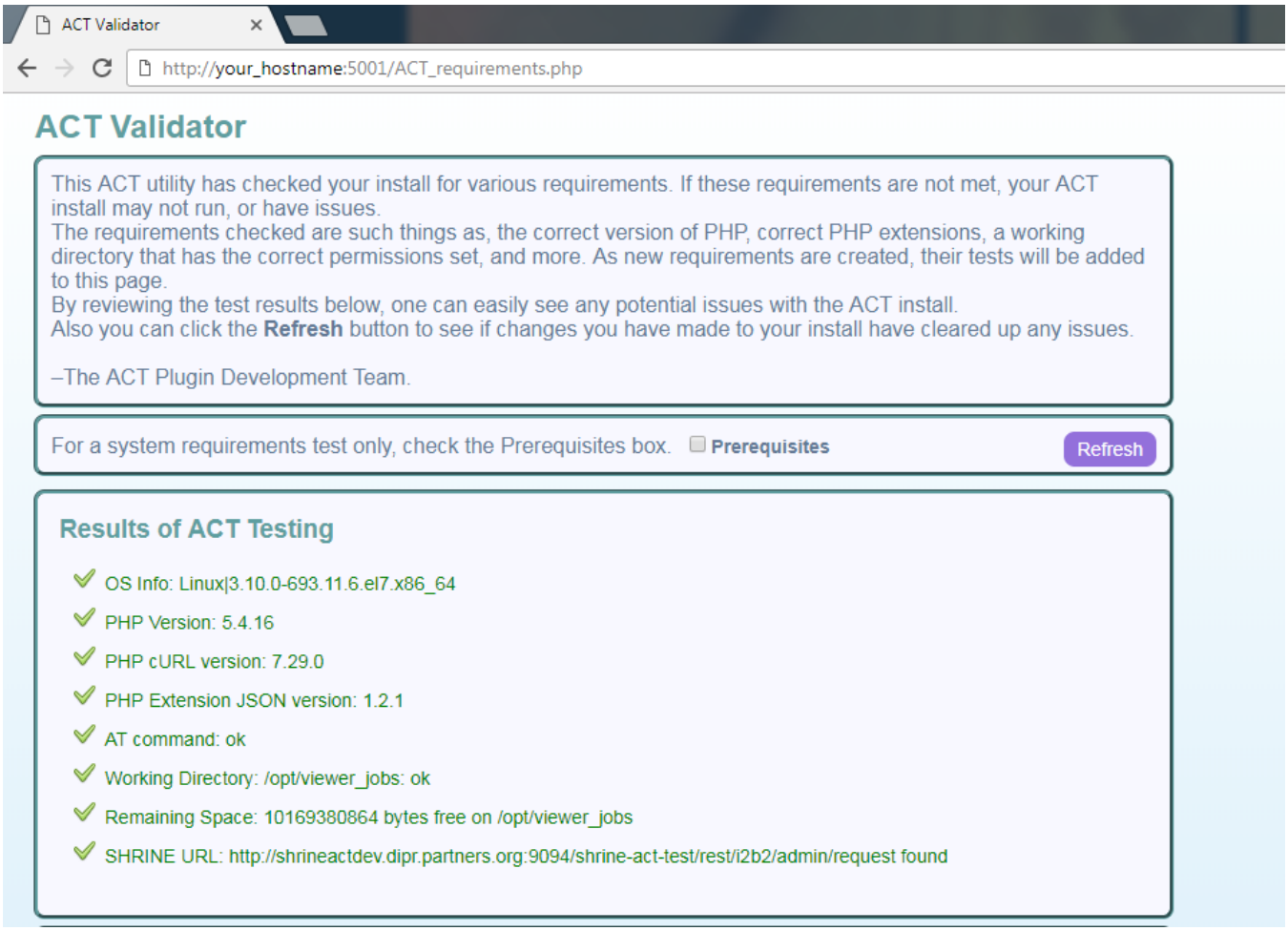
```
Now starting the ACT container. Please wait...


ACT container should be running at http://your_hostname:5001/
```

8

Verify the ACT Web Client is running by going to http://your_hostname:5001 in a browser.



If there are any issues, go to http://your_hostname:5001/ACT_requirements.php



You have completed all the steps! Here are some helpful Docker commands:

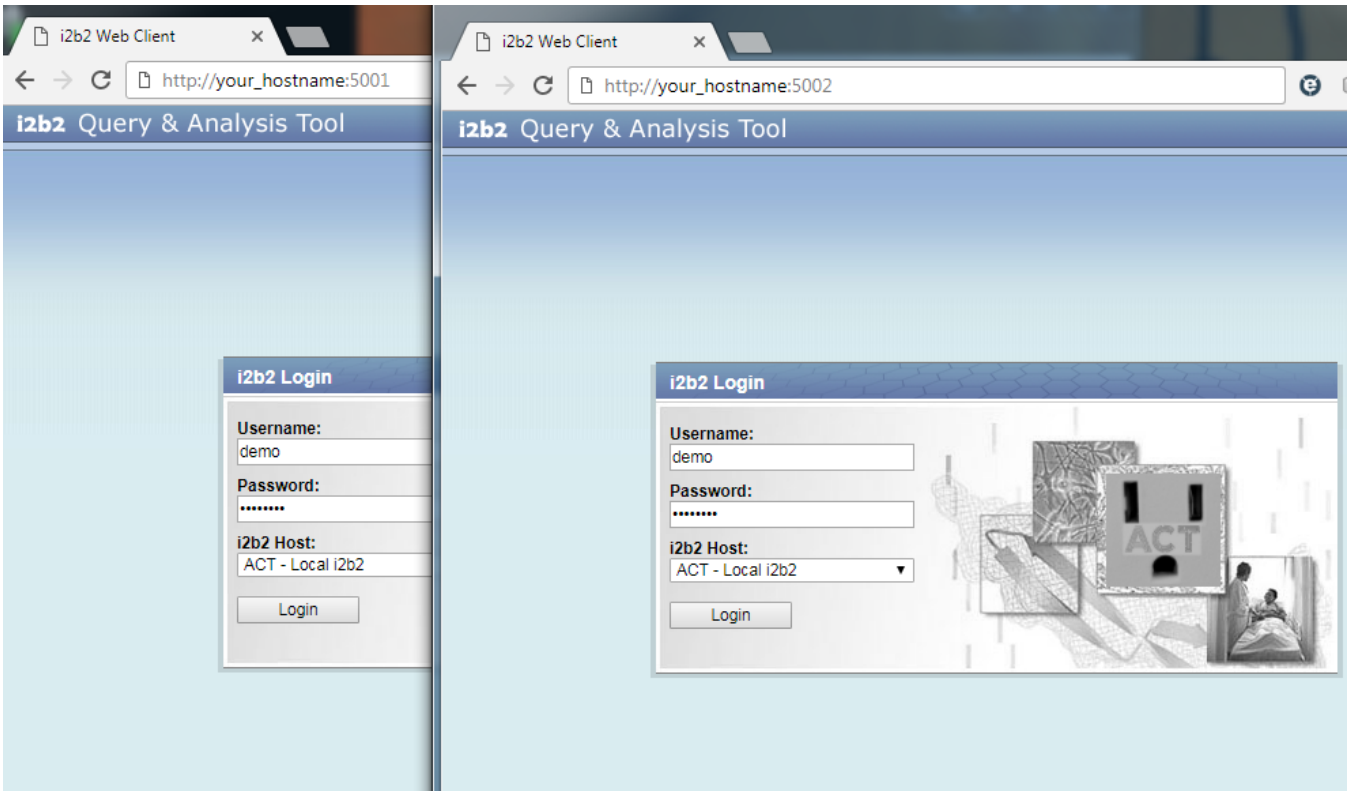
To delete the running act_container, enter in: docker container rm -f act_container

```
[root@acttest act4]# docker container rm -f act_container
act_container
[root@acttest act4]#
```

To run a container based on the act image, type in docker run --name [anyname] -d -p [hostPort:80] name of image]
In the exmaple beow we have started two ACT containers running at the same time

```
[root@acttest act4]# docker run --name=actTwo -d -p 5001:80 act_image
d49d698b37bf5d176728b3efee1b20883eed31699502df2bd9429217e112b169
[root@acttest act4]# docker run --name=actThree -d -p 5002:80 act_image
3ab81746f70115f5f186f7a71fb0101003081e08b06be7b4617d9d85eeee5b84
[root@acttest act4]#
```

Here are the two containers' URLs running at the same time! Note the port numbers.



To get a list of running containers type in: docker container ls

```
[root@acttest act4]# docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                   NAMES
3ab81746f701        act_image           "/bin/sh -c '/usr/sb…"   7 minutes ago       Up 7 minutes        0.0.0.0:5002->80/tcp    actThree
d49d698b37bf        act_image           "/bin/sh -c '/usr/sb…"   7 minutes ago       Up 7 minutes        0.0.0.0:5001->80/tcp    actTwo
[root@acttest act4]#
```

To get to the act_container's shell, type in: docker exec -it act_container sh. You can see this in the example below, here I did a ls once inside the container's shell. To leave the shell, type in exit. This will bring you back to the host OS.

```
[root@acttest act4]# docker exec -it act_container sh
sh-4.2# ls
anaconda-post.log  boot  etc   lib    media  opt   root  sbin  sys  usr
bin                dev   home  lib64  mnt    proc  run   srv   tmp  var
sh-4.2# exit
exit
[root@acttest act4]#
```