**Running VDJbase under Docker**

Requirements:

* A Unix host with at least 2GB of RAM, plus swap (which is used only when building containers)  
  Some R and Python packages will not build without some swap. If containers will not build, RAM is likely to be the problem.
* Up-to-date versions of Docker, Docker Compose and git.

A suitable virtual server can be obtained from Digital Ocean for $10/month as follows:

* Select ‘Create Droplets’
* Click on ‘Marketplace’ under ‘Choose an image’ and select the Docker image
* Select ‘Basic’ under ‘Choose a plan’ and select $10/mo 2GB/50GB/2TB
* Press Create Droplet

Docker conservatively stores many built images and containers. If you build repeatedly you will eventually run out of space. To reclaim space, use the commands

docker image prune

and

docker container prune

Steps:

1. Set the swap:

dd if=/dev/zero of=/var/swap.1 bs=1M count=4024

mkswap /var/swap.1

chmod 0600 /var/swap.1

swapon /var/swap.1

add the following line to /etc/fstab:

/var/swap.1 swap swap noauto 0 0

Once the swap is configured, you can turn it on with the line

swapon /var/swap.1

And off with the line

swapoff -a

You should only turn it on when building containers, and make sure that it is off otherwise. If it is on during normal operation, the website will become unresponsive.

1. Clone the docker config:

git clone <https://github.com/williamdlees/digby_docker>

1. By default, the configuration establishes a secure site using https. LetsEncrypt is used for ssl certificates. This setup requires that your server is reachable from the Internet on ports 80 and 443 and has a valid DNS name that matches the name given in config/nginx/nginx/digby. If you do not wish to use https, there are two files in containers/no\_ssl that must be copied over their counterparts:
   * cp digby\_docker/containers/no\_ssl/digby digby\_docker/config/nginx/nginx/.
   * cp digby\_docker/containers/no\_ssl/docker\_compose.yml digby\_docker/containers/.
2. Review the configuration:

cd digby\_docker/containers/my\_flask/app

cp sample\_secret.cfg secret.cfg

Edit secret.cfg. Put your host address or url in place of the existing ip address in the lines at the end.

Review digby\_docker/docker-compose.yml:

* + The file is written on the assumption that the absolute path to the repo is /root/digby\_docker. If necessary, change the volume mappings to match its location in your installation.
  + Put your email address in place of mine, so that you get warnings if SSL certificate renewal runs into trouble.

1. Enable external access

Do anything that’s necessary to let ports 80 and 443 of your server be accessed. With a Digital Island droplet, you have to enable the ports in ufw:

ufw allow 80/tcp

ufw allow 443/tcp

1. Build the custom images:

cd ../my\_R

docker build --no-cache -t my\_r .  
cd ..  
docker build --no-cache -t my\_flask .

cd ..

docker-compose build

1. Turn swap off (this is very important, the system will thrash if it is left on)

swapoff -a

1. Bring the containers up:

First bring them up in the foreground:

docker-compose up

Check the output carefully for errors: in particular look for stack traces in the flask output. Assuming that everything is ok, Ctrl-C and bring the containers up in the background:

docker-compose up -d

1. Configure the backend admin account and build the genomic data

Browse to http://<host>/admin/create\_user (where <host> is the name or ip address of your server). This will create the admin account. Then browse to http://<host>/admin and log in to the backend as [admin@vdjbase.org](mailto:admin@vdjbase.org), password admin.

Click Admin. Click on User, then Create, and create a new account with the admin role, in your own name with a secure password. Make sure to mark the account as active, and enter today’s date in the ‘confirmed’ field. Save this new account, go ‘Back to OGRE’ and logout. Login with the new account. Now go to Admin again and delete the insecure ‘admin@vdjbase.org’ account.

Browse to http://<host>/admin/update\_genomic, which will build the genomic data.

1. Using the system

The system should now be available as follows:

http://<host>/ - front end  
http://<host>/admin - back-end maintenance login  
http://<host>/admin/api - api

The exports directory in the backend server is mapped to your host. Assuming the default configuration in docker-compose.yml, you will find it at /root/digby\_docker/config/flask/exports. You can take in new RepSeq pipeline data by putting it there and following the process outlined in *Running the VDJbase backend under PyCharm*, The study\_data directories are exposed at /root/digby\_docker/study\_data.

1. Updating the system

To update, stop the containers:

cd digby\_docker/containers

docker-compose down

Pull digby\_docker to refresh the Angular client image (this is stored in digby\_docker/config/nginx/www):

git pull

rebuild my\_flask as in step 4. This will pull the contents of the digby\_backend repo. My\_r only needs to be rebuilt first if the required R libraries have changed:

cd my\_flask

docker build --no-cache -t my\_flask .

restart the containers:

docker-compose up -d

1. Troubleshooting

The flask logs are at digby\_docker/config/flask/log  
The nginx logs are at digby\_docker/config/nginx/log/nginx  
The container logs can be viewed by executing docker-compose logs

You must be in the directory digby\_docker/config to issue docker-compose commands.

1. Managing datasets

Please refer to the document ‘Running the VDJbase backend under Pycharm’ in the digby\_backend repo.