

Build and Deployment - First time

This document describes the process of building and deploying the application for the first time. It has been tested on a linux machine, specifically on Void Linux, however any modern linux distro should work such as Ubuntu and it might work in Windows with some difficulty. Some common problems and solutions are listed in the troubleshooting section.

Prerequisites

- Completed “Deployment platform setup”
- Docker account
- Access to the project repository
- git
- ssh
- docker

Building

1. Clone the repo

Run the following in a terminal:

```
git clone https://gitlab.ecs.vuw.ac.nz/ENGR301-302-2018/  
Project-14/Figshare-Data-Visualizer
```

This copies the latest version of the source code to a directory called `Figshare-Data-Visualizer`

2. Run `cd Figshare-Data-Visualizer`

3. Run `cd deployment`

4. Run `./build`

This builds the source code into runnable images. The output images should be in `Figshare-Data-Visualizer/deployment/images`.

Transferring images

5. Run `./transfer_images "path to private key" ec2-user@"ip address of server"`

This transfer the images and scripts onto the server. The path to the private key is simply the path to the private key downloaded in the “Deployment platform setup”. The ip address of server can be found from the AWS console if you followed the “Deployment platform setup”.

Setting up server

6. Run `ssh -i "path to private key" ec2-user@"ip address of server"`

This logs you into the server and allows you to execute commands on it. The following commands are all to be executed on the server.

7. Run `sudo yum update -y`

This updates the software on the server to the latest version. Preventing potential security issues.

8. Run `sudo yum install -y docker`

This installs docker, the tool that allows the server to run the images.

9. Run `sudo service docker start`

This starts docker.

10. Run `sudo usermod -a -G docker ec2-user`

This gives you permission to use docker

11. Run `exit` and then `ssh -i "path to private key" ec2-user@"ip address of server"`

For step 10 to take effect, you need to logout and log back in.

12. Install docker-compose

1. Goto <https://github.com/docker/compose/releases>
2. Look at the most recent release and run the installation commands For example it should look like

```
curl -L https://github.com/docker/compose/releases/download  
/1.23.0-rc3/docker-compose-`uname -s`-`uname -m`  
-o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

13. Run ./load_images

This loads the images into docker

14. Run ./deploy

This deploys the web server

15. Done, you can now logout by running exit

16. Check if its been successful by navigating to the website in your browser.

1. Enter the ip address of the server into the browser.
2. You should see an empty list of datasets
3. Enter the ip address of the server with /api/datasets.json
4. e.g. <http://htsdb.ecs.vuw.ac.nz/api/datasets.json>
5. There should be no errors, it might respond with nothing though.

If the above two don't work see the troubleshooting section.

Troubleshooting

Unprotected private key

To fix this run `chmod 600 "path to private key"`

Error Permission Warning: denied (step 12)

To fix this simply prepend the installation commands with `sudo` e.g.

```
sudo curl -L https://github.com/docker/compose/releases
/download/1.23.0-rc3/docker-compose-`uname -s`-`uname -m`
-o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

Get <https://registry-1.docker.io/v2/library/node/manifests/latest>: unauthorized: incorrect username or password

This is most likely an issue with your docker login. Make sure you are logging in with your docker id and not email. Your docker id can be found on the docker website. To login again you can run `docker login`. Alternatively, try logging out and logging back in. You can logout by running `docker logout`

-bash: "filename": /bin/bash^M: bad interpreter: no such file or directory

This issue is to do with windows line endings being incompatible with unix. To fix this simply run the following to convert the file to unix line endings.

```
sudo yum install -y dos2unix
```

```
dos2unix "filename"
```

The authenticity of host can't be established

This is to be expected since this is the first time you are ssh-ing into the server.

Site not working (step 16)

To fix this try redeploying. This can be done by simply running two commands

```
ssh -i "path to private key" ec2-user@"ip address of server"
```

```
./deploy
```

It might take one or two redeployments to fix it. If it continues not to work, error logs can be obtained by running:

```
ssh -i "path to private key" ec2-user@"ip address of server"
```

```
docker-compose down
```

```
export SECRET_KEY_BASE=${openssl rand -base64 32}
```

```
docker-compose up
```

This will attach the process and its output to your terminal. Type Ctrl-C to terminate once finished. You might need to access the problem webpage to kick off the error.