DTaaS on Single Vagrant Machine

This directory contains code for running DTaaS application inside one vagrant VM. The setup requires a machine which can spare 16GB RAM, 8 vCPUs and 50GB Hard Disk space to the vagrant box.

A dummy **foo.com** URL has been used for illustration. Please change this to your unique website URL.

Please follow these steps to make this work in your local environment.

1. Create **dtaas** Vagrant box. Copy *vagrant* SSH private key here. This shall be useful for logging into the vagrant machine create for single-machine deployment. You would have created an SSH key pair - *vagrant* and *vagrant.pub*. The *vagrant* is the private SSH key and is needed for the next steps.
2. Update the **Vagrantfile**. Fields to update are:
   1. Hostname (`node.vm.hostname = "foo.com"`)
   2. MAC address (`:mac => "xxxxxxxx"`). This change is required if you have a DHCP server assigning domain names based on MAC address. Otherwise, you can leave this field unchanged.
   3. Other adjustments are optional.
   4. Copy *vagrant* SSH private key into the current directory (`deploy/vagrant/single-machine`).
3. Execute the following commands from terminal

```
vagrant up
vagrant ssh
```

The Traefik gateway configuration file will be at `/home/vagrant/DTaaS/servers/config/gateway/dynamic/fileConfig.yml`. Update it as per instructions in this README.

Change the React website configuration in *client/build/env.js*.

```
window.env = {
  REACT_APP_ENVIRONMENT: 'development',
  REACT_APP_URL_LIB:
'http://foo.com/user1/shared/filebrowser/files/workspace/?token=admin',
  REACT_APP_URL_DT: 'http://foo.com/user1/lab',
  REACT_APP_URL_WORKBENCH: 'http://foo.com/user1',
};
```

Serve the react website. From inside the vagrant machine,

```
cd ~/DTaaS/client
nohup serve -s build -l 4000 & disown
```

Now you should be able to access the DTaaS application at: *http://foo.com*