This doc serves as a user manual and a description on steps to run a demo **using three computers.**

NOTE: Since the computers use local node servers now for transmitting data it may not be possible to test this without three computers or three VM’s linked together with a software defined router of some sort.

There are four roles to set up a demo.

Computer 0- service provider: we may consider to be hosted by commercial provider. For now, we run the server ourselves.

Computer 1- user: who has task to be computed and upload his python script for tensorflow

Computer 2- provider: who has resource to compute, download the script and upload the result

Computer 3- validator: who validate the result, download from provider and submit T/F.

NOTE: role of provider and validator may change back and forth, but you cannot validate your own result.

1. How to run a service provider (computer 0) and set up the testing environment.

1. Download code
   1. Begin by cloning the repository on github. This can be found at:

**https://github.com/taoluwork/BCAI.git**

* 1. or within a command line use the command:

**$ git clone $ npm install**

1. install packages
   1. once the files have been cloned navigate to **./bcai\_deploy/**
   2. once in this folder execute: **$ npm install**

NOTE: this may require sudo permission.

If this fails delete the node\_modules folder and package-lock.json and retry

* 1. then execute: **$ truffle migrate –reset --network ropsten**

this re uploads the contract to the ropsten ethereum network

* 1. then navigate to client folder and execute: **$ npm install**

NOTE: if this fails delete the node\_modules folder and package-lock.json and retry

1. build the website

**$ npm run build** OR $ **sudo npm run-script build**

1. serve the website
   1. to serve the web page execute: **$ serve -s build**
   2. Note: the address that is marked “on your network” is where the other three computers will navigate to in the browser

2. How to do a tensorflow training on user node (computer 1).

1. Download code
   1. Begin by cloning the repository on github. This can be found at:

**https://github.com/taoluwork/BCAI.git**

* 1. or within a command line use the command:

**$ git clone https://github.com/taoluwork/BCAI.git**

1. install packages
   1. navigate to **/BCAI/ML/LocalUser**
   2. execute: **$ npm install**

if this fails delete the node\_modules folder and package-lock.json and retry

1. begin server

from /gitLab/ML/LocalUser execute: **$ node localEnv.js (-4, -6, or leave blank)**

* + 1. example execution: **$ node localEnv.js** 
       1. If -4 is used then you are stating that your public IP is in IPV4
       2. If -6 is used then you are stating that your public IP is in IPV6
       3. If left blank then the default is left in IPV4
    2. this should be left running until the demo has been finished

if you cannot run node, refer to the following links to install the latest node:

<https://nodesource.com/blog/installing-node-js-tutorial-using-nvm-on-mac-os-x-and->[ubuntu/](https://nodesource.com/blog/installing-node-js-tutorial-using-nvm-on-mac-os-x-and-ubuntu/)

<https://nodejs.org/en/download/>

NOTE: install node via nvm is also recommended.

1. browser and metamask
   1. open either google chrome or firefox
   2. install the addon using the website: <https://metamask.io/>
   3. create a user as prompted to
   4. Once your account has been set up create an account, and click the network drop down and select ropsten network
      1. this is the network that we will be testing in
   5. then select the deposit button
   6. then select from test network sink
   7. once the webpage has been redirected to the test network faucet select request 1 ether from faucet a few times so that your account is not empty
   8. then wait for the block chain to validate the block and award you the test coins
   9. navigate to the address that computer is serving the webpage to and allow for metamask to connect to the webpage e.g.: 130.39.223.54:5000
2. submit task (an example task (data.zip) is included in path: /BCAI/ML/examples)
   1. From the page enter your time, target percentage, and price you are willing to pay
   2. Then select browse and from the file manager search for data.zip that is located in: /gitLab/ML/files
   3. select submit task and confirm on the popup
3. tracking and get result
   1. by clicking check status and refresh you are able to check that status of your task and see how many users are connected respectively

3. How to run a provider / validator node (computers 2 and 3).

1. Download code
   1. Begin by cloning the repository on github. This can be found at:

**https://github.com/taoluwork/BCAI.git**

* 1. or within a command line use the command:

**$ git clone https://github.com/taoluwork/BCAI.git**

1. install packages
   1. navigate to **/BCAI/ML/LocalUser**
   2. execute: **$ npm install**

if this fails delete the node\_modules folder and package-lock.json and retry

1. docker
   1. download and install docker using the steps found on the webpage: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>
   2. refer to the steps-to-install file in Docs.
2. begin server
   1. from /gitLab/ML/LocalUser execute: $ node localEnv.js (-4, -6, or leave blank)
      1. example execution: $ node localEnv.js
         1. If -4 is used then you are stating that your public IP is in IPV4
         2. If -6 is used then you are stating that your public IP is in IPV6
         3. If left blank then the default is left in IPV4
      2. this should be left running until the demo has been finished
3. browser and metamask
   1. open either google chrome or firefox
   2. install the addon using the website: <https://metamask.io/>
   3. create a user as prompted to
   4. Once your account has been set up create an account, and click the network drop down and select ropsten network
      1. this is the network that we will be testing in
   5. then select the deposit button
   6. then select from test network sink
   7. once the webpage has been redirected to the test network faucet select request 1 ether from faucet a few times so that your account is not empty
   8. then wait for the block chain to validate the block and award you the test coins
   9. navigate to the address that computer is serving the webpage to and allow for metamask to connect to the webpage e.g.: 130.39.223.54:5000
4. fetch from provider
   1. once at the web page select the user mode button to switch to worker mode
   2. enter your time, target percentage, and price
   3. then select apply provider then confirm in the popup
5. run the script and waif for the resulting
   1. once you have been assigned as a provider or validator select check status
   2. the dataID or resultID will appear depending on if you are the provider or validator respectively
   3. then select the downlaod button
   4. as long as the server is connected it will execute the code itself
      1. it may ask for your password so that it may execute the docker code in sudo mode
      2. this is a default for docker
6. submit T/F result.
   1. As a provider once the server is done executing the code, a folder called result.zip will appear in the /gitLab/ML/LocalUser folder. Upload this through the browse button in the web page and select submit result
   2. As a validator once the server is done executing the code, a file called fin.txt will appear in the /gitLab/ML/LocalUser folder. Open the txt and if the value is true select the true button and if false select the false button on the webpage.

* Notes:
  + When uploading files make sure to put them in zip files first.
  + if you wish to change the role of a given computer end the localEnv.js server and restart it with the proper tag in as the command line argument
  + if the web page does not load properly please refresh
  + when switching between accounts within meta mask, change it in the browser drop down then either click the switch mode button twice or reload the page to allow for the web page to focus on the proper account.
  + If The downloaded zip folder is broken, please delete the broken version and re-download