Local Demo Installation Instructions

To run a local installation of the demo you need to also install the common libraries from the common directory first. There are also a number of pre-requsites, as detailed below.

The Demo contains a React Web UI (Client) with an API to back it, which shows an overview of all the entities interacting.

Pre-Requisites

Git installed. Check version in Terminal >git --version
NPM & Node. Node +v10 (tested with v14.15.4). Check version in Terminal >node -v

Set up AWS Account

You need to have a Amazon Web Services Account set up which has Access Keys set up to create dynamoDB (database) and S3 (storage).

Steps:
- Set up an free basic support AWS account at https://aws.amazon.com/
- Sign into the Management console at https://signin.aws.amazon.com/
- (optional) change region to London (eu-west-2)
- Create S3 Bucket
-- Search for S3 Services. Create bucket from button. Give a unique name and save (use default settings)
- Create Access Key & Secret for S3 and DynamoDB access
-- Navigate to My Security Credentials - Access Keys - 'Create New Access Key' (this opens IAM)
-- Click on Show Access Key and save Access Key ID and Secret Access Key in a safe place (you will need them below for S3Connection STORAGE-AWS-ACCESS-KEY-ID & STORAGE-AWS-SECRET-ACCESS-KEY) & dynamoDbConnection (DB-AWS-ACCESS-KEY-ID & DB-AWS-SECRET-ACCESS-KEY)
- Make a note of the Region your account is using. It is shown in the browser adress bar (should be region=eu-west-2) You will need this below for the DB-AWS-REGION & STORAGE-AWS-REGION.

Clone Repo & Install Common and API

Steps:
- In Terminal naviate to where you want the repository to be cloned:
> git clone --single-branch --branch develop
https://github.com/iotaledger/poc-p2p-energy.git
> cd poc-p2p-energy
> cd common
> npm install
> npm run build
> cd ../demo/api
> npm install

Create an IOTA Wallet & obtain Tokens

After installing the NPM modules for the API you first need to set up an IOTA Wallet and obtain tokens before running and initiating the API. Steps:
- Generate a wallet seed (for help see https://iota.guide/article/how-to-generate-iota-wallet-seed/). On a mac in Terminal >cat /dev/urandom |LC_ALL=C tr -dc 'A-Z9' | fold -w 81 | head -n 1.
- Make a note of the 81 letter WALLET_SEED created.
- Generate an address from the wallet seed where token can be sent
-- Create a generate-address.js file located in '/demo/api/generate-address.js`, as below
--- note: replace WALLET_SEED with the 81 letter WALLET_SEED created above

```
////////////////////////////////////////////////
// Generate an address
////////////////////////////////////////////////

const Iota = require('@iota/core');

// Connect to a node
const iota = Iota.composeAPI({
    provider: 'https://nodes.devnet.thetangle.org:443'
});

// Define the security level of the address
```

```
const securityLevel = 2;

// The seed that will be used to generate an address
const seed =
    'WALLET_SEED';


// If this address is spent, this method returns the next unspent address with the lowest index
iota.getNewAddress(seed, { index: 0, securityLevel: securityLevel, total: 1 })
    .then(address => {
        console.log('Your address is: ' + address);
    })
    .catch(err => {
        console.log(err)
    });

////////////////////////////////////////////////

-- In Terminal:
> node generate-address.js
Make a note of the 81 letter WALLET_ADDRESS created.

- Send Devnet tokens to the WALLET_ADDRESS
-- Navigate to 'https://faucet.devnet.iota.org/' in browser, paste in your WALLET_ADDRESS to obtain tokens.
-- After a short while click on the 'Check Balance' link to see your devnet tokens have been sent to your WALLET_ADDRESS
```

Create API Module local.json file

```
- Create a config file in the API module located in '/demo/api/src/data/config.local.json'
- Fill in the details you obtained from the AWS pre-requisites and the IOTA Wallet above.

////////////////////////////////////////////////

{
        "nodes": [
```

```
            {
                "provider": "https://altnodes.devnet.iota.org:443",
                "depth": 3,
                "mwm": 9
            },
            {
                "provider": "https://nodes.devnet.iota.org:443",
                "depth": 3,
                "mwm": 9
            }
        ],
        "dynamoDbConnection": {
            "region": "DB-AWS-REGION",
            "accessKeyId": "DB-AWS-ACCESS-KEY-ID",
            "secretAccessKey": "DB-AWS-SECRET-ACCESS-KEY",
            "dbTablePrefix": "p2p-energy-demo-local-"
        },
        "s3Connection": {
            "region": "STORAGE-AWS-REGION",
            "accessKeyId": "STORAGE-AWS-ACCESS-KEY-ID",
            "secretAccessKey": "STORAGE-AWS-SECRET-ACCESS-KEY",
            "bucketPrefix": "p2p-energy-demo-local-"
        },
        "walletSeed": "WALLET-SEED"
}
```

//

Run & Initialize API

Back in Terminal (still within ./demo/api):
> npm run build

You should then be able to start it with ">npm run start" and open "http://localhost:4000" in the browser to see the version page.

If that is OK you can then initialise all the DB and S3 tables/containers by opening "http://localhost:4000/init" in the browser. You will get a response in terminal and in

browser to confirm that the initalisation worked. Check for errors.

Note: If there are problems with your Keys and you get errors. Update the config.local.json file, save. Then repeat from > npm run build

Install & Run Client

Open a new Terminal:
> cd demo/client

> npm install

Create Client Module local.json file

Create a config file in the Client module located in '/demo/client/public/data/client.local.json`, as below (no changes needed)

//

```
{
	"nodes": [
		{
			"provider": "https://altnodes.devnet.iota.org:443",
			"depth": 3,
			"mwm": 9
		},
		{
			"provider": "https://nodes.devnet.iota.org:443",
			"depth": 3,
			"mwm": 9
		}
	],
	"apiEndpoint": "http://localhost:4000/",
	"tangleExplorer": {
		"transactions":
"https://explorer.iota.org/devnet/transaction/:transactionHash/devnet",
		"bundles": "https://explorer.iota.org/devnet/bundle/:bundleHash/devnet",
		"mam": "https://explorer.iota.org/devnet/mam/:root/:mode/:key/devnet"
```

```
    }
}
```

//

Back in Terminal:
> npm run build
> npm run start

The Client should now be running on http://localhost:3000/ - Open in Browser

NOT USED

- Create DynamoDB Access Key & Secret (as per: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/authentication-and-access-control.html)
-- Click on Users within IAM, Add user (User name e.g. testuser, tick Programmatic Access)
-- Cleck Next (permissions). Tick AdministratorAccess can call Group Name Admin. Click Next, next to Create user.
-- Click Secret Access Key and make a note of the Keys (you will need them below for DynamoDB DB-AWS-ACCESS-KEY-ID & DB-AWS-SECRET-ACCESS-KEY)

Running Local Demo once Installed

Starting up Demo

TerminaL:
> cd demo/api
> npm run start
Run "http://localhost:4000/init" from Browser to initilise
New terminal window:
> cd demo/client
> npm run start

Loading & Configuring Grid

From client (http://localhost:3000) in Browser:
- Click on Grid (http://localhost:3000/grid), provide a Grid name and click Create.

- Click on Configure
-- Add producer including 2 sources
-- Add 2 consumers