

10/22

<http://www.techtonet.com/how-to-install-and-execute-truffle-on-an-ubuntu-16-04/>

このページを参考に Truffle の環境を構築した。

Truffle は DApp 開発にとっても便利なものであるらしく今回導入をしてみることにした、現状の開発の進捗としては、Javascript が何となく読めるようになったことと、それによって Web3.js の導入方法に関してもなんとなくではあるが理解ができ、Web3.js が行っていることはシンプルに Geth コンソールを Web 上からコマンド操作するというものであることがわかった。
とりあえず、今回は Ubuntu の方に Truffle を入れていく。(理由としては、Windows ではなく Mac での導入方法ばかりであり、やはり開発は Mac で行うのがいいのでは? と思い知らされたためである。)

なにがどのようなことを行っているのかはさっぱりわからないが、とりあえずサイトの通りにコマンドを入力していく。

```
# sudo apt-get update && sudo apt-get -y upgrade
# sudo apt-get -y install curl git vim build-essential
```

これで Package をインストールしているらしい。
相変わらず、この辺の導入部分のコマンドの意味に関しては全くわからない...

今回は勉強も兼ねて調べてみました。

Sudo .. 意味としては「指定したコマンドを別のユーザの権限で実行する」というものらしい。「substitute user do」の略や「superuser do」の略とも言われている。Ubuntu では root 権限でデスクトップ操作を行う方法が存在しないので管理者権限が必要な作業は sudo を用いて実施しなければならない。そのため、sudo は管理者権限で実行というものになる。

E ... 現在の環境変数を保持してコマンドを実行する。

Apt-get ... 基本的に様々なインストールなどに用いられていることが多い気がするコマンド。コマンドの意味に関してもそこまで外れていないように見える。この apt-get コマンドは基本的に単体では用いず、その後続けるコマンドによってその動作を決めている。

Install ... パッケージのインストール及び更新

update ... パッケージリストの更新

upgrade ... インストールされているパッケージの更新

autoremove ... 使われてないパッケージの削除

-y ... 問い合わせに関して全て y と応える(これを入れないと何回も承認しなくてはならないのか...)

Curl ... URL を使用してデータを転送するためのコマンドラインツール。ライブラリ「libCurl」をベースにしてドサシ、多くのプロトコルに対応している。基本的にはコマンドラインまたはスクリプトでデータを転送できる。何千ものソフトウェア・アプリケーションのインターネット転送バックボーンとして利用されている。「メールリスト投稿数」「ソースコードコミット頻度」「リリース頻度」などでの活発な開発が行われている。

参考 URL https://www.ossnews.jp/oss_info/Curl

build-essential ... 調べても調べても詳細が全く出てこないが、とりあえずパッケージの集合体をインストールするために必要なのが build-essential というものであるらしい。一応その中に含まれているパッケージとしては次のようなものが挙げられるようである。(これを今回の環境構築のどこに使用するのはわからないままである。)
「binutils、dpkg-dev、g++、g++-4.2、gcc、gcc-4.2、ibc6-dev、libgomp1、libstdc++6-4.2-dev、libtimedate-perl、linux-libc-dev、make、patch」流石にこれらを一一つ意味調べしていくのは心が折れるので割愛。
見たところ、C や C++ の環境がインストールされているような? 実際、調べてみると C++ の環境構築に build-essential を用いていることが多かった。おそらく、これに含まれているいくつかのパッケージの中身を今回のサイトでは使用しているのであろう。

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

最初に curl と入っていることからきくと curl コマンドであろう。先ほどいった curl を使用して sudo を実行しているのだと推測できる。

Curl URL ... URL をパラメータにしてそのコンテンツを標準出力させるのが基本的な使い方。テキストなサイトの URL をコピーしてこのコマンドを実行してみたらそのページの HTML や Javascript など

のソース情報が出力された。Chrome のデベロッパーツールのような役割を担うことができるようである。

- s ... ダウンロードの進捗とエラーメッセージを表示しない。標準のエラー出力に進捗を表示する。これを防止するために S を混ぜることでエラーメッセージは表示してくれるようになる。
- L ... デフォルトの状態ではリダイレクト設定されている URL にアクセスした時に curl はリダイレクト先の URL に対してリクエストの発行をしない。そのため、リダイレクトされている場合でもデータが欲しい場合にはこのオプションを指定すればよい。ある程度の信頼がおける場所へのリクエストであれば毎回つけてもいいのでは?とされている。

```
# sudo apt-get install -y nodejs
```

node.js ... スケーラブルなネットワークアプリケーションを構築するために設計された非同期型のイベント駆動の Javascript 環境。サーバーサイドの Javascript とも言われている。node.js はライブラリではなく Javascript のアプリケーションプラットフォーム。クライアントもサーバーも同じ言語で記述できるようになることで利便性が向上している。参考 URL: <https://eng-entrance.com/what-is-nodejs>

```
# sudo npm install -g express
```

これは npm という node.js のコマンドを使用して express を導入しようとしている。これによって導入される express の概要は以下のようになっている。

Express は node.js において王道的な開発をスピードアップするためのフレームワーク。具体的な開発効率向上例として REST API が存在する。

例：写真共有サービス

写真共有サービスでとある写真を返す REST API を提供するとき、そのリクエストをサーバー側は下記のように内容を解析する必要がある。

- ・このリクエストは写真を 1 枚要求している
- ・リクエストしている写真の ID は YOUR_PHOTO_ID

これを正規表現などでパターンマッチングすることで URI を解析することも可能であるが Express では以下のようなシンプルなコードで書くことができる。

```
// 任意の写真を提供する API
```

```
app.get("/api/photo/:photoId", function(req, res, next){  
  // req.params.photoId でユーザーが指定した写真 ID を識別できる  
  console.log(req.params.photoId);  
});
```

正規表現の URI 解析はかなりのコード量を必要とするのでこれだけの記述で書くことができるのはかなり魅力的である。

このコマンドの実行結果を以下に貼り付ける。

正直見るに耐えないものになるのもう一度環境を構築するということがない限りは開くことはないと思われる。

```
sudo npm install -g express
```

```
/usr/lib
```

```
├── express@4.16.4  
  ├── accepts@1.3.5  
  │   ├── mime-types@2.1.21  
  │   │   └── mime-db@1.37.0  
  │   └── negotiator@0.6.1  
  ├── array-flatten@1.1.1  
  ├── body-parser@1.18.3  
  │   ├── bytes@3.0.0  
  │   ├── http-errors@1.6.3  
  │   │   └── inherits@2.0.3  
  │   ├── iconv-lite@0.4.23  
  │   └── safer-buffer@2.1.2  
  ├── raw-body@2.3.3  
  ├── content-disposition@0.5.2  
  └── content-type@1.0.4
```

```

├── cookie@0.3.1
├── cookie-signature@1.0.6
├── debug@2.6.9
│   └── ms@2.0.0
├── depd@1.1.2
├── encodeurl@1.0.2
├── escape-html@1.0.3
├── etag@1.8.1
├── finalhandler@1.1.1
│   └── unpipe@1.0.0
├── fresh@0.5.2
├── merge-descriptors@1.0.1
├── methods@1.1.2
├── on-finished@2.3.0
│   └── ee-first@1.1.1
├── parseurl@1.3.2
├── path-to-regexp@0.1.7
├── proxy-addr@2.0.4
│   ├── forwarded@0.1.2
│   └── ipaddr.js@1.8.0
├── qs@6.5.2
├── range-parser@1.2.0
├── safe-buffer@5.1.2
├── send@0.16.2
│   ├── destroy@1.0.4
│   └── mime@1.4.1
├── serve-static@1.13.2
├── setprototypeof@1.1.0
├── statuses@1.4.0
├── type-is@1.6.16
│   └── media-typer@0.3.0
├── utils-merge@1.0.1
└── vary@1.1.2

```

このようなファイル階層によって保存されているのだろうということがわかる。

\$ sudo npm install -g ethereumjs-testrpc を実行すると以下のような出力が得られた。

npm WARN deprecated ethereumjs-testrpc@6.0.3: ethereumjs-testrpc has been renamed to ganache-cli, please use this package from now on.

/usr/bin/testrpc -> /usr/lib/node_modules/ethereumjs-testrpc/build/cli.node.js

> uglifyjs-webpack-plugin@0.4.6 postinstall /usr/lib/node_modules/ethereumjs-testrpc/node_modules/uglifyjs-webpack-plugin

> node lib/post_install.js

/usr/lib

```

├── ethereumjs-testrpc@6.0.3
│   └── webpack@3.12.0
│       ├── acorn@5.7.3
│       ├── acorn-dynamic-import@2.0.2
│       │   └── acorn@4.0.13
│       └── ajv@6.5.4

```

- └─ fast-deep-equal@2.0.1
- └─ fast-json-stable-stringify@2.0.0
- └─ json-schema-traverse@0.4.1
- └─ uri-js@4.2.2
- └─ punycode@2.1.1
- └─ ajv-keywords@3.2.0
- └─ async@2.6.1
- └─ lodash@4.17.11
- └─ enhanced-resolve@3.4.1
- └─ graceful-fs@4.1.11
- └─ object-assign@4.1.1
- └─ escape@3.6.0
- └─ es6-map@0.1.5
- └─ d@1.0.0
- └─ es5-ext@0.10.46
- └─ next-tick@1.0.0
- └─ es6-iterator@2.0.3
- └─ es6-set@0.1.5
- └─ es6-symbol@3.1.1
- └─ event-emitter@0.3.5
- └─ es6-weak-map@2.0.2
- └─ esrecurse@4.2.1
- └─ estraverse@4.2.0
- └─ interpret@1.1.0
- └─ json-loader@0.5.7
- └─ json5@0.5.1
- └─ loader-runner@2.3.1
- └─ loader-utils@1.1.0
- └─ big.js@3.2.0
- └─ emojis-list@2.1.0
- └─ memory-fs@0.4.1
- └─ errno@0.1.7
- └─ prr@1.0.1
- └─ readable-stream@2.3.6
- └─ core-util-is@1.0.2
- └─ inherits@2.0.3
- └─ isarray@1.0.0
- └─ process-nexttick-args@2.0.0
- └─ safe-buffer@5.1.2
- └─ util-deprecate@1.0.2
- └─ mkdirp@0.5.1
- └─ minimist@0.0.8
- └─ node-libs-browser@2.1.0
- └─ assert@1.4.1
- └─ util@0.10.3
- └─ inherits@2.0.1
- └─ browserify-zlib@0.2.0
- └─ pako@1.0.6
- └─ buffer@4.9.1
- └─ base64-js@1.3.0
- └─ ieee754@1.1.12
- └─ console-browserify@1.1.0

- └── date-now@0.1.4
- └── constants-browserify@1.0.0
- └── crypto-browserify@3.12.0
 - └── browserify-cipher@1.0.1
 - └── browserify-aes@1.2.0
 - └── buffer-xor@1.0.3
 - └── browserify-des@1.0.2
 - └── des.js@1.0.0
 - └── evp_bytestokey@1.0.3
 - └── browserify-sign@4.0.4
 - └── bn.js@4.11.8
 - └── browserify-rsa@4.0.1
 - └── elliptic@6.4.1
 - └── brorand@1.1.0
 - └── hash.js@1.1.5
 - └── hmac-drbg@1.0.1
 - └── minimalistic-assert@1.0.1
 - └── minimalistic-crypto-utils@1.0.1
 - └── parse-asn1@5.1.1
 - └── asn1.js@4.10.1
 - └── create-ecdh@4.0.3
 - └── create-hash@1.2.0
 - └── cipher-base@1.0.4
 - └── md5.js@1.3.5
 - └── hash-base@3.0.4
 - └── ripemd160@2.0.2
 - └── sha.js@2.4.11
 - └── create-hmac@1.1.7
 - └── diffie-hellman@5.0.3
 - └── miller-rabin@4.0.1
 - └── pbkdf2@3.0.17
 - └── public-encrypt@4.0.3
 - └── randombytes@2.0.6
 - └── randomfill@1.0.4
 - └── domain-browser@1.2.0
 - └── events@1.1.1
 - └── https-browserify@1.0.0
 - └── os-browserify@0.3.0
 - └── path-browserify@0.0.0
 - └── process@0.11.10
 - └── punycode@1.4.1
 - └── querystring-es3@0.2.1
 - └── stream-browserify@2.0.1
 - └── stream-http@2.8.3
 - └── builtin-status-codes@3.0.0
 - └── to-arraybuffer@1.0.1
 - └── xtend@4.0.1
 - └── string_decoder@1.1.1
 - └── timers-browserify@2.0.10
 - └── setimmediate@1.0.5
 - └── tty-browserify@0.0.0
 - └── url@0.11.0

```

├── punycode@1.3.2
├── querystring@0.2.0
├── util@0.10.4
├── vm-browserify@0.0.4
├── indexof@0.0.1
├── source-map@0.5.7
├── supports-color@4.5.0
├── has-flag@2.0.0
├── tapable@0.2.8
├── uglifyjs-webpack-plugin@0.4.6
├── uglify-js@2.8.29
├── uglify-to-browserify@1.0.2
├── yargs@3.10.0
├── camelcase@1.2.1
├── cliui@2.1.0
├── center-align@0.1.3
├── align-text@0.1.4
├── longest@1.0.1
├── lazy-cache@1.0.4
├── right-align@0.1.3
├── wordwrap@0.0.2
├── window-size@0.1.0
├── webpack@1.6.0
├── chokidar@2.0.4
├── anymatch@2.0.0
├── micromatch@3.1.10
├── arr-diff@4.0.0
├── define-property@2.0.2
├── is-descriptor@1.0.2
├── is-accessor-descriptor@1.0.0
├── is-data-descriptor@1.0.0
├── kind-of@6.0.2
├── extend-shallow@3.0.2
├── assign-symbols@1.0.0
├── is-extendable@1.0.1
├── is-plain-object@2.0.4
├── extglob@2.0.4
├── define-property@1.0.0
├── is-descriptor@1.0.2
├── is-accessor-descriptor@1.0.0
├── is-data-descriptor@1.0.0
├── kind-of@6.0.2
├── expand-brackets@2.1.4
├── define-property@0.2.5
├── extend-shallow@2.0.1
├── posix-character-classes@0.1.1
├── extend-shallow@2.0.1
├── fragment-cache@0.2.1
├── kind-of@6.0.2
├── nanomatch@1.2.13
├── is-windows@1.0.2
├── kind-of@6.0.2

```

```

├─── object.pick@1.3.0
│   └─── regex-not@1.0.2
├─── async-each@1.0.1
├─── braces@2.3.2
│   ├─── arr-flatten@1.1.0
│   ├─── array-unique@0.3.2
│   ├─── extend-shallow@2.0.1
│   │   └─── is-extendable@0.1.1
│   ├─── fill-range@4.0.0
│   │   ├─── extend-shallow@2.0.1
│   │   ├─── is-number@3.0.0
│   │   │   ├─── kind-of@3.2.2
│   │   │   └─── is-buffer@1.1.6
│   │   ├─── repeat-string@1.6.1
│   │   └─── to-regexp-range@2.1.1
│   ├─── isobject@3.0.1
│   ├─── repeat-element@1.1.3
│   ├─── snapdragon@0.8.2
│   │   ├─── base@0.11.2
│   │   │   ├─── cache-base@1.0.1
│   │   │   │   ├─── collection-visit@1.0.0
│   │   │   │   │   ├─── map-visit@1.0.0
│   │   │   │   │   └─── object-visit@1.0.1
│   │   │   ├─── get-value@2.0.6
│   │   │   ├─── has-value@1.0.0
│   │   │   │   ├─── has-values@1.0.0
│   │   │   │   └─── kind-of@4.0.0
│   │   │   ├─── set-value@2.0.0
│   │   │   │   └─── extend-shallow@2.0.1
│   │   │   ├─── to-object-path@0.3.0
│   │   │   ├─── union-value@1.0.0
│   │   │   │   ├─── set-value@0.4.3
│   │   │   │   └─── extend-shallow@2.0.1
│   │   │   ├─── unset-value@1.0.0
│   │   │   │   ├─── has-value@0.3.1
│   │   │   │   ├─── has-values@0.1.4
│   │   │   │   └─── isobject@2.1.0
│   │   ├─── class-utils@0.3.6
│   │   ├─── arr-union@3.1.0
│   │   ├─── define-property@0.2.5
│   │   │   ├─── static-extend@0.1.2
│   │   │   ├─── define-property@0.2.5
│   │   │   │   ├─── object-copy@0.1.0
│   │   │   │   ├─── copy-descriptor@0.1.1
│   │   │   │   └─── define-property@0.2.5
│   │   ├─── component-emitter@1.2.1
│   │   ├─── define-property@1.0.0
│   │   │   ├─── is-descriptor@1.0.2
│   │   │   ├─── is-accessor-descriptor@1.0.0
│   │   │   ├─── is-data-descriptor@1.0.0
│   │   │   └─── kind-of@6.0.2
│   │   ├─── mixin-deep@1.3.1

```

- for-in@1.0.2
 - is-extendable@1.0.1
- pascalcase@0.1.1
- debug@2.6.9
 - ms@2.0.0
- define-property@0.2.5
 - is-descriptor@0.1.6
 - is-accessor-descriptor@0.1.6
 - is-data-descriptor@0.1.4
 - kind-of@5.1.0
- extend-shallow@2.0.1
- map-cache@0.2.2
- source-map-resolve@0.5.2
 - atob@2.1.2
 - decode-uri-component@0.2.0
 - resolve-url@0.2.1
 - source-map-url@0.4.0
 - urix@0.1.0
- use@3.1.1
- snapdragon-node@2.1.1
 - define-property@1.0.0
 - is-descriptor@1.0.2
 - is-accessor-descriptor@1.0.0
 - is-data-descriptor@1.0.0
 - kind-of@6.0.2
 - snapdragon-util@3.0.1
- split-string@3.1.0
- to-regexp@3.0.2
 - safe-regexp@1.1.0
 - ret@0.1.15
- glob-parent@3.1.0
 - is-glob@3.1.0
 - path-dirname@1.0.2
- is-binary-path@1.0.1
 - binary-extensions@1.12.0
- is-glob@4.0.0
 - is-extglob@2.1.1
- lodash.debounce@4.0.8
- normalize-path@2.1.1
 - remove-trailing-separator@1.1.0
- path-is-absolute@1.0.1
- readdirp@2.2.1
- upath@1.1.0
- neo-async@2.6.0
- webpack-sources@1.3.0
 - source-list-map@2.0.1
 - source-map@0.6.1
- yargs@8.0.2
- camelcase@4.1.0
- cliui@3.2.0
 - string-width@1.0.2
 - code-point-at@1.1.0


```

├── is-fullwidth-code-point@1.0.0
├── number-is-nan@1.0.1
├── strip-ansi@3.0.1
│   ├── ansi-regex@2.1.1
│   └── wrap-ansi@2.1.0
├── string-width@1.0.2
├── decamelize@1.2.0
├── get-caller-file@1.0.3
├── os-locale@2.1.0
│   └── execa@0.7.0
│       ├── cross-spawn@5.1.0
│       │   ├── lru-cache@4.1.3
│       │   │   ├── pseudomap@1.0.2
│       │   │   └── yallist@2.1.2
│       │   └── shebang-command@1.2.0
│       │       └── shebang-regex@1.0.0
│       └── which@1.3.1
│           └── isexe@2.0.0
├── get-stream@3.0.0
├── is-stream@1.1.0
├── npm-run-path@2.0.2
│   └── path-key@2.0.1
├── p-finally@1.0.0
├── signal-exit@3.0.2
├── strip-eof@1.0.0
├── lcid@1.0.0
│   └── invert-kv@1.0.0
├── mem@1.1.0
│   └── mimic-fn@1.2.0
├── read-pkg-up@2.0.0
│   └── find-up@2.1.0
│       ├── locate-path@2.0.0
│       │   ├── p-locate@2.0.0
│       │   │   ├── p-limit@1.3.0
│       │   │   └── p-try@1.0.0
│       │   └── path-exists@3.0.0
│       └── read-pkg@2.0.0
│           ├── load-json-file@2.0.0
│           ├── parse-json@2.2.0
│           │   └── error-ex@1.3.2
│           │       └── is-arrayish@0.2.1
│           ├── pify@2.3.0
│           └── strip-bom@3.0.0
├── normalize-package-data@2.4.0
│   ├── hosted-git-info@2.7.1
│   ├── is-builtin-module@1.0.0
│   │   └── builtin-modules@1.1.1
│   ├── semver@5.6.0
│   ├── validate-npm-package-license@3.0.4
│   ├── spdx-correct@3.0.2
│   │   └── spdx-license-ids@3.0.1
│   └── spdx-expression-parse@3.0.0
└── fsevents@1.2.4

```

```
| |   └── spdx-exceptions@2.2.0
| |   └── path-type@2.0.0
| └── require-directory@2.1.1
| └── require-main-filename@1.0.1
| └── set-blocking@2.0.0
| └── string-width@2.1.1
|   └── is-fullwidth-code-point@2.0.0
|     └── strip-ansi@4.0.0
|       └── ansi-regex@3.0.0
| └── which-module@2.0.0
| └── y18n@3.2.1
| └── yargs-parser@7.0.0
|   └── camelcase@4.1.0
```

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.2
(node_modules/ethereumjs-testrpc/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for
fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"ia32"})
```

警告文が三個出ているがとりあえず無視することにする。先ほどのコマンドで出力されたものに比べてかなり増えている。今回のコマンドで様々な場所にファイルが格納されて今のようになっているのであろう。決して見直すことはないと思いたいがいっつ何が必要になるかわからないので記述しておく。

Testrpc
EthereumJS TestRPC v6.0.3 (ganache-core: 2.0.2)

Available Accounts

```
=====
(0) 0xc119c435c7ec704d447b732647c89900fb2e0982
(1) 0x2f521ab490bea6713cdf88888cee3485fdb1682
(2) 0xbf35e6c514b9aa2ecc35abc244fc8ab633d6c8d1
(3) 0x04654c5d9e834f972e369926cfdd21e01bf9021e
(4) 0x8619d3280d100cd6799a4d20f83337824497594a
(5) 0x2dc74222d59ea84873ee080dbdf5cc2adeb8acc0
(6) 0x00e05089d9e9066be4617d642ef495ec14ce2be4
(7) 0x9aa8cdb917fd5be8f1fa753e2ba31d046343d164
(8) 0xfb49b5b8ce784a7a5667a31f3e7743bb86791588
(9) 0x13a7ce8787c8656792a235d0f715d7da1246e94c
```

Private Keys

=====

- (0) 5a056725891010c60d542d5171dbdb2bef7729feedf1f8c2f300b2952b885f90
- (1) 0a1189ed958eb5047969da5933fbdb9087bd2d4c0001b0988b23821324a1f239
- (2) e054679345795e9ab2451da1da32a0968a92964488014bdb821823f92b94d664
- (3) 19c5e6da249bb1190dc98a65248f824d7ef720d3a6b519bf1f95abd97a8da081
- (4) 204ebfadb4e9108fe2e5d104bd4831249982bc62e3400f7ca669b792b9523534
- (5) df5b034d1c82579c1991d41e65d9c14fba908440de18427878347d410a359899
- (6) 25becf81e3ef8f581f9551f7bc00d4ef522827c3a21841fae624bd3a9262cb88
- (7) bd971cfe533d2c333a37143eb58c04c016509cf2f0e5d61beb73be5714305b61
- (8) 4382ab474221cae5def9996b282f46d046dc668f30a2f5be904f3d17e72c139e
- (9) 4e0a444c6c985ec6c6b37bca509bf1ec227c7ee967bae4d948b37c5a632ec20

HD Wallet

=====

Mnemonic: deer brain humble jealous wisdom bounce random busy castle reward shrimp girl
Base HD Path: m/44'/60'/0'/0/{account_index}

Listening on localhost:8545

このあと入力関係を一切受け付け無い状態となり、Ctrl+C で強制終了
待機して反応を待つべきであったような気がする。とりあえずもう一度実行してみて別のコンソールで確認してみようと思う。

先ほどの testrpc コマンドは testrpc ネットを構築することを目的とするコマンドであったらしい。これが正常に稼働しないとプライベートネットが構築されないため Solidity や Dapp に関する作業が行えないらしい。そのため先ほどの警告を剥がしていかないとならないので警告文をクリアしていく。

npm WARN deprecated ethereumjs-testrpc@6.0.3: ethereumjs-testrpc has been renamed to ganache-cli, please use this package from now on.

```
/usr/bin/testrpc -> /usr/lib/node_modules/ethereumjs-testrpc/build/cli.node.js  
> uglifyjs-webpack-plugin@0.4.6 postinstall /usr/lib/node_modules/ethereumjs-testrpc/node_modules/uglifyjs-webpack-plugin  
> node lib/post_install.js
```

この警告文は、「ethereumjs-testrpc」というファイルは現在「ganache-cli」というファイルに名前が書き換わっているのでもそちらを利用してくれとのこと。testrpc が起動しない理由にはならなそう。

*npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.2
(node_modules/ethereumjs-testrpc/node_modules/chokidar/node_modules/fsevents):*

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"ia32"})

調べたところ、オプションスキップによってこれらのファイルはインストールされていないので把握してくれとの警告のようではあるが、解決方法としてこのスキップしたファイルたちを手動でインストールする方法が挙げられているらしく、実行してみたが、全くダメで全ての項目においてエラーが吐かれる。

Truffle init

コマンドは問題なく通り、サイトの通りに「myproject」に truffle の関連ファイルが格納された。
しかし、別コンソールで行っている方の「testrpc」コマンドの方の通りがなんとなく悪い。

Localhost:8545

の記述から全く進行を見せず、これはすでに動いているからため、このコンソールに割かれるはずのリソースは testrpc の動作に利用しているため他のコンソールから実行してくださいという意味なのかが判断に困る。

しかし、Truffle のデプロイ?migrate というコマンドは

Error: No network specified. Cannot determine current network.

at Object.detect (/usr/lib/node_modules/truffle/build/webpack:/packages/truffle-core/lib/environment.js:31:1)

at /usr/lib/node_modules/truffle/build/webpack:/packages/truffle-core/lib/commands/migrate.js:91:1

at finished (/usr/lib/node_modules/truffle/build/webpack:/packages/truffle-workflow-compile/index.js:53:1)

at /usr/lib/node_modules/truffle/build/webpack:/packages/truffle-compile/index.js:314:1

at /usr/lib/node_modules/truffle/build/webpack:/packages/truffle-compile/profiler.js:158:1

at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:3888:1

at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:473:1

at replenish (/usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:1006:1)

at iterateeCallback (/usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:995:1)

at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:969:1

```
at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:3885:1
at /usr/lib/node_modules/truffle/build/webpack:/packages/truffle-compile/profiler.js:154:1
at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:1140:1
at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:473:1
at iteratorCallback (/usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:1064:1)
at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:969:1
at /usr/lib/node_modules/truffle/build/webpack:/~/async/dist/async.js:1137:1
at /usr/lib/node_modules/truffle/build/webpack:/packages/truffle-compile/profiler.js:133:1
at FSReqWrap.oncomplete (fs.js:123:15)
```

このエラーとともに弾かれてしまい、ネットワークが存在していないためのエラーであるということまでは理解できる。このエラーが Truffle 自体がもうすでに 32bit 環境での動作を保証していないせいで起こるものなのかはわからないが現状進行が止まってしまった。とりあえず、この Ubuntu16.04 での Truffle 導入は諦めて WindowsPC の方での環境構築を再度行ってみることにする。

幸いにも先日購入した参考書が Windows の Truffle 環境構築方法を用意してくれていたはずなのでそちらを参考にしながら環境を構築していこうと思う。

全くもって理解不能のエラーだったので対処のしようがない。
せっかく Linux コマンドの意味調べを行ったというのに...

Windows のほうで割とすんなりうまく行ったので Ubuntu 側でも実験開始。
testrpc は使用しなくても Truffle でテストネットは作成できるらしい。
なので Truffle を用いて開発が行えるのかを実験していく。

Truffle compile --all

コマンドでコンパイルが行える。このコンパイル処理を行った成果物は/build/contracts に置かれる。このファイルは Truffle の内部動作に不可欠なものであるようなので手動操作などは絶対に行ってはならないらしい。新たにコントラクトをコンパイルしデプロイすると内部ファイルの内容も変更されていくので本当に触れてはならないディレクトリであるらしい。

Truffle migrate

コマンドを実行したところエラーが出てしまった。

No network specified. Cannot determine current network.

エラーはこれで意味は「ネットワークは指定されていません。現在のネットワークを特定できません。」という意味であるらしい。前回詰まった testrpc の部分にも関係あるのだろうか？

しかし、node.js、npm はともにインストールされている状態であり、testrpc 不稼働理由もわからないままである。三種の PC を同時進行で開発環境設定を行っていたためかそのへんに対する理解が及んでいない。

develop コンソールでの migrate 処理に関しては多くのエラーはなく、
Using network 'develop'.
Network up to date.
ネットワークを使用して '開発'.
最新のネットワーク。
という説明文?のようなものであった。

そもそも migrate (マイグレーション) とは端的にはマイグレーションファイルを作成することにある。Ethereum ネットワークにスマートコントラクトをデプロイする際には、マイグレーションファイルを実行する。各々がデプロイタスクのステージングを担当していて、開発の進捗とともにプロジェクトの新たなコントラクトが追加されることを前提に書かれていてプロジェクトに新たなコントラクトを追加する際には新規でスクリプトを追加する。実行されたマイグレーションの履歴は特別なマイグレーションコントラクトによってブロックチェーン上に記録される。

公式サンプル Metacoin の実装

```
mkdir metacoin
```

```
cd metacoin
```

```
truffle unbox metacoin
```

の三種コマンドを実行

truffle unbox コマンドは Truffle フレームワークが提供するサンプルをローカル環境に展開するコマンド
サンプルは Truffle Boxes:「<http://truffleframework.com/boxes/>」に上がっているらしい

metacoin の導入後、コンパイルからマイグレーション処理を行ったところ問題なく実行可能であった。

今回、metacoin のコントラクトアドレスとなったのは

0xf25186b5081ff5ce73482ad761db0eb0d25abfbf

であった。

環境によって値は違うとは書いてあったが、すべて一致したアドレスが出現しているところを見ると、コントラクトの内容によってハッシュ処理されていると見たほうが良さそう？ならば、毎回同一の内容に対して同一のアドレスが変換されるのではないかと思う。

truffle における、m = metacoin(“0xf25186b5081ff5ce73482ad761db0eb0d25abfbf”)

のコマンドがやはり実行されない。

これは、変数mの中に metacoin 以下を格納設定し、今後 m と入力するだけで上記のコマンドが実行されるというシンボリックリンクのようなものののだが、思うように行かない。

testrpc はやはり

localhost:8545

が出現した時点でそのコンソールは使用不可になり、そのコンソールによって testrpc ネットが発生、維持されていると考えて問題ないようである。

その後、サイトにしがたって、truffle のテスト機能を用いて MetaCoin のコントラクトを実行してみたところ以下のような結果を得ることができた。

Using network 'test'.

Compiling ./contracts/ConvertLib.sol...

Compiling ./contracts/MetaCoin.sol...

Compiling ./test/TestMetacoin.sol...

Compiling truffle/Assert.sol...

Compiling truffle/DeployedAddresses.sol...

TestMetacoin

- ✓ testInitialBalanceUsingDeployedContract (110ms)

- ✓ testInitialBalanceWithNewMetaCoin (101ms)

Contract: MetaCoin

- ✓ should put 10000 MetaCoin in the first account

- ✓ should call a function that depends on a linked library (56ms)

- ✓ should send coin correctly (166ms)

5 passing (2s)

一応参考にしたサイトのとおりに出力されている。

サイトに記載されていた Genesis.json を作成する意味。

初期に記述されて格納されている Genesis.json ではパブリックネットと同様の設定のプライベートネットになってしまう。それでは高速で開発を行いたい場合には適していないので Genesis.json を書き換えることで高速なマイニングなどを可能にしている。そのため、実装したい開発環境に合わせてプライベートネットの設定を作成することができるということが Genesis.json を作成するという作業を行う理由となっている。

マイグレーション問題に関して

Truffle の開発している公式の Metacoins をインストールしたディレクトリに「truffle-config.js」というファイルが存在する。そのファイルが殻になっていたことが今回のマイグレーション問題の原因であったらしい。

testrpc 起動中のコンソールと migrate コマンドを実行したコンソールの両方が連動して動作を開始し、マイグレーション処理を実行したことが今回でわかった。

両者に出たログは以下の通り

localhost:8545 (testrpc 動作中)

net_version

eth_accounts

eth_accounts

net_version

net_version

eth_sendTransaction

eth_newBlockFilter

eth_getFilterChanges

eth_getTransactionReceipt

eth_getFilterChanges

eth_getTransactionReceipt

Transaction: 0xeb1d26cc70ed0cfaa906a4495b20d6bfc5d68b93fe977eed88d08c208b0534

Contract created: 0x8c2da6d4ca2b229410a2344c203d8f121b8de046

Gas usage: 277462

Block Number: 859

Block Time: Wed Oct 24 2018 23:02:15 GMT+0900 (JST)

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getFilterChanges

eth_getTransactionReceipt

eth_getCode

eth_uninstallFilter

eth_sendTransaction

eth_getTransactionReceipt

Transaction: 0x74d0fccf445f42e5aec7fcddfa67bcd7e1b39ebf9f88165bfbdaf168cdd9f4e

Gas usage: 42008

Block Number: 863

Block Time: Wed Oct 24 2018 23:02:19 GMT+0900 (JST)

eth_getTransactionReceipt

eth_accounts

net_version
net_version
eth_sendTransaction
eth_newBlockFilter
eth_getFilterChanges
eth_getTransactionReceipt

Transaction: 0x8c2164a3464c51dcba6a1b66bbd16f381601ddf759e03383730550cfdc5023ec
Contract created: 0x817e3312061b0e23adaf22df4518212bf16cfc91
Gas usage: 108240
Block Number: 864
Block Time: Wed Oct 24 2018 23:02:20 GMT+0900 (JST)

eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getTransactionReceipt
eth_getCode
eth_uninstallFilter
eth_sendTransaction
eth_newBlockFilter
eth_getFilterChanges
eth_getTransactionReceipt

Transaction: 0x556f14ffb225fd0b9f25cf7778a7b5acec22a69aac2714b625a162a246d509cf
Contract created: 0xc54b4f31ea5173d29f612a9cdb737ea1c439002c
Gas usage: 347432
Block Number: 867
Block Time: Wed Oct 24 2018 23:02:23 GMT+0900 (JST)

eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getFilterChanges
eth_getTransactionReceipt
eth_getCode
eth_uninstallFilter
eth_sendTransaction
eth_getTransactionReceipt

Transaction: 0x3c0fde94bc45c83703ff3edb352f8808d386f07bb4bdbaf696a7876454155e82
Gas usage: 27008
Block Number: 871
Block Time: Wed Oct 24 2018 23:02:27 GMT+0900 (JST)
eth_getTransactionReceipt

マイグレーション処理実行サイド
yuta@yuta-Vostro-230:~/metacoin\$ truffle migrate
Using network 'development'.

```
Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0xb9b1d26cc70ed0cfaa906a4495b20d6bfc5d68b93fe977eed88d08c208b0534
  Migrations: 0x8c2da6d4ca2b229410a2344c203d8f121b8de046
Saving successful migration to network...
  ... 0x74d0fccf445f42e5aec7fcddfa67bcd7e1b39ebf9f88165bfbdfaf168cdd9f4e
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying ConvertLib...
  ... 0x8c2164a3464c51dcba6a1b66bbd16f381601ddf759e03383730550cfdc5023ec
  ConvertLib: 0x817e3312061b0e23adaf22df4518212bf16cfc91
  Linking ConvertLib to MetaCoin
  Deploying MetaCoin...
  ... 0x556f14ffb225fd0b9f25cf7778a7b5acec22a69aac2714b625a162a246d509cf
  MetaCoin: 0xc54b4f31ea5173d29f612a9cdb737ea1c439002c
Saving successful migration to network...
  ... 0x3c0fde94bc45c83703ff3edb352f8808d386f07bb4bdbaf696a7876454155e82
Saving artifacts...
```

今回の実行で Metacoin のアドレス?が変更になったため環境によって値が変化することが判明した。
Windows の方で実行できなかったことに関してもこれが原因の可能性が高いのであとでやってみることにする。

先ほどから失敗が続いている変数へのコマンドの格納に関して
m = MetaCoin.at("0xc54b4f31ea5173d29f612a9cdb737ea1c439002c")
という入力をする事でmというコマンドが MetaCoin 以下のコマンドと同一のものとして扱われるというものであるらしいがそれを入力したらコンソールが停止してしまった。
またしても原因不明。とりあえず、CTL+C で落として再度入力して見たいと思う。

Windows の方では testrpc の起動の問題であるようにおもえる。インストール時点のコマンドで ethereumjs-testrpc というものを入力してインストールしているので testrpc 自体がすでに Etheruem のネットワークとして機能している。そのため、testrpc を起動している状態でないと Truffle のコマンドの特にデプロイなどの部分は機能しないということのようである。

```
M = MetaCoin.at("MetaCoin.address")
でmが MetaCoin 以下と同義であるという定義付けする。
ここで重要なのは、develop で Truffle を再構築すると過去にデプロイしたデータは消えてしまうということがわかった。
そのため、develop 後、すぐに migrate して、mに同義付を行うことで無事にコントラクトを実行することに成功した。
m.getBalance(web3.eth.accounts[0])
で Truffle と testrpc が自動生成したアカウントの MetaCoin の残高を確認した、。
{ [String: '10000'] s: 1, e: 4, c: [ 10000 ] }
```