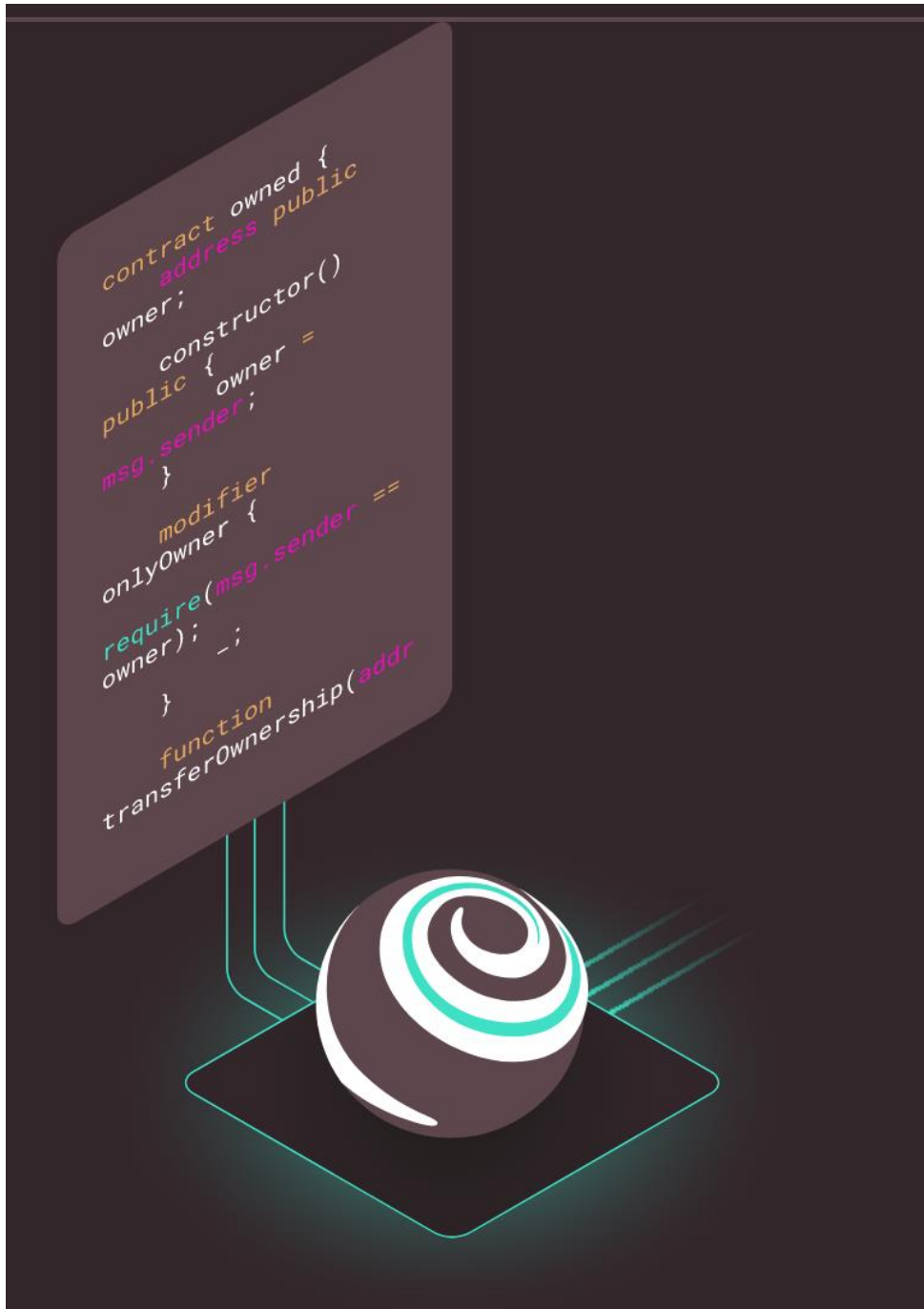
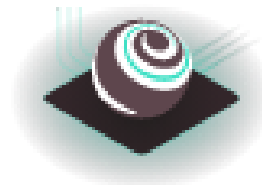


PD-7.0 Truffle



PD-7.1 Install Truffle



```
>npm install -g truffle
+ truffle@5.1.52
added 144 packages from 62 contributors in 23.3s

>truffle version
Truffle v5.1.52 (core: 5.1.52)
Solidity v0.5.16 (solc-js)
Node v14.15.0
Web3.js v1.2.9
```

<https://www.trufflesuite.com/docs/truffle/overview>

<https://github.com/trufflesuite/truffle>

<https://www.trufflesuite.com/truffle>

PD-7.1 Truffle basics, create initial project

```
>truffle init
```

```
√ Preparing to download  
√ Downloading  
√ Cleaning up temporary files  
√ Setting up box
```

```
Unbox successful. Sweet!
```

```
Commands:
```

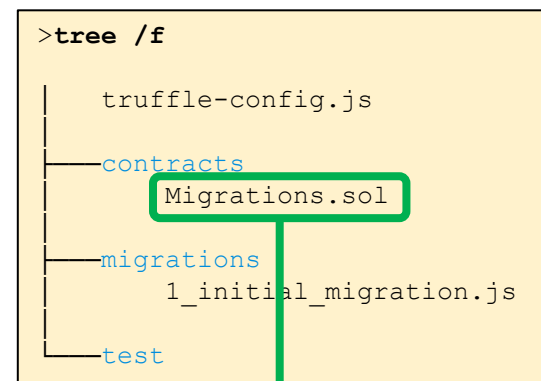
```
Compile:      truffle compile  
Migrate:      truffle migrate  
Test contracts: truffle test
```

```
>tree /f
```

```
truffle-config.js  
├── contracts  
│   └── Migrations.sol  
├── migrations  
│   └── 1_initial_migration.js  
└── test
```

PD-7.1 Internal administration Migration.sol

```
Migrations.sol x
1  pragma solidity >=0.4.21 <0.6.0;
2
3  contract Migrations {
4      address public owner;
5      uint public last_completed_migration;
6
7      constructor() public {
8          owner = msg.sender;
9      }
10
11     modifier restricted() {
12         if (msg.sender == owner) _;
13     }
14
15     function setCompleted(uint completed) public restricted {
16         last_completed_migration = completed;
17     }
18
19     function upgrade(address new_address) public restricted {
20         Migrations upgraded = Migrations(new_address);
21         upgraded.setCompleted(last_completed_migration);
22     }
23 }
```



The Migration.sol contract keeps track of which migrations were done on the current network.

PD-7.1 Example solidity contract HelloWorld.sol

```
HelloWorld.sol x
1  pragma solidity ^0.5.12;
2
3  contract HelloWorld {
4      string public welcome = "Hello World!";
5  }
```

```
>tree /f
|
|-- truffle-config.js
|-- contracts
|   |-- Migrations.sol
|   |-- HelloWorld.sol
|-- migrations
|   |-- 1_initial_migration.js
|   |-- 2_deploy_contracts.js
|-- test
```

```
2_deploy_contracts.js x
1  var HelloWorld = artifacts.require("HelloWorld");
2  module.exports = function(deployer) {
3      deployer.deploy(HelloWorld);
4      // Additional contracts can be deployed here
5  };
```

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/migrations/2_deploy_contracts.js

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/contracts/HelloWorld.sol

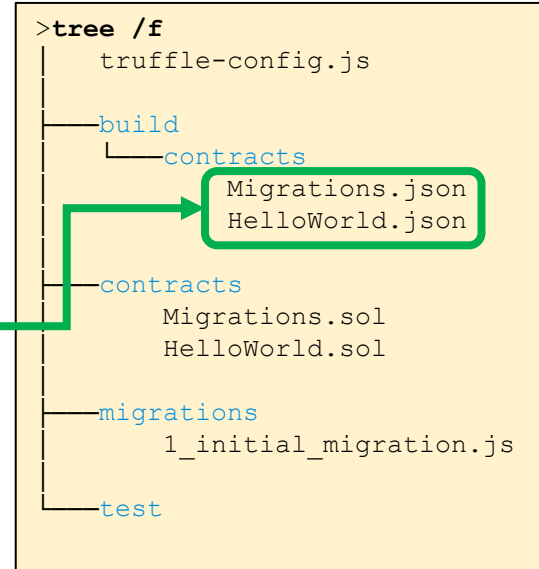
PD-7.1 Compile using truffle

```
>truffle compile
```

```
Compiling your contracts...
```

```
=====
```

```
> Compiling .\contracts\HelloWorld.sol  
> Compiling .\contracts\Migrations.sol  
> Artifacts written to ...HelloWorld\build\contracts  
> Compiled successfully using:  
- solc: 0.5.12+commit.7709ece9.Emscripten.clang
```



https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/build/contracts/HelloWorld.json

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/build/contracts/Migrations.json

<https://www.trufflesuite.com/docs/truffle/getting-started/compiling-contracts>

PD-7.1 Compile result 1/2

Migrations.json

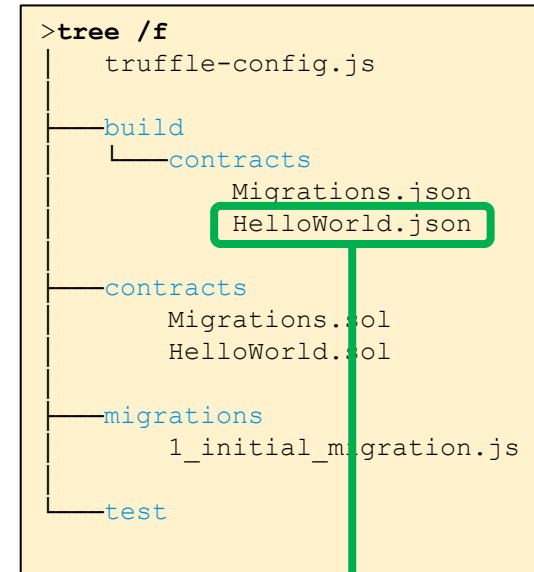
```
Migrations.json x
1  {
2    "contractName": "Migrations",
3    "abi": [
71   "metadata": "{\"compiler\":{\"version\":\"
72   "bytecode": "0x608060405234801561001057600
73   "deployedBytecode": "0x6080604052348015610
74   "sourceMap": "34:480:1:-;;;123:50;8:9:-1;5
75   "deployedSourceMap": "34:480:1:-;;;8:9:-1
76   "source": "pragma solidity >=0.4.21 <0.6.0
77   "sourcePath": "Z:/blockchain/web3examples/
78   "ast": {
727  "legacyAST": {
1376 "compiler": {
1380   "networks": {},
1381   "schemaVersion": "3.0.19",
1382   "updatedAt": "2019-12-06T09:31:40.341Z",
1383   "devdoc": {
1386   "userdoc": {
1389 }
```

```
>tree /f
truffle-config.js
├── build
│   └── contracts
│       └── Migrations.json
│           └── HelloWorld.json
├── contracts
│   ├── Migrations.sol
│   └── HelloWorld.sol
├── migrations
│   └── 1_initial_migration.js
└── test
```

PD-7.1 Compile result 2/2

HelloWorld.json

```
HelloWorld.json x
1  {
2    "contractName": "HelloWorld",
3    "abi": [
20   "metadata": "{\"compiler\":{\"version\":\"0
21   "bytecode": "0x6080604052604051806040016040
22   "deployedBytecode": "0x60806040523480156100
23   "sourceMap": "28:67:0:-;;;53:38;;;;;;;;;;;;;
24   "deployedSourceMap": "28:67:0:-;;;8:9:-1;5
25   "source": "pragma solidity ^0.5.12;\r\n\r\n
26   "sourcePath": "Z:/blockchain/web3examples/e
27   "ast": {
111  "legacyAST": {
195  "compiler": {
199   "networks": {},
200   "schemaVersion": "3.0.19",
201   "updatedAt": "2019-12-06T09:31:40.336Z",
202  "devdoc": {
205  "userdoc": {
208  }
```



PD-7.1 Next step: Migrate

JavaScript is used to deploy contracts to the Ethereum network

Command line:

```
> Truffle migrate
```

```
....
```

Interactive:

```
> Truffle console
```

```
truffle(development)> Truffle migrate
```

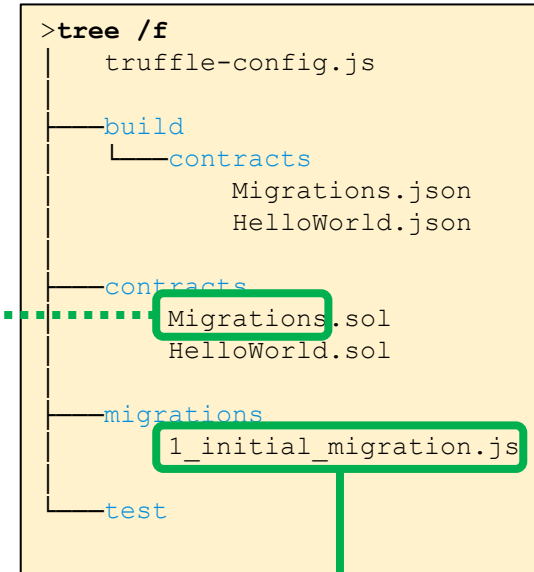
```
....
```

PD-7.1 Truffle - migrate

Migrations.sol

```
>truffle console
truffle(development)> truffle migrate
Compiling your contracts...
=====
> Everything is up to date, there is nothing to compile.
Starting migrations...
=====
> Network name:      'development'
> Network id:       5777
> Block gas limit:  0x6691b7
1_initial_migration.js
=====
Migrations.sol
  Deploying 'Migrations'
  -----
  > transaction hash:  0xa62d51bb663b697acfb60350935d87932d8bda6491bc71ff09e8481723e158f4
  > Blocks: 0         Seconds: 0
  > contract address: 0x5f5640BE4eb668b9DAbb67Cb56dC2Cfbc26D1Bb6
  > block number:     39
  > block timestamp:  1575634038
  > account:          0x6B5bB8441DD079F8Da87FF48F74F3A4F08bf417B
  > balance:          99.926789805
  > gas used:         263741
  > gas price:        20 gwei
  > value sent:       0 ETH
  > total cost:       0.00527482 ETH
  > Saving migration to chain.
  > Saving artifacts
  -----
  > Total cost:       0.00527482 ETH
Summary
=====
> Total deployments:  1
> Final cost:         0.00527482 ETH
```

PD-7.1 Javascript for migrate



```
1 const Migrations = artifacts.require("Migrations");
2
3 module.exports = function(deployer) {
4   deployer.deploy(Migrations);
5 };
```

The numbering convention:

- **x_script_name.js**, with x starting at 1.
- Own contracts would typically start at 2_....
- Migration.sol keeps track of the deployed numbers

<https://www.trufflesuite.com/docs/truffle/getting-started/running-migrations>

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/migrations/1_initial_migration.js

PD-7.1 Javascript for migrate

```
HelloWorld.sol x
```

```
1 pragma solidity ^0.5.12;  
2  
3 contract HelloWorld {  
4     string public welcome = "Hello World!";  
5 }
```

```
>tree /f  
|  
|   truffle-config.js  
|  
|-- contracts  
|   |-- Migrations.sol  
|   |-- HelloWorld.sol  
|  
|-- migrations  
|   |-- 1 initial migration.js  
|   |-- 2_deploy_contracts.js  
|  
|-- test
```

```
2_deploy_contracts.js x
```

```
1 var HelloWorld = artifacts.require("HelloWorld");  
2 module.exports = function(deployer) {  
3     deployer.deploy(HelloWorld);  
4     // Additional contracts can be deployed here  
5 };
```

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld/migrations/2_deploy_contracts.js

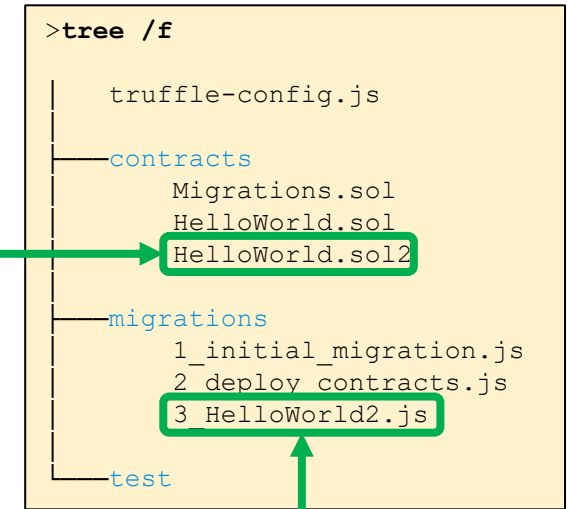
https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld1/contracts/HelloWorld.sol

PD-7.2 More contracts

```
HelloWorld2.sol x
1 pragma solidity ^0.5.12;
2 import "./HelloWorld.sol";
3
4 contract HelloWorld2 ←
5   address public OtherContract;
6   ..
7   // address of "HelloWorld.sol" instance has to be supplied
8   constructor (address ReferredContract) public {
9     OtherContract = ReferredContract; ..
10  }
11  ..
12  function Message () public view returns (string memory) {
13    return HelloWorld(OtherContract).welcome ();
14  } ..
15 }
```

```
3_HelloWorld2.js x
1 var HelloWorld = artifacts.require ("HelloWorld");
2 var HelloWorld2 = artifacts.require ("HelloWorld2");
3 module.exports = async function (deployer) {
4   Hello = await HelloWorld.deployed ();
5   await deployer.deploy (HelloWorld2, Hello.address); // supply address
6   Hello2 = await HelloWorld2.deployed ();
7
8   console.log (`HelloWorld is at address: ${Hello.address}`);
9   console.log (`Message from HelloWorld: ${await Hello.welcome ()}`);
10  console.log (`HelloWorld2 is at address: ${Hello2.address}`);
11  console.log (`HelloWorld2 links to: ${await Hello2.OtherContract ()}`);
12  console.log (`Message from HelloWorld2: ${await Hello2.Message ()}`);
13  };

```



https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld2/contracts/HelloWorld2.sol

https://github.com/web3examples/ethereum/blob/master/truffle_examples/HelloWorld2/migrations/3_HelloWorld2.js

PD-7.2 Javascript for Truffle Migrate

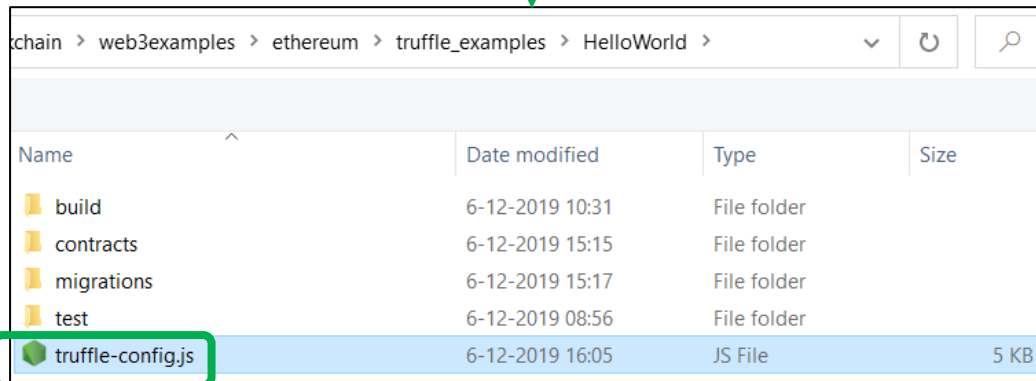
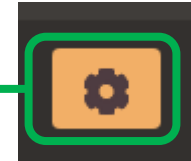
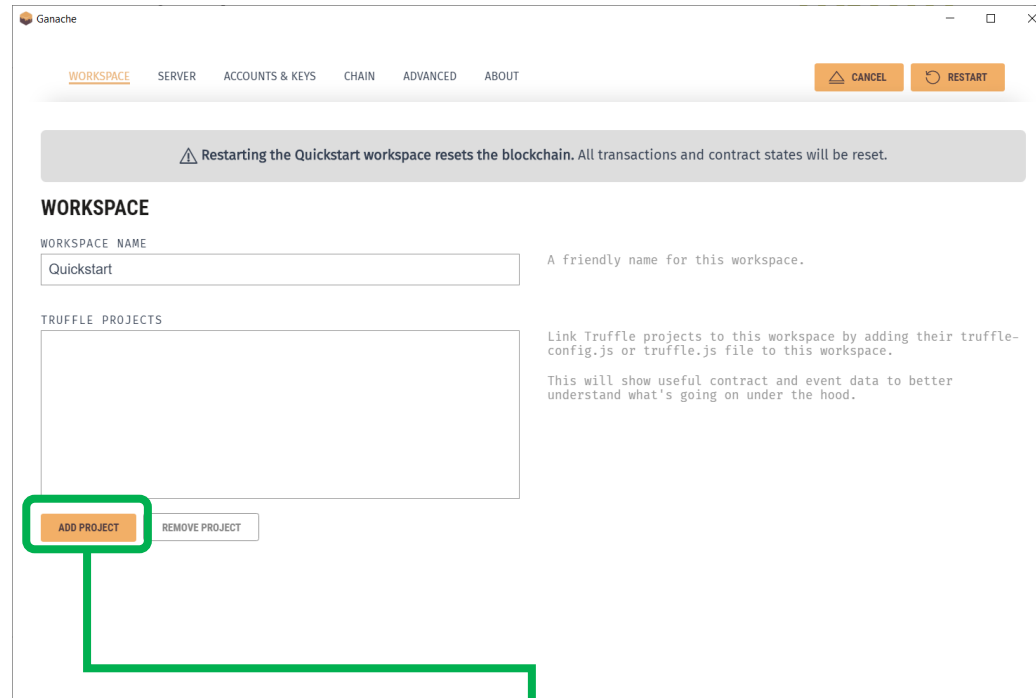
Simplified version of Web3.js

```
3_HelloWorld2.js x
1  var HelloWorld = artifacts.require("HelloWorld");
2  var HelloWorld2 = artifacts.require("HelloWorld2");
3  module.exports = async function(deployer) {
4    ...Hello = await HelloWorld.deployed()
5    ...await deployer.deploy(HelloWorld2, Hello.address); // supply address
6    ...Hello2 = await HelloWorld2.deployed()
7
8    ...console.log(`HelloWorld is at address: ${Hello.address}`);
9    ...console.log(`Message from HelloWorld: ${await Hello.welcome()}`)
10   ...console.log(`HelloWorld2 is at address: ${Hello2.address}`);
11   ...console.log(`HelloWorld2 links to: ${await Hello2.OtherContract()}`);
12   ...console.log(`Message from HelloWorld2: ${await Hello2.Message()}`) ...
13  };
```

<https://www.trufflesuite.com/docs/truffle/reference/contract-abstractions#contract-abstraction-api>

<https://www.trufflesuite.com/docs/truffle/reference/contract-abstractions#contract-instance-api>

PD-7.3 Connect to Ganache



PD-7.3 Contract info in Ganache

The screenshot shows the Ganache application window. The top navigation bar includes icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS (highlighted), EVENTS, and LOGS. A search bar is present on the right. Below the navigation bar, a status bar displays network settings: CURRENT BLOCK 6, GAS PRICE 2000000000, GAS LIMIT 6721975, HARDFORK PETERSBURG, NETWORK ID 5777, RPC SERVER HTTP://127.0.0.1:8545, MINING STATUS AUTOMINING, and WORKSPACE QUICKSTART. There are buttons for SAVE, SWITCH, and a settings gear.

The main content area shows the selected contract 'HelloWorld' at the path 'Z:\blockchain\web3examples\ethereum\truffle_examples\HelloWorld'. Below this is a table of deployed contracts:

NAME	ADDRESS	TX COUNT	STATUS
HelloWorld	0x4a07a27eF50c2bA0df2e8635A3Ef43E16a391343	0	DEPLOYED
HelloWorld2	0x349564dde70E5E7131d739921E04D4813f6D99Cf	0	DEPLOYED
Migrations	0x826a78659B4eca86BB1Fa181c85D21c1EfE72849	2	DEPLOYED

PD-7.3 Contact details

The screenshot shows the Ganache application interface. At the top, there is a navigation bar with icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS (highlighted), EVENTS, and LOGS. A search bar is also present. Below the navigation bar, a status bar displays various network parameters: CURRENT BLOCK (6), GAS PRICE (2000000000), GAS LIMIT (6721975), HARDFORK (PETERSBURG), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:8545), MINING STATUS (AUTOMINING), and WORKSPACE (QUICKSTART). There are also buttons for SAVE, SWITCH, and a settings gear.

The main content area shows the details for a contract named "HelloWorld". A "← BACK" button is on the left. The contract details include:

- ADDRESS: 0x4a07a27eF50c2bA0df2e8635A3Ef43E16a391343
- BALANCE: 0.00 ETH
- CREATION TX: 0xFe15bB914285F07Fc8CaBd624ef8cc8C884f3EaF0B8d1963AE6728862D237A21

The "STORAGE" section shows a single key-value pair: `welcome : string "Hello World!"`. This code is highlighted with a green box. The "TRANSACTIONS" section is currently empty, displaying "NO TRANSACTIONS".

PD-7.3 Contract details

Ganache

ACCOUNTS BLOCKS TRANSACTIONS **CONTRACTS** EVENTS LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 6 GAS PRICE 20000000000 GAS LIMIT 6721975 HARDFORK PETERSBURG NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:8545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART SAVE SWITCH

← BACK **Migrations**

ADDRESS 0x826a78659B4eca86BB1Fa181c85D21c1EfE72849 BALANCE 0.00 ETH

CREATION TX 0x796298f348204ea64213b8322d5CDc5FBa15699A204Eb667C18836B0c656f019

STORAGE

```
{ 2 items
  last_completed_migration : uint 3
  owner : address "0x6B5bB8441DD079F8Da..."
}
```

TRANSACTIONS

TX HASH	0x857d4b410913ffb3142d034c5193dbc3301a930de66edbc8f2925ccb68496a72	CONTRACT CALL	
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x6B5bB8441DD079F8Da87FF48F74F3A4F08bf417B	Migrations	27023	0
TX HASH	0xf6c69cbd66add4e713dfb2a10a69055f096b0061165a24de95a37ec432f30223	CONTRACT CALL	
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x6B5bB8441DD079F8Da87FF48F74F3A4F08bf417B	Migrations	27023	0

PD-7.4 Generate Mnemonic => deployment address

```
> npm install bip39
> npm install ethereum-mnemonic-privatekey-utils
> npm install web3
```

Alternative: use
mnemonic from a
wallet like MetaMask

```
gen_mnemonic.js x
1  const bip39 = require("bip39");
2  const pkutils = require("ethereum-mnemonic-privatekey-utils");
3  const Web3 = require("web3");
4
5  const web3 = new Web3();
6  const mnemonic = bip39.generateMnemonic();
7  console.log(mnemonic);
8
9  const privateKey = pkutils.getPrivateKeyFromMnemonic(mnemonic);
10 console.log(privateKey);
11
12 const address = web3.eth.accounts.privateKeyToAccount(privateKey).address;
13 console.log(address);
```

```
> node gen_mnemonic.js
```

```
orange prefer lyrics catch tape plug swim enough liar urban fan nothing
5a2a92262f8b7ff4333b88402131ce40200feba834f24e76a2a5d7810a4e2b98
0xC405315b116f5d966bF1984c25243b9b5AF1cF28
```

PD-7.4 Store Mnemonic

```
.secret x  
1 word1 word2 word3 word4 word5 word6 word7 word8 word9 word10 word11 word12
```

Account

0x6c728716a68499d486cda1701ab13c7b57f30aa0

Transfer some ETH to the newly created address

PD-7.5 Deploy on other networks

```
> npm install @truffle/hdwallet-provider
```

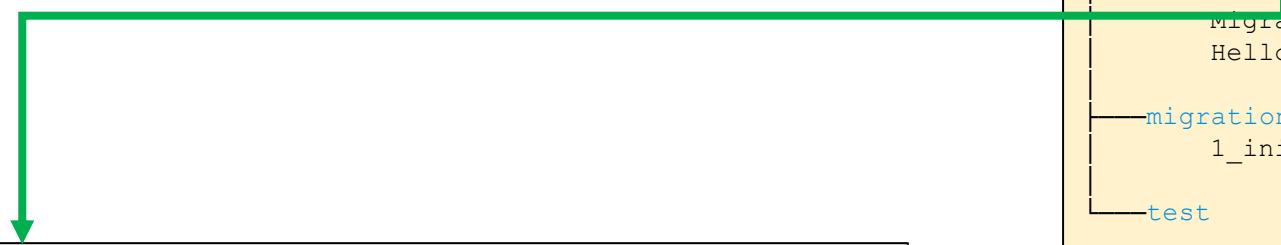
```
+ @truffle/hdwallet-provider@1.2.0
```

```
added 119 packages from 146 contributors and audited 809 packages in 24.683s
```

PD-7.5 Change configuration

Truffle-config.js

```
>tree /f
├── truffle-config.js
├── build
│   └── contracts
│       ├── Migrations.json
│       └── HelloWorld.json
├── contracts
│   ├── Migrations.sol
│   └── HelloWorld.sol
├── migrations
│   └── 1_initial_migration.js
└── test
```

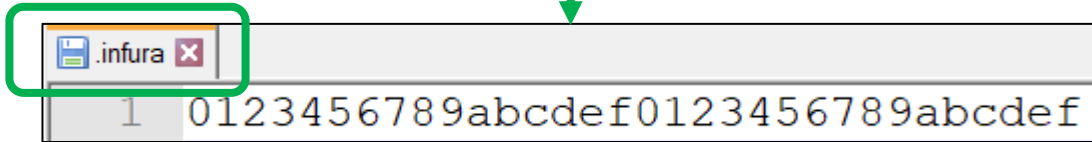


```
truffle-config.js x
1  /**
26  module.exports = {
27  /**
36  networks: {
37  /**
44  development: {
45    host: "127.0.0.1", // Localhost (default: none)
46    port: 8545, // Standard Ethereum port (default: none)
47    network_id: "*", // Any network (default: none)
48  },
49  /**
78  },
79  mocha: { // Set default mocha options here, use special reporters etc.
82  compilers: { // Configure your compilers
83    solc: {
94  }
95  }
```

PD-7.5 Signup for Infura key



The screenshot shows the Infura registration page in a browser. The page has an orange header with the Infura logo and the word "INFURA". Below the header, the text "Get Started for Free" is centered. There are three input fields for "NAME", "EMAIL", and "PASSWORD". Below these fields is a checkbox labeled "Stay up-to-date with our newsletter". A red "SIGN UP" button is positioned below the checkbox. At the bottom of the form area, there is a line of text: "Signing up signifies you have read and agree to the [Terms of Service and Privacy Policy](#)". At the very bottom of the page, there is a link: "Already have an account? [Log In](#)".

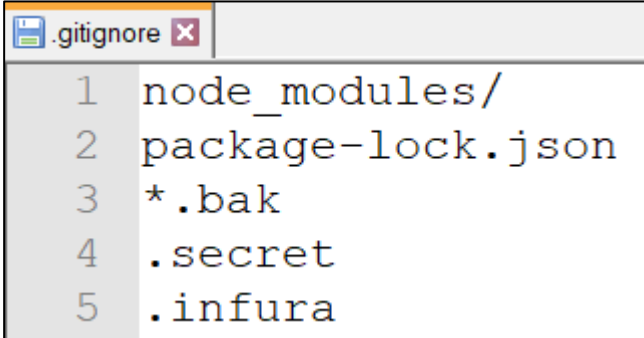


A terminal window is shown with a green box around the prompt ".infura". Below the prompt, the text "0123456789abcdef0123456789abcdef" is displayed, representing a hexadecimal string.

PD-7.5 Protect secrets

truffle-config.js

```
const fs = require('fs');
const mnemonic = fs.readFileSync(".secret")
    .toString().trim(); // contains mnemonic
const infuraKey = fs.readFileSync(".infura")
    .toString().trim(); // infura key
```

A screenshot of a text editor window showing a .gitignore file. The window title is ".gitignore" with a close button. The file contains five lines of text, each preceded by a line number from 1 to 5. The lines are: "node_modules/", "package-lock.json", "*.bak", ".secret", and ".infura".

```
.gitignore
1 node_modules/
2 package-lock.json
3 *.bak
4 .secret
5 .infura
```

PD-7.5 Adapt configuration truffle-config.js

```
truffle-config.js
1  const HDWalletProvider = require('@truffle/hdwallet-provider');
2
3  const fs = require('fs');
4  const mnemonic = fs.readFileSync(".secret").toString().trim(); // contains mnemonic
5  const infuraKey = fs.readFileSync(".infura").toString().trim(); // infura key
6
7  var adr;
8
9
10
11 module.exports = {
12   networks: {
13     development: {
14       host: "127.0.0.1", // Localhost (default: none)
15       port: 7545, // Standard Ethereum port (default: none)
16       network_id: "*", // Any network (default: none)
17     },
18     ropsten: {
19       provider: () => new HDWalletProvider(mnemonic, `https://ropsten.infura.io/v3/${infuraKey}`),
20       network_id: 3, // Ropsten's id
21       gas: 5500000, // Ropsten has a lower block limit than mainnet. Default is 6721975.
22       skipDryRun: true
23     },
24     rinkeby: {
25       provider: () => new HDWalletProvider(mnemonic, `https://rinkeby.infura.io/v3/${infuraKey}`),
26       network_id: 4, // rinkeby id
27       skipDryRun: true
28     },
29     goerli: {
30       provider: () => new HDWalletProvider(mnemonic, `https://goerli.infura.io/v3/${infuraKey}`),
31       network_id: 5, // goerli's id
32       gas: 300000, // default: 6721975, // limit: 8000000
33       skipDryRun: true
34     },
35     aethereum: {
36       provider: () => {
37         if (!adr) {
38           adr = new HDWalletProvider(mnemonic, "https://api.avax-test.network/ext/bc/C/rpc");
39           console.log("Make sure there is balance on deployment account: ${adr.getAddress()}");
40         }
41         return adr;
42       },
43       network_id: 1, // aethereum id
44       gas: 300000, // limit: 25968880
45       gasPrice: 470000000000, // minimum amount
46       skipDryRun: true
47     },
48     // default gasPrice: 20000000000, // 20 gwei
49   },
50   mocha: {},
51   compilers: { solc: {} }
52 }
```

```
rinkeby: {
  provider: () => new HDWalletProvider(mnemonic, `https://rinkeby.infura.io/v3/${infuraKey}`),
  network_id: 4, // rinkeby id
  skipDryRun: true
},
```

https://github.com/web3examples/ethereum/blob/master/truffle_examples/TestnetDeploy/truffle-config.js

<https://www.trufflesuite.com/docs/truffle/reference/configuration#networks>

PD-7.5 Truffle migrate

```
truffle migrate --network development --reset
truffle migrate --network ropsten --reset
truffle migrate --network rinkeby --reset
truffle migrate --network goerli --reset
truffle migrate --network athereum --reset
```

PD-7.5 Fund account

```
>truffle migrate --network aethereum --reset  
Deployment account: 0x6c728716a68499d486cda1701ab13c7b57f30aa0
```

```
Compiling your contracts...
```

```
=====
```

```
> Everything is up to date, there is nothing to compile.
```

```
Starting migrations...
```

```
=====
```

```
1_initial_migration.js
```

```
=====
```

```
Replacing 'Migrations'
```

```
-----
```

```
Error: *** Deployment Failed ***
```

```
"Migrations" could not deploy due to insufficient funds
```

```
* Account: 0x6c728716a68499d486cDA1701AB13C7b57f30aA0
```

```
* Balance: 0 wei
```

```
* Message: insufficient funds for gas * price + value
```

```
* Try:
```

```
+ Using an adequately funded account
```

```
+ If you are using a local Geth node, verify that your node is synced.
```

PD-7.5 Deploy on Athereum

```
>truffle migrate --network athereum --reset
```

```
Deployment account: 0x6c728716a68499d486cda1701ab13c7b57f30aa0
```

```
Compiling your contracts...
```

```
=====
```

```
> Everything is up to date, there is nothing to compile.
```

```
Starting migrations...
```

```
=====
```

```
> Network name: 'athereum'
```

```
> Network id: 1
```

```
> Block gas limit: 0x7a1200
```

```
1_initial_migration.js
```

```
=====
```

```
2_deploy_contracts.js
```

```
=====
```

```
Summary
```

```
=====
```

```
> Total deployments: 2
```

```
> Final cost: 0.000000000000456932 ETH
```