

COMPTE RENDU – PET SHOP

MISE EN PLACE DE L'ENVIRONNEMENT

[illegible][illegible]

```
nathan@nathan-VirtualBox:~$ sudo npm install -g truffle
npm WARN deprecated flat@4.1.0: Fixed a prototype pollution security issue in 4.1.0, please upgrade to ^4.1.1 or ^5.0.1.
/usr/local/bin/truffle -> /usr/local/lib/node_modules/truffle/build/cli.bundled.js

> truffle@5.1.47 postinstall /usr/local/lib/node_modules/truffle
> node ./scripts/postinstall.js
```

INSTALLATION DE SUBLIME TEXT

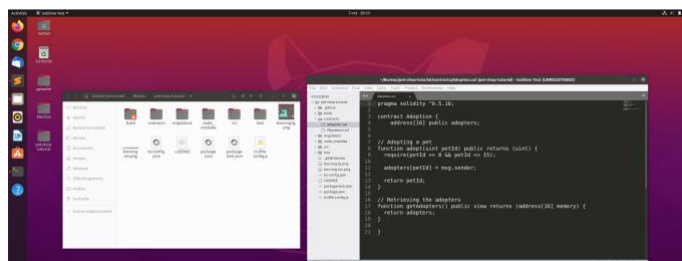
Commandes à saisir

[illegible]

CREATION DU DOSSIER POUR LE TUTORIEL ET OUVERTURE DU FICHIER UTILISÉ POUR LE TP

Commandes à saisir pour créer le dossier et préparer le dossier truffle du TP

```
mkdir pet-shop-tutorial
cd pet-shop-tutorial
truffle unbox pet-shop
```



Voici le résultat visuel avec l'outil Sublimtext

ECRITURE DU SMART CONTRACT

CREATION ET REMPLISSAGE DU FICHIER ADOPTION.SOL

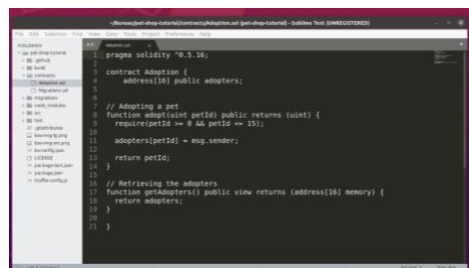
Ce fichier contient une variable, les adoptants.

La ligne : `address[16] public adopters;`

Ceci est un tableau d'adresses Ethereum. Les tableaux contiennent un type et peuvent avoir une longueur fixe ou variable. Dans ce cas, le type est adresse et la longueur est 16

Ensuite, il y a deux fonctions :

La première Adopting a pet permet de vérifier que l'animal à adopter est disponible ; mettre en relation l'animal et l'adopteur et la deuxième fonction Retrieving the adopters permet de confirmer l'opération.



COMPILATION DU PROGRAMME

Il faut saisir la commande :

Truffle compile

Cette commande permet de vérifier que tout est bien compilé sans erreurs.

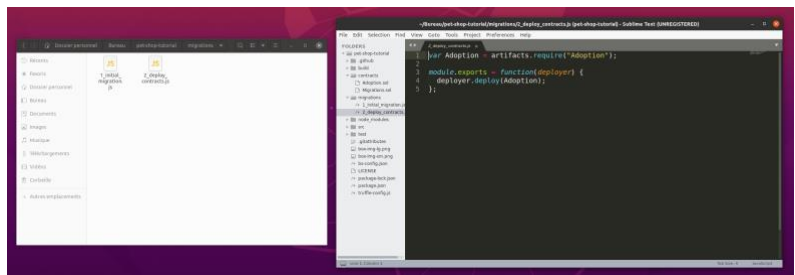
```
nathan@nathan-VirtualBox:~$ ls
Bureau  Documents  migrations  Musique  snap  test  Vidéos
contracts  Images  Modèles  Public  Téléchargements  truffle-config.js
nathan@nathan-VirtualBox:~$ cd Bureau/pet-shop-tutorial/
nathan@nathan-VirtualBox:~/Bureau/pet-shop-tutorial$ truffle compile

Compiling your contracts...
=====
> Everything is up to date, there is nothing to compile.

nathan@nathan-VirtualBox:~/Bureau/pet-shop-tutorial$
```

CREATION ET REMPLISSAGE DU FICHIER 2_DEPLOY_CONTRACTS.JS

Ce fichier permet de faire migrer les contrats sur la blockchain.



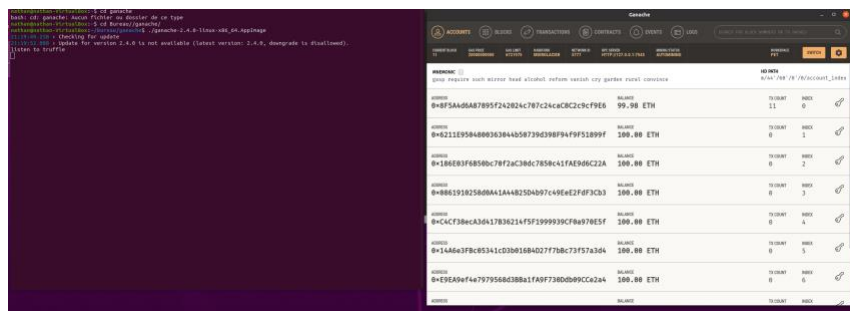
INSTALLATION ET MISE EN PLACE DE GANACHE

C'est une blockchain personnelle pour le développement utilisant Ethereum que l'on peut utiliser pour déployer des contrats, développer des applications et exécuter des tests.

Télécharger ganache via le lien : <https://www.trufflesuite.com/ganache>, on obtient une image du type ganache-1.3.0-x86_64.AppImage.

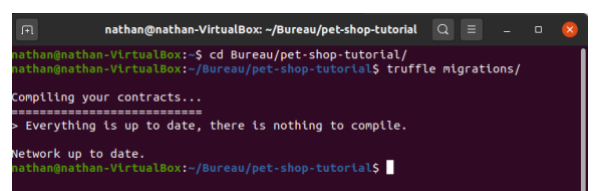
Ensuite, saisir : `chmod a+x ganache-1.3.0-x86_64.AppImage` et `./ganache-1.3.0-x86_64.AppImage`

Ainsi, ganache est téléchargé.



Maintenant, on effectue la migration du contrat dans la blockchain. Chaque migration entraîne une légère perte d'ETH.

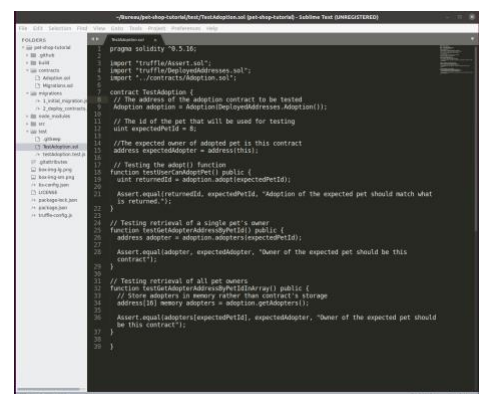
On tape la commande : `Truffle migrate`

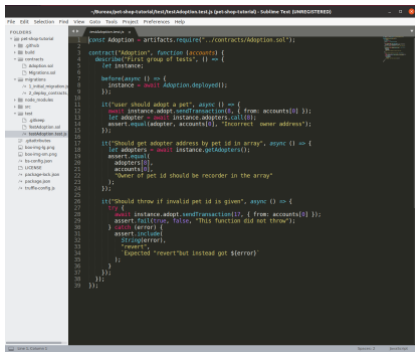


Création et remplissage du fichier TestAdoption.sol

Ce fichier comprend plusieurs imports permettant de varier les tests.

Ensuite, des fonctions afin de tester différents scénarios utilisateurs.

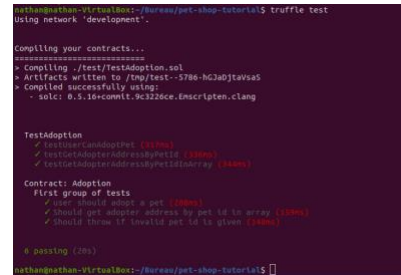




Création et remplissage du fichier TestAdoption.test.js

Ce fichier comprend plusieurs fonctions afin de tester si l'utilisateur qui adopte un animal est le bon utilisateur et qu'il adopte le bon animal.

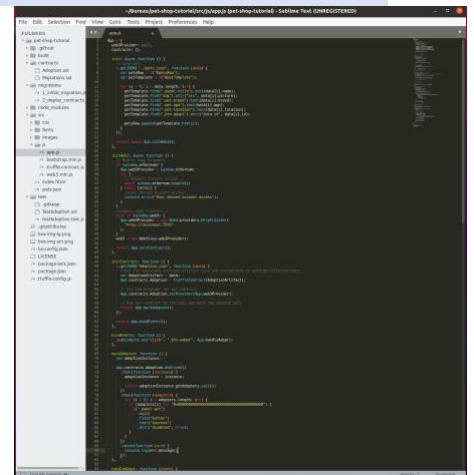
Vérifions les tests avec : truffle test



CREATION DE L'INTERFACE UTILISATEUR

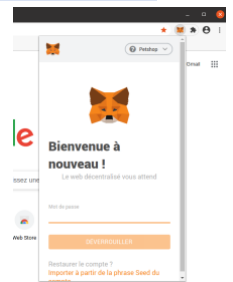
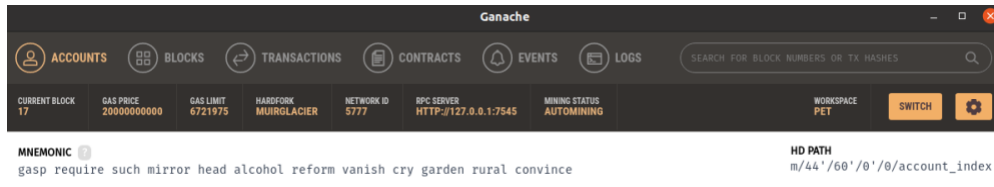
Se rendre dans app.js

Une fois ici, il y a un grand nombre de fonctions à compléter afin de mettre en place ce qui concerne les animations, les actions réalisés par les utilisateurs et avec dapp etc



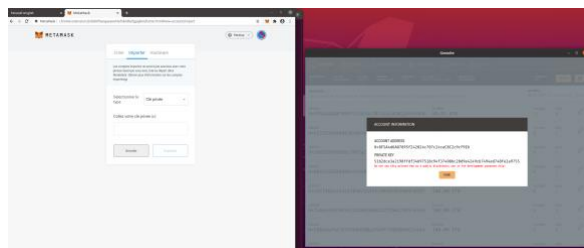
INSTALLATION ET CONFIGURATION DE METAMASK

Il faut installer chrome et l'extension metamask. Il faut choisir l'option qui indique que l'on a déjà un compte et entrer la « seed phrase » que l'on trouve sur ganache.



Ensuite, il faut créer un nouveau réseau et indiquer l'adresse RPC server que l'on trouve sur ganache <http://127.0.0.1:7545>

Ensuite, on peut importer les comptes de ganache en cliquant sur importer un compte en entrant la clé privée de l'un des 10 comptes possédants chacun 100 ETH



- Réseau principal Ethereum
- Réseau de test Ropsten
- Réseau de test Rinkeby
- Réseau de test Goerli
- Réseau de test Kovan
- Localhost 8545
- Petshop

Il faut créer un nouveau réseau principal avec le nom de votre réseau et l'adresse RPC.

Nom du réseau
Petshop

Nouvelle URL de RPC
<http://127.0.0.1:7545>

ChainID (hexadécimal)
0x00

Symbole (facultatif)
ETH

Blocage URL de l'explorateur (facultatif)

Annuler Importer

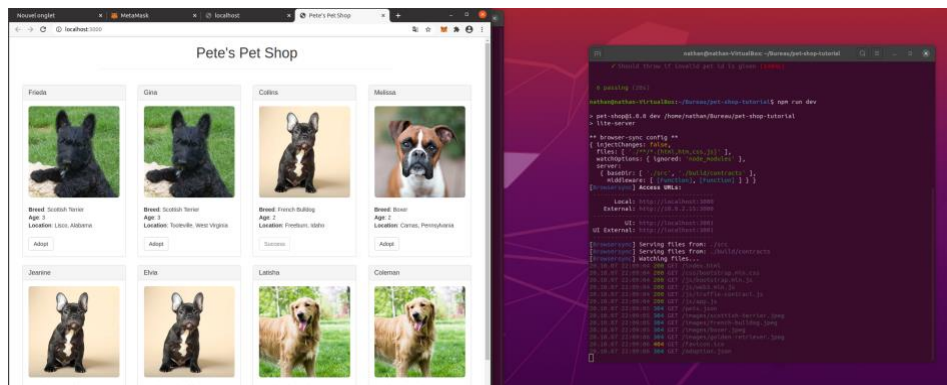
CREATION ET LANCEMENT DU SERVEUR LOCAL

Il faut ouvrir **bs-config.json** et le remplir

```
{
  "server": {
    "baseDir": ["/src", "/build/contracts"]
  }
}
```

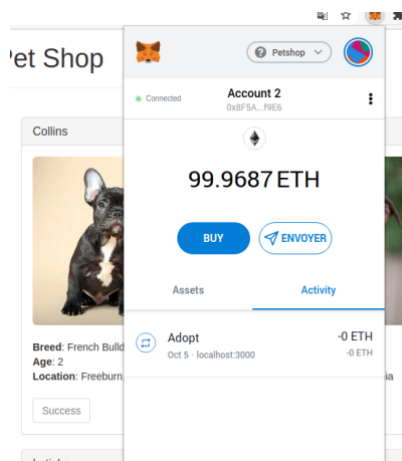
Ces lignes permettent de choisir les éléments à mettre sur le serveur.

Lancement du serveur avec la commande : `npm run dev`



Ensuite, il faut accepter la connexion du site à metamask.

Maintenant, on peut adopter l'animal que l'on souhaite et suivre l'historique des transactions depuis metamask.



Chacune des transactions est ainsi enregistré dans la blockchain avec l'identifiant du compte.