|  |  |  |  |
| --- | --- | --- | --- |
| VERSIONS | DATE | OBJET | NOM |
| 1.0 | 02/01/2018 | Première version du document d’installation du POC ethereum | FH |
| 1.1 | 03/01/2018 | update installation document | FH |
| 1.2 | 04/01/2018 | modification | FH |
| 1.3 | 17/05/2018 | Première version du document d’installation du POC ethereum | YH |
| 1.4 | 23/07/2018 | Mise à jour du document d’installation | YH |

Sommaire

[1 Introduction 3](#__RefHeading___Toc322619149)

[2 Liste de prérequis 3](#__RefHeading___Toc322619150)

[3 Les environnements 3](#__RefHeading___Toc2255_338905324)

[4 Mise en place de l’environnement de Production 3](#__RefHeading___Toc2261_338905324)

# Introduction

Cette procédure décrit les différentes opérations à réaliser pour installer *le poc ethereum Donation.* Ce document ne traite que de la configuration de production.

# Liste de prérequis

Les prérequis à vérifier et compléter avant l’installation sont les suivants :

| N° | Pré requis | Illustration |
| --- | --- | --- |
|  | truffle | sudo npm install -g truffle |
|  | metamask | Extension google |
|  | Ganache (La nouvelle version de testrpc) | Disponible sous ce lien https://truffleframework.com/ganache |

# Les environnements

Un environnement de production pour :

* *Le déploiement du smart contrat (solidity script) sur le réseau ethereum public via truffle.Ce déploiement sera effectué via le provider infura.*

# Mise en place de l’environnement de Production

Commentaires et remarques sur les actions du chapitre

| Étape | Action | Illustration |
| --- | --- | --- |
|  | installation de git :  sudo yum install git | git –version  git version 1.8.3.1 |
|  | installation truffle  en tant qu’admin (via un sudo)  sudo npm install -g truffle | truffle version  Truffle v4.1.8 (core: 4.1.8)  Solidity v0.4.23 (solc-js) |
|  | Installation du plugin Metamask depuis le site officiel :  <https://metamask.io/>  Il faut selectionner le réseau ethereum. Dans notre cas, il faut utiliser un réseau de test (Ropsten test net). (cercle bleu)  A l’initialisation du plugin un compte ethereum est demandé. L’utilisateur peux importer un compte existant ou bien créer un nouveau compte. (cercle rouge)  Attention n’oublier pas ce mot de passe/ ne le partager avec personne d’autre.  Le bouton « buy » permet de recevoir des ether « de test ». Il est possible d’aller faire une demande directement auprès d’un faucet (par exemple, http://faucet.ropsten.be:3001/) | selection du réseau en bleu.  création/import d’un compte ethereum en rouge |
|  | Récupération du code source du projet par clonage du repository github.  git clone https://github.com/Siggg/donation.git | L’utilisateur doit s’identifier au près de github afin de s’assurer des droits d’accès au repository. L’utilisateur doit être enregistrer et doit avoir les droits d’accès. Un mot de passe est demandé pour le clonage des sources. |
|  | Une fois le clonage est effectué.  cd ~/prod/donation/solidity  truffle compile | Produit l’ABI du contrat dans le repertoire ~/donation/solidity/build |
|  | Installation des node\_modules utilisé par truffle (pour déploiement et tests fonctionnels)  cd ~/prod/donation  npm install |  |
|  | déploiement du smart contrat (via truffle)  cd ~/prod/donation/solidity  truffle migrate –network ropsten | Using network 'ropsten'.  Running migration: 1\_initial\_migration.js  Replacing Migrations...  ... 0xf41532a8f1d5fd15cb502299a6b7478d417d8d19d674d249fc4d0e26f797181a  Migrations: 0x7ec775a31a9a0fd09de36f57cebd4c2fd99ef660  Saving successful migration to network...  ... 0x392c0e1c919ea12afbac74c4eca38f4214672edd3daa004dd4c0c92cfcbb8ab6  Saving artifacts...  Running migration: 2\_deploy\_contracts.js  0x1ac471f56404c4fe49c378b5b9bc63fb689944b9  Replacing DonationV2...  ... 0xf248af805e49da9708381364484fd8dfc8ac759fdd91a9abc720344fd8824231  DonationV2: 0x8dce0cb938afb96420f3c373cd1a2c7432bf11c4  >> DonationV2 deployed at address 0x8dce0cb938afb96420f3c373cd1a2c7432bf11c4  Saving successful migration to network...  ... 0x30594e444f93f2b3be456e57851c1c36cb4a62f0159aad68c5d982f707b06149  Saving artifacts… |
|  | déploiement du smart contrat (via remix)   * Copier le code source du smart contrat disponible sur le github sous : [donation](https://github.com/Siggg/donation)/[solidity](https://github.com/Siggg/donation/tree/master/solidity)/[contracts](https://github.com/Siggg/donation/tree/master/solidity/contracts)/DonationV2.sol * Aller sur le site de remix IDE [https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.19+commit.c4cbbb05.js](https://remix.ethereum.org/" \l "optimize=false&version=soljson-v0.4.19+commit.c4cbbb05.js) * Créer un nouveau fichier et le nommer DonationV2.sol (comme le contrat à déployer) * Coller le code source du smart contract sous le fichier créé * Se connecter à metamask via la seed phrase (Restore from seed phrase) * Taper la seed phrase et créer un nouveau mot de passe * Sur remix, dans l’onglet « Settings », il est possible de préciser la version du compilateur solidity.   Dans l’onglet « Run », si metamask est installé et le compte ethereum débloqué alors Remix initialise son envirronnement (cercle rouge) automatiquement avec le réseau choisi par metamask (ici ropsten(3) … le 3 represente l’identfiant du réseau).  Injected web3 signifie que remix injecte les commandes sur le nœud ethereum spécifié (dans notre cas metamask indique ropsten).  On peut déployer alors notre smart contrat en cliquant sur le bouton « Deploy » (cercle vert) en indiquant comme paramètre l’adresse du certifieur (ici c’est "0x03A00d82b285e4A7ad3997c1aa22DaD0A1410720")   * Un popup de metamask s’affiche pour signer la transaction du déploiement. Cliquer sur le bouton « SUBMIT » pour procéder au déploiement. * Une fois déployé le smart contrat apparaîtra. On peut alors récupérer son adresse en cliquant sur le bouton copier (cercle rouge).   Dans cet exemple, l’adresse est "0x4226bf9cf5255d7f22fc7b397c30b1c382e0eb80" |  |
|  | Installation de ganache testrpc | Ganache est disponible sous <https://truffleframework.com/ganache>  Une fois installé, il suffit de le lancer et faire quelques modifications pour récupérer les mêmes comptes ethereum utilisés pour les tests. Pour cela, il faut :   1. Lancer l’application ganache 2. Partir sur réglages > ACCOUNTS & KEYS 3. Désactiver l’option **«  AUTOGENERATE HD MNEMONIC »** 4. Copier coller ce Mnemonic « **door sadness shallow hire fame lesson wonder scan donate caution apple chicken** » 5. Cliquer sur restart pour prendre en considération les modifications effectuées |