Direction des Systèmes d'Information

Installation de POC Ethereum (Donation)

hauts-de-seine
LE DÉPARTEMENT

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

Sommaire

1 Introduction

Cette procédure décrit les différentes opérations à réaliser pour installer *le poc ethereum Donation.*

Les chapitres peuvent être des chapitres décrivant les pré-requis, le schéma de l'architecture, la préparation des serveurs à l'installation, le détail de l'installation, les étapes post-installation, la qualification de l'installation, les tests à effectuer, la mise à jour documentaire, le ***retour arrière*** *(obligatoire)...*

Faire un chapitre par sujet**.**

2 Liste de prérequis

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

N°	Pré requis	Illustration
1.	gcc compiler	
2.	go compiler	`sudo yum install golang`
3.	redhat 7.3	`os`

La check-list de vérification est référencée *XXX-NNNN-V (check list pré requis .....).sfx*.

3 Les environnements

Un environnement de production pour :

- *le lancement d'un nœud Ethereum sur le réseau public.*
- *le deploiement de smart contrat (solidity script) sur le réseau ethereum public via truffle.*

4 Mise en place de l'environnement de Production

Commentaires et remarques sur les actions du chapitre

Éta	Action	Illustration
4.	update certificates	

Éta	Action	Illustration
	sudo yum reinstall ca-certificates	
5.	installation de git : `sudo yum install git`	git –version git version 1.8.3.1

6.	`1) retrieve golang tar ball :`
	`sudo yum install golang`
	
	or
	
	`wget https://storage.googleapis.com/golang/go1.8.5.linux-amd64.tar.gz`
	tar -xvf go1.8.5.linux-amd64.tar.gz
	
	`2) configure go path :`
	`export GOROOT=/usr/local/go`
	`export GOPATH=$HOME`
	`export PATH=$GOPATH/bin:$GOROOT/bin:$PATH`
	
	`go version`
		go version go1.8.5 linux/amd64

7.	`install GCC compiler :`
	
	`yum group install "Development Tools"`
	`which gcc`
		/usr/bin/gcc

8.	installation node `curl --silent --location https://rpm.nodesource.com/setup_8.x | sudo bash -` yum install -y nodejs	node -v v8.9.4 npm -v 5.6.0
9.	installation truffle en tant qu'admin (via un su) npm install -g truffle	truffle version Truffle v4.0.4 (core: 4.0.4) Solidity v0.4.18 (solc-js)
10.	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.

Éta	Action	Illustration
11.	Pour Redhat git clone https://github.com/ethereum/go-ethereum.git cd go-ethereum make geth export PATH=/home/frank/go-ethereum/build/bin:$PATH	Build the geth executable for your system.
12.	Creation d'un compte ethereum geth --datadir "~/donation/data" account new	Produit une adresse correspondant au compte de l'utilisateur.
13.	Lancement d'un nœud ethereum sur le réseau public de test appeler testnet. Le lancement du nœud necessite l'authentification du premier compte au lancement du nœud. geth --datadir "/home/frank/prod/donation/data" --unlock "0xfe3925a85b04be979ef5c6b1bad5361dc30d89b6" --port "30303" --rpc --rpcaddr "localhost" --rpcport "8545" --rpcapi "admin,eth,miner,net,web3" --mine --minerthreads 1 --etherbase "0xfe3925a85b04be979ef5c6b1bad5361dc30d89b6"	Une invite de commande demande le mot de passe du compte[0] (unlock « 0 »). Une fois le compte ouvert, la synchronisation du nœud démarre. I0103 15:17:26.051038 eth/downloader/downloader.go:1437] imported 14 state entries in 21.773646ms: processed 4640446, pending at least 272 I0103 15:18:24.173302 eth/downloader/downloader.go:1437] imported 1 state entries in 2.90049ms: processed 4640447, pending at least 17 Une fois la synchronisation terminée, le nœud est lancé et chaque bloc miné est loggé : par exemple, I0102 11:07:09.246739 miner/worker.go:573] commit new work on block 5837 with 0 txs & 0 uncles. Took 106.137µs
14.	Une fois la synchronisation finie , compilation du smart contrat cd ~/donation/solidity truffle compile --all	Produit l'ABI du contrat dans le repertoire ~/donation/solidity/build
15.	deploiement du smart contrat cd ~/donation/solidity truffle migrate –network live	Le script de déploiement indique l'adresse du contrat : Running migration: 1_initial_migration.js Deploying Migrations... Migrations:

Éta	Action	Illustration
		0x6fa8919fd68a4542e97b600d9389da337484c778 >> Successfully deployed Migrations contract Saving successful migration to network... Saving artifacts... Running migration: 2_deploy_contracts.js Deploying Donation... Donation: 0x1c09980c497ddd30fd80202421da1bb65bfe0bad >> Donation deployed at address 0x1c09980c497ddd30fd80202421da1bb65bfe0bad Saving successful migration to network... Saving artifacts...
16.		