Deploying a smart contract node

We will see how to deploy a node capable of launching smart contracts. As part of this tutorial, we use a test blockchain, whose list of master nodes is: 167.86.103.31:8080,5.189.168.49:8080,173.212.229.88:8080,62.171.153.36:8080,167.86.124.188:8080

The administrators of this blockchain are members of the Kalima Systems team. For any administrative manipulation, such as the authorization of a node, please contact: jerome.delaire@kalima.io

I. Expand node

We consider in this tutorial that the smart contract node is already developed. If not, you can expand it by following the SmartContractNode tutorial.

II. Install the node

To work, the node must be installed on a linux machine with a public IP address, because the contracts will be deployed on this machine by a devOps chain via SSH. If you want to install a smart contract node in a machine that does not have a public IP address , we are able to offer an alternative solution.

You can then install the node you developed on your machine.

III. Connect the node to the blockchain

To work, the node must be authorized on the blockchain on an address list of our choice with at least one authorization on /Kalima_Scripts (to retrieve encryption information and contract signatures).

This authorization step is done by an admin, we authorize the node via a temporary serialId (valid for 5 minutes). In order for a blockchain administrator to authorize your smart node contract, you must provide them with:

- The IP address of the machine that will host the node
- The user of the machine for whom the contracts will be deployed. If the user is rcs for example, the contracts will be deployed in /home/rcs/contracts/
- The list of smart contracts that your node will be able to launch

IV. Create a git directory for contracts

The contracts are stored on a git directory, and then the encrypted contracts are deployed to the smart contract nodes. So you need to create a git directory for your smart contracts . This git directory must be readable by the devOps string.

As part of this tutorial, you can create a directory on our git server: https://git.kalimadb.com/

It is then necessary to give read access to the gocd user, and to indicate the link of the directory.

V. Copy the SSH public key

For contracts to be deployed on your machine, the devOps chain must be able to access them. To do this, you must copy the SSH public key of the devOps string to your machine. As part of this tutorial, the key has been provided to you by Kalima Systems, if not, please contact jerome.delaire@kalima.io

To copy the key:

ssh-copy-id -i id_rsa.pub <user>@<ip_cible>

With:

- user The user of your machine ➔
- ip_cible➔ Your machine's IP address