How to easily install a local RSK node and then interact with it.

Thomas Zoughebi, Eureka

Here is how I managed to « simply » (my point of view) install a Rootstock local node on two cheap notebooks :

- ✔ Bodhi Linux 64b on ASUS EeeBook X205TA.
- ✔ Bodhi Linux 32b on ASUS EeePC 1001HA.

Bodhi Linux is based on Ubuntu Linux with a Moksha desktop and is pretty lightweight.

This help is based on the RSK team wiki and personal help from Ruben Altman of RSK :
https://github.com/rsksmart/rskj/wiki

Requirements :

- Install **Node.js** (and *npm*) if not already done : https://nodejs.org/
- Install **GIT** if not already done : https://git-scm.com/
- Install **solc** – you need to install the Solidity Compiler as a linux package, this is **not** the Node.js package we are talking about here. You can have the *solc* Node.js *module* installed also, but you need the independant package for the RSK node. And then write the path to the solc app somewhere to remember.

  Terminal commands :

```
sudo -s
add-apt-repository ppa:ethereum/ethereum
apt update
apt install solc
```

Then :

1) RSKJ

Git clone the RSKJ repository :
https://github.com/rsksmart/rskj

2) JAVA (the node is written in Java)

Let's install JAVA easily and set the Oracle v8 as default.

If you need, follow this link for details and other OS versions :
http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubuntu-via-ppa.html

Terminal commands :

```
sudo -s
add-apt-repository ppa:webupd8team/java
apt update
apt install oracle-java8-installer
apt install oracle-java8-set-default
```

To check if installation went correctly, check the java version :

```
java -version
```

As admin, modify « **/etc/environment** » and add the following line :

```
JAVA_HOME="/usr/lib/jvm/java-8-oracle"
```

3) SDKMAN

We will need Sdkman to easily install Gradle.

Terminal commands :

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
```

Of course, you need the « curl » package to do this...

To check if installation went correctly :

```
sdk version
```

4) GRADLE

We will need Gradle to import the Rskj node project source into Intellij IDEA.

Terminal command :

```
sdk install gradle 4.1
```

5) INTELLIJ IDEA (Community)

We will use IDEA to compile the RSK node.

Download the tar.gz : https://download.jetbrains.com/idea/ideaIC-2017.2.3.tar.gz
Extract it into a folder using the following Terminal command into the folder :

```
tar xfz ideaIC-2017.2.3.tar.gz
```

Now we launch IDEA :
In a Terminal, and from the folder you extracted the tar.gz, go into « idea/bin/ ».
Then, type the following command to load the script :

```
./idea.sh
```

When IDEA is launched, you should have a window with different choices. Choose « **Import project** ».
Then, browse to the « rskj » directory where you cloned the rskj github repository and click « **Next** ».
Choose « Gradle » project import and click « **Next** ».
Within the dialog, select « *Use default gradle wrapper* » and then click « **Finish** ». Keep IDEA opened.

Download the file « standalone.conf » and edit the last line « solc.path = /xxx/xxxx/» replacing the « /xxx/xxxx/ » with the installation's path of your own « solc » app.
Copy the modified « standalone.conf » into « rskj-core/config/ » directory of your github clone.

Back into IDEA, at top right corner you should see an arrow, click on it and select « **Edit Configuration...** ».
As « application » name it « Rskj ». Then fill the fields like this :

Main class:	co.rsk.Start
VM options:	-Drsk.conf.file=/xxxxxx/rskj/rskj-core/src/main/resources/config/standalone.conf
Working directory:	/xxxxxx/rskj

Replace the « /xxxxxx/ » with your own pathes.

« *Use classpath of module* » field should be automatically set as « rskj-core_main ».
« *JRE* » field should be automatically set as « Default (1.8 - SDK of 'rsk-core_main' module) ».
Don't modify anything else and click « **Apply** » and « **OK** ».

Your node is set!

To launch the node, use the green arrow next to the « *Edit Configuration...* » menu you used earlier.

6) RSK Console

To interact with the node we can use the RSK console. It's a Node.js app, as a JavaScript.

Download « console.js » and « package.json » from here :
https://github.com/rsksmart/utilities/tree/master/console

Place them both in a folder you can easily find again.
To launch the console, use the following Terminal command :

```
node console.js -server HOST:PORT
```

Remember that you need everything in the folder you placed the JavaScript « console.js ». Everything which is required by the code of this file. It means you may need Node.js modules locally installed into this folder too.

You can now, at the time I'm writing this « How to » use almost every Web3.js commands to speak with the node.