Upgrade to 1.25.1

Quick upgrade guide to 1.25.1 (from 1.24.x)

Quick update guide for SPOs who have the previous version already installed (and all the previous dependencies) - if you are new - then follow the installation guide.

Let's start with backing up current binaries

```
cd .local/bin/

# let's create a folder with the version number
mkdir -p $(cardano-node version | grep -oP '(?<=cardano-node )[0-9\.]+')

# copying files to the created folder
cp cardano-node $(cardano-node version | grep -oP '(?<=cardano-node )[0-9\.]
cp carnao-cli $(cardano-node version | grep -oP '(?<=cardano-node )[0-9\.]+
```

Let's move forward with upgrades

```
# let's update the system first
sudo apt-get update -y
sudo apt-get upgrade -y

```

```
# let's create directory where we will be downloading sourece code
cd ~
mkdir -p source
cd source
```

```
# just in case you already had a sourece direcotry with cardano-node source
rm -rf cardano-node
```

```
# let's clone source code from git
git clone https://github.com/input-output-hk/cardano-node.git

cd cardano-node
git fetch --all --recurse-submodules --tags

# checking out the 1.25.1 version
git checkout tags/1.25.1

```

```
# let's update cabal
cabal clean
cabal update

# ensuring that we are using cabal 8.10.2 version
# by specifying the  particular compiler to be used.
cabal configure --with-compiler=ghc-8.10.2
```

```
# addinng extra flages for libsodium library
echo "package cardano-crypto-praos" >>  cabal.project.local
echo "  flags: -external-libsodium-vrf" >>  cabal.project.local
```

```
# now let's compile the code
cabal build all
```

Before the next step - STOP your cardano node so it doesn't lock the carano-node file for overwriting

```
# moving the freshly compiled binaries to bin folder
mkdir -p ~/.local/bin/
cp -p dist-newstyle/build/x86_64-linux/ghc-8.10.2/cardano-cli-1.25.1/x/cardi
cp -p dist-newstyle/build/x86_64-linux/ghc-8.10.2/cardano-node-1.25.1/x/carc
```

```
# let's check if we have successfully installed the latst cardano-node and c
which cardano-node && which cardano-cli
cardano-node --version
cardano-cli --version
```

you should now have similar output:

> **cardano@local**:**~/cnode/scripts**$ which cardano-node && which cardano-cli
> /home/cardano/.local/bin/cardano-node
> /home/cardano/.local/bin/cardano-cli

> **cardano@local**:**~/cnode/scripts**$ cardano-node --version
> cardano-node 1.25.1 - linux-x86_64 - ghc-8.10
> git rev 9a7331cce5e8bc0ea9c6bfa1c28773f4c5a7000f

> **cardano@local**:**~/cnode/scripts**$ cardano-cli --version
> cardano-cli 1.25.1 - linux-x86_64 - ghc-8.10
> git rev 9a7331cce5e8bc0ea9c6bfa1c28773f4c5a7000f

Now we have to update also the configuration files!

```
cd ~/cnode/config/
```

If you have custom parameters, then this will save the configuration of EKG port, PROMETHEUS port and listening address

```
export CURRENT_EKG_PORT=$(jq -r .hasEKG mainnet-config.json)
export CURRENT_PROMETHEUS_PORT=$(jq -r .hasPrometheus[1]  mainnet-config.jsc
export CURRENT_PROMETHEUS_LISTEN=$(jq -r .hasPrometheus[0]  mainnet-config.j
echo $CURRENT_EKG_PORT $CURRENT_PROMETHEUS_PORT $CURRENT_PROMETHEUS_LISTEN
```

Backing up old configs is always a good idea.

```
mkdir -p archive
mv mainnet-config.json archive/
mv mainnet-byron-genesis.json archive/
mv mainnet-shelley-genesis.json archive/
```

Let's download the latest configs(currently 5367762) - we will not touch the topology file.

```
# downloading latst  configs
export LAST_BUILD=$(curl -s https://hydra.iohk.io/job/Cardano/cardano-node/c
echo $LAST_BUILD
```

```
wget -q -O mainnet-config.json  https://hydra.iohk.io/build/${LAST_BUILD}/dc
wget -q -O mainnet-byron-genesis.json https://hydra.iohk.io/build/${LAST_BU
wget -q -O mainnet-shelley-genesis.json https://hydra.iohk.io/build/${LAST_I
#list downloaded files
ls -al mainnet*
```

Let's save the previous settings to the downloaded files:

```
# saving previous parameters to the config file:
jq .hasEKG=$CURRENT_EKG_PORT  mainnet-config.json |  sponge mainnet-config.j
jq .hasPrometheus[0]=\"${CURRENT_PROMETHEUS_LISTEN}\"  mainnet-config.json
jq .hasPrometheus[1]=$CURRENT_PROMETHEUS_PORT  mainnet-config.json |  sponge
```

That's it - now just start your node and you should be done.

> in this release Promethous metrics have changed so those who are using Grafana or other monitoring tools that rely on that data will have to update accordingly