**Procedure for Preparing a PLCnext Runtime-based Device for a Node Application**

Author: J. Kustan

Firmware Version: 2020.6.1

|  |  |  |
| --- | --- | --- |
| **Revision History** | | |
| V0\_1 | 6-3-19 | Original submission. |
| V0\_2 | 6-4-19 | Updated clock sync procedure. |
| V0\_2b | 9-4-19 | Added highlights to min requirements for Azure via Node.js. |
| V0\_2c | 2-20-20 | Updated for FW 2020.0. |
| V0\_2d | 11-6-20 | Added instructions to check and update firmware versions. |
| V0\_2e | 1-6-21 | Updated Package Manager and Nginx Web Server config instructions. |
| V0\_2f | 1-7-21 | Updated clock sync instructions. |
| V0\_3 | 2-2-21 | Updated doc to pertain to any Node app. Removed PEG specifics. |
| V0\_4 | 4-1-21 | Modified Node installation procedure to resolve an install error. |
|  | 4-1-21 | Updated Nginx Web Server instructions. |
|  | 4-8-21 | Rearranged some steps and sections for clarity. |

**Root Access**

In */etc/ssh/* @ file *sshd\_config*, uncomment line "PermitRootLogin yes". This will allow ssh as root again. Root will be required for many of the operations below. If this procedure is giving you trouble, just login as admin and *su root* (assuming you already gave the root user a password via *sudo passwd root*).

**Firmware Version (**[**firmware repo**](https://www.phoenixcontact.com/online/portal/us/pxc/product_detail_page/!ut/p/z1/3Vdbb9owGP0r7IFH40ucGCbtgaYM2lEKBVqSl8hxHJout4YApb9-DlD1sjVoQkhbIkuxre873zmOj-VAG86gHfNVMOd5kMQ8VGPLNpzJ2aTTGRIDX4_0M3RxrpudPr3BvQGBt9CGds5daOGiFwZuxrMNtJYLGRcTaeBBi1BEicGKceIPTfMHtIaAYIApQNu0PMhDCS0zifMsCUOZ1UCtPTNr32sE60QNXhDuSvm0FVg53V0--uRpo8P5dmlIG-8DSkqMZexMx85wZk7H0FKEmEO6GOEexVfXNyOKRv1uazQe90xT1-DdKpBrOI2TLFLfY7xdU1EsqjSo5JpwgddiElDhEuBSJoDOmnqL-NzDksMeOlChT46scAAenxT-HJ0W_tjFuSzdLFe08Ffw8Phot6Et1OaXTzmcFdapo7V06-g-ieQBjrdox-o3572aMpNCBiuZKaSCfpZ4S5Gfy5wH4eIil9HFe4t-tGzq-VHiyX12sgiKs-EdlhpPCuq7OSF-fvS3ggh5PF_y-VZOuSBSNUG4aoK0agkaI1Q1QaRqgnDVBFXNQ92qeahbtS3Xrdqh0GX_nSA1scwCaKVPAiQCBIqCt2XzVZX-9lJZXRsDN2qsRdRADRXa2BdzdrFOWoAduDpGPFDdfeKijsJgkW8TVX_YN2MV7-RS3MdJmMw3jkiiNIllnC-cV_515DPmI-kiQDhDgCKhgabBWoB5PqE6ksTw3AO3hUv2-SUaG9LnmEggPaLgGcaAoxYFTV26LhLcb2r4APwZPS28cVL4m9PCs9MuDtNPCj84ln3571efHuOhNBQlHkr_1kNpFDW1ooHpDD1Q63K1Xk_8yHSb-9dGe55I9_lNayJtn7NrzmCgr962fzBhAGxXn0fOVUf7w2vR_vILWMEYHw!!/?uri=pxc-iframe-integration:firmwaredownload&prodid=2404267)**)**

To check for the current firmware version of the PLCnext device, in */etc/plcnext*:

*cat arpversion*

To update the firmware, run the .exe from the latest firmware download to extract the *\*.raucb* file. Copy the *\*.raucb* file to the */opt/plcnext* directory. From the */opt/plcnext* directory, run the script to start the update:

*sudo update-axcfxx52*

**Package Manager** (<http://ipkg.nslu2-linux.org/optware-ng/buildroot-armeabihf/Packages.html>)

*wget -O - http://ipkg.nslu2-linux.org/optware-ng/bootstrap/buildroot-armeabihf-bootstrap.sh | sh*

*export PATH=$PATH:/opt/bin:/opt/sbin* (Path is not retained after reboot. Need to add own profile instructions. For PEG, use: *source ~/.bash\_profile*).

*/opt/bin/ipkg update*

*/opt/bin/ipkg list* (to see all the packages now available to you!)

**Node JS**

Install node:

*ipkg install node*

Change NTP server to sync clocks. Any external token-based connection will not function without synchronized clocks! In /etc @ ntp.conf, add:

*server <server name/ip>* (for example, time.google.com)

To initiate a manual time synchronization to a remote server (required!):

*date -s "$(curl -s --head http://google.com | grep ^Date: | sed 's/Date: //g')"*

It is also possible to synchronize the PLC’s clock to your computers clock using the PLCnext Engineering tool.

Update Node:

npm install -g n

*n stable* (or desired version instead of stable, for example: ‘10.16.0’)

**PM2**

Install PM2:

*npm install pm2 -g*

Basic commands to operate PM2 (see <https://www.npmjs.com/package/pm2> for a more complete list):

* *pm2 list*
* *pm2 stop*
* *pm2 restart*
* *pm2 delete*

Install PM2 Log Rotate:

*npm i pm2-logrotate*

*pm2 install pm2-logrotate*

All application logs using standard out and standard error out will now be funneled to PM2 and can be found in /home/root/.pm2/logs.

**Git**

Install Git:

*ipkg install git*

In */opt*:

*git clone <repo url>* (if target repo is private, you will be prompted for credentials).

In the cloned repo directory, to install all app dependencies (package.json contents):

*npm install*

**Daemonize Your App**

PM2 will monitor and keep this application alive forever with the following steps:

*pm2 start <appName>.js - -watch* (adding the watch option will restart the app on file changes)

*pm2 startup* (auto generate machine specific startup script)

*pm2 save* (saves the list of current running applications for respawn at machine boot)

**MySQL** (if your app uses an SQL database – plcnext-sparkplug.js does not!)

*ipkg install mysql*

The mySQL daemon will start listening on port 0 unless the following is done before running:

in */opt/etc/* @ file *my.cnf*, comment out "skip-networking".

To start the mysqld daemon:

(/opt/bin; ./mysqld)

Login:

*mysql -u root -p [enter]*

*[enter]* (no root password by default)

To import a DB schema, first create a placeholder DB:

*create database <db\_name>;*

Import the DB export into the placeholder DB:

*mysql -u root -p azure\_demo < <db\_name>.sql;*

**SQLite** (coming soon!)

**Nginx Web Server**

To properly serve a node application and preserve all original PLCnext web-based content, replace the nginx.conf file in /etc/nginx with the one located in the install directory of this repository. Doing so will give your application exclusive use of ports 80 and 443. All original web-based content will be rerouted to their original URL’s, but post-fixed with ‘:180’ (if the application originally used port 80) or ‘:1443’ (if the application originally used port 443).

**Conclusion**

To ensure that the node daemon starts and that PM2 is properly configured to restart your app, reboot the controller one last time. Your app should come back up with it!