ECA – Linux Development Server Build (Manual Steps)

|  |
| --- |
|  |
| * Version: 0.5 (Draft) * INTERNAL |
|  |

|  |  |
| --- | --- |
| Document control | |
|  |  |
|  |  |
| |  |  | | --- | --- | | **Author** | Steven Eyre | | **Owner** | Andrew Raison | | **Approved by** | Paul Gabriel | | **Approval date** |  | | **Effective from** |  | | **Next review date** |  | | **Source location** |  | | **Audience** | IT department | | |

|  |  |
| --- | --- |
| Change history | |
|  |  |
|  |  |
| |  |  |  |  | | --- | --- | --- | --- | | Version | Date | Amended by | Details | | 0.1 | 25/11/2019 | Steven Eyre | First draft | | 0.5 | 15/01/2020 | Steven Eyre | Updates for changed disk layout/sizings | |  |  |  |  | |  |  |  |  | |  |  |  |  | | |

Contents

[Introduction 3](#_Toc29304500)

[Overview 3](#_Toc29304501)

[Purpose/Scope 3](#_Toc29304502)

[Procedure 3](#_Toc29304503)

[Description 3](#_Toc29304504)

[Assumptions 3](#_Toc29304505)

[Centos Install Pre-Requisites 4](#_Toc29304506)

[Server Build 5](#_Toc29304507)

[Hyper-V – Create VM 5](#_Toc29304508)

[Hyper-V – install Linux 13](#_Toc29304509)

[User $HOME Post Customisation 28](#_Toc29304510)

[Ssh keys 28](#_Toc29304511)

[NPM Modules 28](#_Toc29304512)

[3RD Party Software 29](#_Toc29304513)

[NEW BUILDS 29](#_Toc29304514)

[NodeJS/Npm 29](#_Toc29304515)

[DotNetCore + AspNetCore RUNTIMES 29](#_Toc29304516)

[LEGACY 29](#_Toc29304517)

[LATEST 30](#_Toc29304518)

[Visual Studio 30](#_Toc29304519)

[LEGACY 30](#_Toc29304520)

[LATEST 31](#_Toc29304521)

[Goole Chrome (Latest) 31](#_Toc29304522)

[LEGACY/LATEST 31](#_Toc29304523)

[VISUAL STUDIO Extensions 32](#_Toc29304524)

[NPM Modules (as specific user) 33](#_Toc29304525)

[FOR NPM UPGRADE I DID THE FOLLOWING 35](#_Toc29304526)

[DEFINITIVE LIST OF NPM MODULES 35](#_Toc29304527)

[On LNLXTST01 35](#_Toc29304528)

[Postman (Latest) 35](#_Toc29304529)

[LEGACY/LATEST 35](#_Toc29304530)

[Ca-Certificates – Write enable for Devs 36](#_Toc29304531)

[LEGACY/LATEST 36](#_Toc29304532)

[Ca-Certificates – Write enable for Devs 36](#_Toc29304533)

[LATEST 36](#_Toc29304534)

[Related documents 37](#_Toc29304535)

[Related documents 37](#_Toc29304536)

|  |  |
| --- | --- |
| Introduction | |
|  |  |
|  |  |
| Overview ECA brought in-house the StarRate development environment. This necessitates several linux Servers (Both Desktop/Gnome and management servers be built. Previously builds were by hand/knowledge specific to Starrate  This document covers a manual build to ECA/industry standards on ECA hardware. Purpose/Scope Step by Step manual build procedure for Linux Desktops | |

|  |  |
| --- | --- |
| Procedure | |
|  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Description ***Centos*** *is an open source Linux distribution* [*https://www.centos.org/*](https://www.centos.org/) *. It is derived from* [*Red Hat Enterprise Linyx (RHEL)*](https://www.centos.org/)  *sources.*  Since March 2004, CentOS Linux has been a community-supported distribution derived from sources freely provided to the public by Red Hat. As such, CentOS Linux aims to be functionally compatible with RHEL. We mainly change packages to remove upstream vendor branding and artwork. CentOS Linux is no-cost and [free to redistribute](https://www.centos.org/legal). Assumptions While this document covers the build/maintenance procedures – below assumptions are made:   1. A broad understanding of IT standards/methodologies. 2. Basic Linux administration skills. 3. Familiarity with ECA procedure and standards.   Whwthr Centos Install Pre-Requisites **[URL]** [**https://github.com/spacewalkproject/spacewalk/wiki/HowToInstall**](https://github.com/spacewalkproject/spacewalk/wiki/HowToInstall)  **SERVER SPECS**   * Hyper V VM * 1-2CPU * 2GB RAM minimum, 6GB recommended * 16GB Disk for OS/Root * Centos 7.7 DVD/ISO (*or later iteration if available*)   + **[URL]** [**Centos Download**](https://www.centos.org/download/) * Make sure your underlying OS is fully up-to-date (build from DVD should suffice)   **FIREWALL/PORTS**   * Outbound open ports 80, 443   **SOFTWARE REPOSITORIES**  Check re DVD and OS   |  |  | | --- | --- | | **repo id** | **Repo name** | | epel/x86\_64 | Extra Packages for Enterprise Linux 7 | | group\_spacewalkproject-java-packages | copr repo for java packages @spacewalkproject | | spacewalk/x86\_ | Spacewalk |  |  |  | | --- | --- | | Server Build | | |  |  | |  |  |  Hyper-V – Create VM Each server requires a new VM creating on the HyperV platform (lnvh7.eca-international.local)  **Remote Desktop to the server:**    **Server Manager**  **Windows Start (Bottom Left)–> Server Manager**    **Server Manager, Tools, Hyper-V Manager**    **Hyper-V Manager**  1st Check that enough CPU/RAM/Storage are available on the lnvh7.eca-international.local host. (recommend check with management if this is the target Hyper-V host)    **Hyper-V Manager, New, Virtual Machine(Top Right Panel Actions)**    **New, Virtual Machine Wizard, VM Name and Location of VM Store**    Specify Name of Machine  Change Virtual Machine Store/Location to be **D:\** (a folder with VM name is created where data files are stored)  **New, Virtual Machine Wizard, Specify Generation 2**    **New, Virtual Machine Wizard, Specify Generation**  6Gb Ram = 6144MB    **New, Virtual Machine Wizard, Configure Networking**  Virtual Guests    **New, Virtual Machine Wizard, Connect Virtual Hard Disk**  **Disk = 32GB**    **New, Virtual Machine Wizard, Installation Options**  Install Source/DVD/ISO (Downloaded from Centos Mirrors)    **New, Virtual Machine Wizard, Summary**    **Hyper-V – VM, Settings, Security (updates before booting !!!)**  Select VM you just created) Right Click, Settings, Security Tab  Update to be **Microsoft UEFI Certificate Authority, click apply.**    **Hyper-V – VM, Settings, Connect**   Hyper-V – install Linux **Hyper-V – VM, Settings, Connect (cont’d)**    **Click on Start**  **Linux Install**  Click in window, and user Arrow keys to move/highlight **Install Centos**    **Linux Install, Lang & Keyboard**  Select **English (United Kingdom)** from Right hand Panel, **Select Continue**.    **Note Timezone/Keyboard are now updated to be GB**    **Linux Install, Software Selection, Software Selection, Gnome Desktop**    **Select Done**  **DISK LAYOUT**  **Linux Install, System, Installation Destination sda / 28GB**  Select 28GB/SDA device (should only be 1 x disk)  **Select Other Storage Options, Select I will Configure partitioning**    We then need to re-layout the partition as we want to make 2 x additional Filesystems (legacy they had 1 x FS) But going forward we want them co nfigure 3 x FS.  / # 20.5GB  /home # 6.0GB  /etc/ECA # 1.5GB  **Select Done**  **See next page**  **Next screen will then let you layout the Filesystems as required**  **Leave Partitioning Scheme as LVM**    **Select “Select here to Create them Automatically**    **We then need to modify/free space to add the extra FS**  **Select the / partition – and change its size down to 16.5 Gib**    **Once size has been changed – click on any other partition so the Available Space**  **Now add the additional mountpoints as below using the + symbol at bottom of the Layout**  **/home – 9GB**  **Ignore 6gb in screenshot**    **/etc/ECA - 1.5G**    **Final Layout should look same as below**    **N.B. /HOME should be 9GB**  **Select Done**  **Summary of changes/Accept Changes**    **Linux Install, Exclamation Marks cleared from sections as you populate them**    **Linux Install, Network and Hostname**  Update Host Name (bottom Left, Apply)    **Configure (bottom Right)**    **Select IPv4 Settings (on row where Ethernet is underlined in blue text)**    **Change:**  **Method: Manual**  **DNS Servers: 192.168.130.2** (Check with network admin for the IP/Subnet you have been provided)  **Search Domains: domain01-starrate.intranet.co.uk**  **Addresses**  **Get IP/Netmask/Subnet (N.B)**    **Save, Enable Ethernet**    **IP 192.168.130.67/24**  **GW 192.168.130.1**  **Linux Install, begin Installation (Root Password, Add “admin” user)**  Select Begin Installation (Bottom Right)    **Linux Install, root Password, User Creation (“admin” user)**    **Select Root Password**    **Enter desired root passwd, Select Done Top Left**  **Select User Creation**    Select Make this user administrator  Current admin password is weak – so you will have to **Click Done twice**  **Reboot** |
|  |
|  |

**REGISTER WITH SPACEWALK SERVER**

Bit of chicken and Egg here ….. To register the server with Spacewalk – you have to have the software installed (and it doesn’t come on the DVD …..)

For ease – I’ve added all the software into a single place

**On Server you wish to register with Spacewalk**

**# cd /etc/yum.repos.d;**

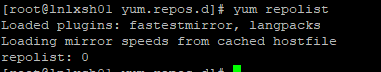
**# mkdir BACKUP**

**# cp \* BACKUP**

**# yum clean all**

**# yum repolist**

**# yum repolist**



**# mkdir /var/tmp/rhn**

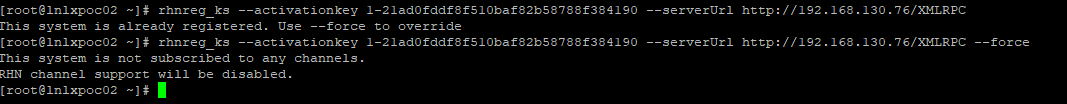
**# cd /var/tmp/rhn**

**# scp admin@192.168.130.76:/var/satellite/RHN-Register/\* .**

**# yum -y install m2crypto-0.21.1-17.el7.x86\_64.rpm python-gudev-147.2-7.el7.x86\_64.rpm python-hwdata-1.7.3-4.el7.noarch.rpm rhn-check-2.0.2-24.el7.x86\_64.rpm rhn-client-tools-2.0.2-24.el7.x86\_64.rpm rhnlib-2.5.65-8.el7.noarch.rpm rhnsd-5.0.13-10.el7.x86\_64.rpm rhn-setup-2.0.2-24.el7.x86\_64.rpm yum-rhn-plugin-2.0.1-10.el7.noarch.rpm pyOpenSSL-0.13.1-4.el7.x86\_64.rpm python-ethtool-0.8-8.el7.x86\_64.rpm libnl-1.1.4-3.el7.x86\_64.rpm python-dmidecode-3.12.2-3.el7.x86\_64.rpm --nogpgcheck**

[CMD]

**# rhnreg\_ks --activationkey 1-21ad0fddf8f510baf82b58788f384190 --serverUrl** [**http://192.168.130.76/XMLRPC**](http://192.168.130.76/XMLRPC)



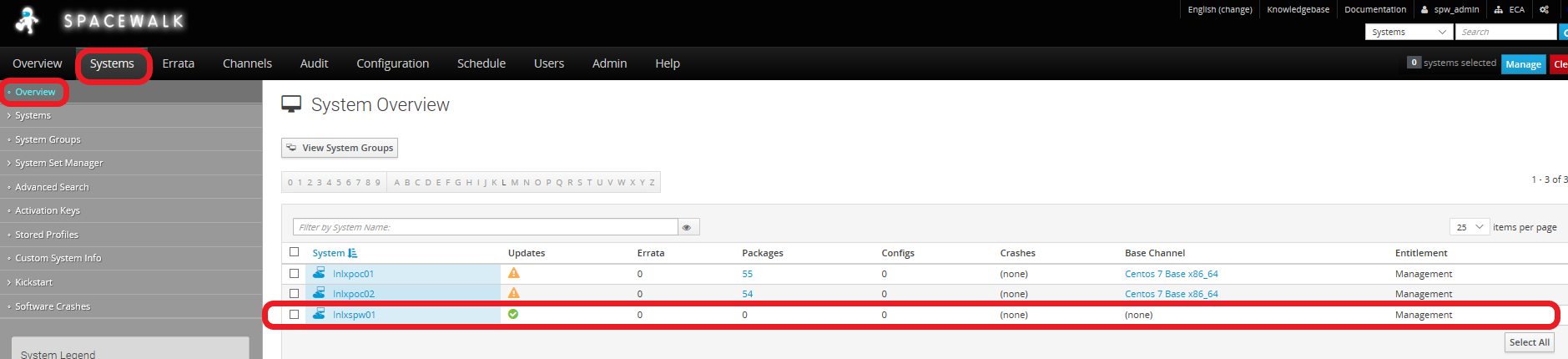
This can take a minute of 2 – so please be patient.

rpm -Uvh <http://192.168.130.76/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm>

**INSTALL KEY**

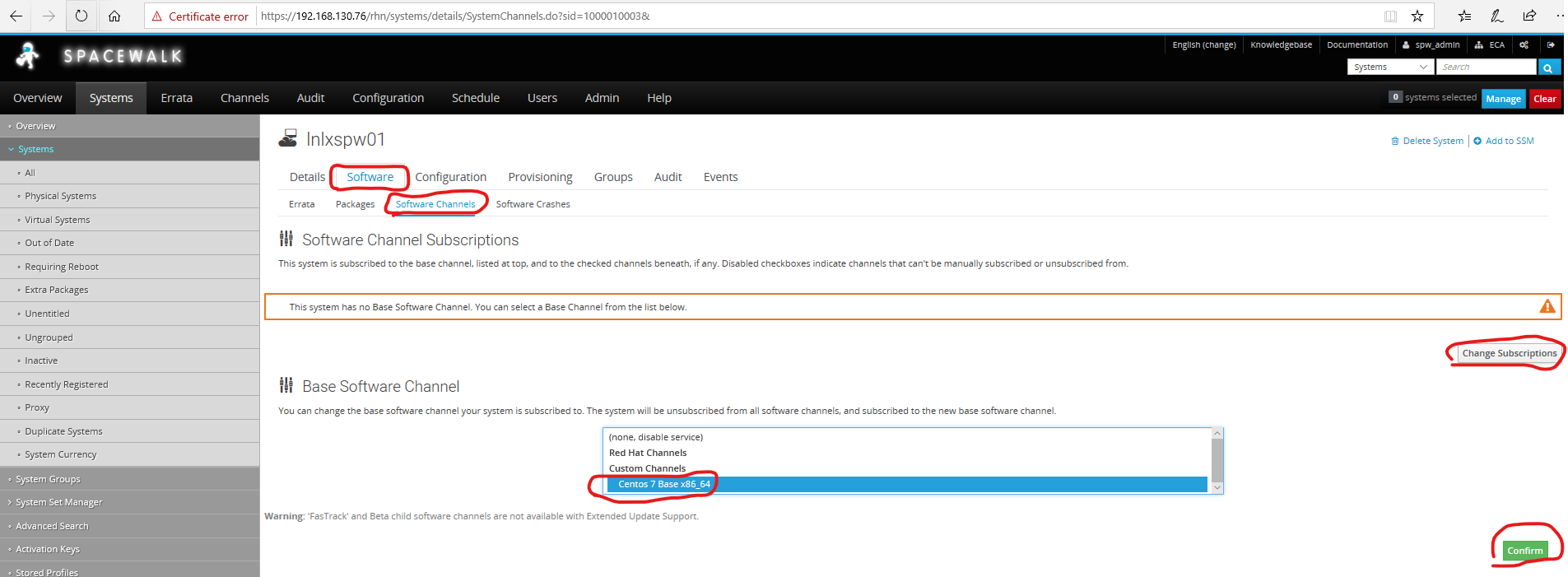
**Check Server listed on Spacewalk Server**

**Systems (Top)–> Overview (Left side pane) –> Systems Overview.**



Add Server to Software Channel

**Systems (Top)–> Overview (Left side pane) –> Select Hostname -> Software (Main panel) -> Software Channels (Main panel)**



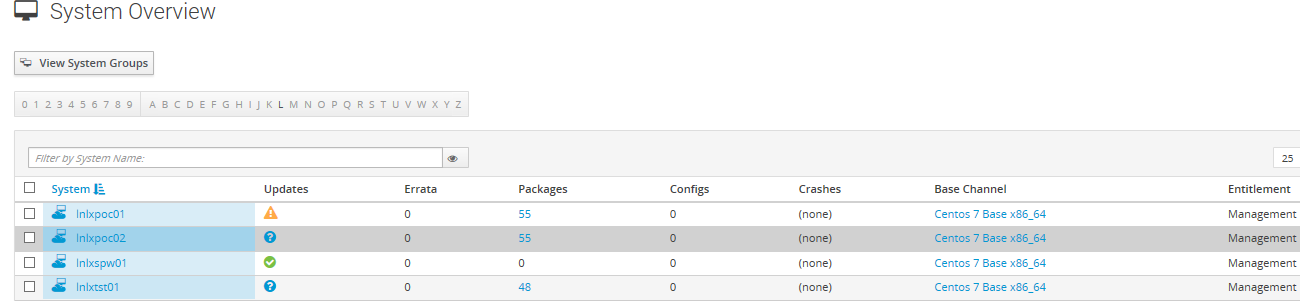
* Select Base Software CHANNEL (Centos 7 Base x86\_64)
* Select Confirm



* Select **Modify Base Software Channel**

**Check Server listed on Spacewalk Server**

**Systems (Top)–> Overview (Left side pane) –> Systems Overview.**



Server now shows as being registered to channel, and should show any down-revision packages (this can take upto 15 minutes to populate dependent upon Spacewalk server/Client host itself)

**INSTALL REMAINING SOFTWARE VIA SPACEWALK SERVER**

On Target server: (cut’n paste bold text to command prompt)

# yum check-update

# yum-update

# yum-update # (make sure not more updates)

# init 6

**# cd /var/tmp/RHN-Register**

**# for i in `cat lnlxpoc01.rpmList`**

**do**

**yum install -y $i --nogpgcheck**

**done**

This will take quite a bit of time as it is working though a list of software – and installing it, along with any dependencies.

**CHECK INSTALLED vs. SOFTWARE LIST**

**# for i in `cat lnlxpoc01.rpmList`**

**do**

**RPM=`rpm -qa --qf '%{name}\n' $i`**

**echo $i = $RPM 2>&1 >> /tmp/rpmQuery.out**

**done**

ADD RHNCFG PACKAGES TO SPACEWALK SERVER

#rpm -Uvh <https://copr-be.cloud.fedoraproject.org/results/@spacewalkproject/spacewalk-2.8-client/epel-7-x86_64/00742644-spacewalk-repo/spacewalk-client-repo-2.8-11.el7.centos.noarch.rpm>

rpm -Uvh <http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install rhncfg\*

**SETUP REPOS**

# yum install -y yum-plugin-tmpreo

# yum install -y spacewalk-repo --tmprepo=https://copr-be.cloud.fedoraproject.org/results/%40spacewalkproject/spacewalk-2.9/epel-7-x86\_64/repodata/repomd.xml -nopgpg

# yum install -y yum-plugin-tmprepo

#yum install -y spacewalk-repo --tmprepo=https://copr-be.cloud.fedoraproject.org/results/%40spacewalkproject/spacewalk-2.9/epel-7-x86\_64/repodata/repomd.xml –nogpg

**INSTALL SOFTWARE**

Add EPEL REPO

[CMD]

# yum install epel-release

**ADD USERS**

NickG

# useradd -c “Nick Gray“ -d /home/NickG -u 1001 -s /bin/bash NickG

ECAReleaseBuilder

# useradd -c “ECAReleaseBuilder“ -d /home/ECAReleaseBuilder -u 1001 -s /bin/bash ECAReleaseBuilder

# useradd -c "Sajed Hussain" -d /home/SajedH -u 1001 -s /bin/bash SajedH

# groupadd xrdpusers

# usermod -G SajedH,xrdpusers SajedH

# id SajedH

**REMOTE DESKTOP**

**[CMD]**

**# yum install**

**xrdp-0.9.11-5.el7.x86\_64.rpm**

**xrdp-selinux-0.9.11-5.el7.x86\_64.rpm**

**xorgxrdp-0.2.11-1.el7.x86\_64.rpm**

**selinux-policy-3.13.1-252.el7.1.noarch.rpm**

**selinux-policy-targeted-3.13.1-252.el7.1.noarch.rpm**

**selinux-policy-devel-3.13.1-252.el7.1.noarch.rpm**

**# for i in 3389/tcp 5000/tcp 5001/tcp 4200/tcp 4300/tcp 5003/tcp 5103/tcp; do firewall-cmd --permanent --zone=public --add-port=$i ; done**

# groupadd xrdpusers

# usermod -G xrdpusers $user # Where $user will be the account you are adding)

# id $user

***e.g.***

***[root@lnlxap01 ~]# id AlexP***

***uid=1001(AlexP) gid=1001(AlexP) groups=1001(AlexP),1002(xrdpusers)***

# systemctl start xrdp

# systemctl status xrdp

# systemctl enable xrdp

# for i in 3389/tcp 5000/tcp 5001/tcp 4200/tcp 4300/tcp 5003/tcp 5103/tcp; do firewall-cmd --permanent --zone=public --add-port=$i ; done

,

# User $HOME Post Customisation

## Ssh keys

## NPM Modules

**LEGACY**

*# cp package.json $HOME (Gordon said to use version on existing hosts)*

*# npm install (Uses package.json as source installs packages into $HOME/node-modules)*

# **3RD Party Software**

## NEW BUILDS

## NodeJS/Npm

**As Root**

Install Node + Npm - Node.js includes npm! - https://nodejs.org/en/ - CLICK Other downloads - CLICK Linux Binaries x64 64-bit - Select latest LTS version (long term support)

<https://nodejs.org/en/download/> **# Linux Binaries (x64)**

**nodejs** rpm is available through Spacewalk server/repos os used that instead.

## DotNetCore + AspNetCore RUNTIMES

### LEGACY

*Install DotNetCore + AspNetCore RUNTIMES - https://packages.microsoft.com/rhel/7/prod/ - LOOK AT LIST OF dotnet-runtime-\*.rpm TO FIND LATEST VERSION TO USE (eg 2.2.4) - save files to default \Downloads folder - dotnet-runtime-deps-2.2.4-rhel.7-x64.rpm - dotnet-host-2.2.4-x64.rpm - dotnet-hostfxr-2.2.4-x64.rpm - dotnet-runtime-2.2.4-x64.rpm (need to install first) - aspnetcore-runtime-2.2.4-x64.rpm*

[*https://packages.microsoft.com/rhel/7/prod/*](https://packages.microsoft.com/rhel/7/prod/)

***# yum install aspnetcore-runtime-2.2.4-x64.rpm dotnet-runtime-deps-2.2.4-rhel.7-x64.rpm dotnet-host-2.2.4-x64.rpm dotnet-hostfxr-2.2.4-x64.rpm dotnet-runtime-2.2.4-x64.rpm -y***

*aspnetcore-runtime-2.2 x86\_64 2.2.4-1*

*dotnet-host x86\_64 2.2.4-1*

*dotnet-hostfxr-2.2 x86\_64 2.2.4-1*

*dotnet-runtime-2.2 x86\_64 2.2.4-1*

*dotnet-runtime-deps-2.2 x86\_64 2.2.4-1*

### LATEST

**Install DotNetCore + AspNetCore** **+ dotnet-sdk** - <https://packages.microsoft.com/rhel/7/prod/> ***save files to default \Downloads and then into /var/satellite/3rd-Party folder on lnlxspw01***

aspnetcore-runtime-3.1.0-x64.rpm

dotnet-host-3.1.0-x64.rpm

dotnet-hostfxr-3.1.0-x64.rpm

dotnet-runtime-3.1.0-x64.rpm

dotnet-runtime-deps-3.1.0-rhel.7-x64.rpm

dotnet-sdk-3.1.100-x64.rpm

dotnet-apphost-pack-3.1.0-x64.rpm

dotnet-targeting-pack-3.1.0-x64.rpm

aspnetcore-targeting-pack-3.1.0.rpm

netstandard-targeting-pack-2.1.0-x64.rpm

**[root@lnlxpoc01 3rd-Party]# yum install dotnet-host-3.1.0-x64.rpm aspnetcore-runtime-3.1.0-x64.rpm dotnet-hostfxr-3.1.0-x64.rpm dotnet-runtime-3.1.0-x64.rpm dotnet-runtime-deps-3.1.0-rhel.7-x64.rpm dotnet-sdk-3.1.100-x64.rpm dotnet-apphost-pack-3.1.0-x64.rpm dotnet-targeting-pack-3.1.0-x64.rpm aspnetcore-targeting-pack-3.1.0.rpm netstandard-targeting-pack-2.1.0-x64.rpm -y**

## Visual Studio

### LEGACY

*Install VSCode - https://code.visualstudio.com - CLICK "Other platforms" - CLICK "rpm" (under Linux - doesn't look like a link BUT IT IS!!) - DON'T CLICK .tar.gz \ 64 bit!!!!!!!!! THIS IS JUST BINARY + SUPPORT FILES - Download \ Save file*

[*https://code.visualstudio.com*](https://code.visualstudio.com)

*https://code.visualstudio.com/#*

***# yum install code-1.40.1-1573664332.el7.x86\_64.rpm –nogpgcheck -y***

*============================================================================*

*Package Arch Version*

*============================================================================*

*Installing:*

*code x86\_64 1.40.1-1573664332.el7*

### LATEST

*Install VSCode - https://code.visualstudio.com - CLICK "Other platforms" - CLICK "rpm" (under Linux - doesn't look like a link BUT IT IS!!) - DON'T CLICK .tar.gz \ 64 bit!!!!!!!!! THIS IS JUST BINARY + SUPPORT FILES - Download \ Save file*

***# yum install code-1.41.1-1576681965.el7.x86\_64.rpm –nogpgcheck -y***

*============================================================================*

*Package Arch Version*

*============================================================================*

*Installing:*

*code x86\_64* 1.41.1-1576681965.el7

## Google Chrome (Latest)

### LEGACY/LATEST

**Install Chrome - download following dependencies (worked Chrome 69+74) -** http://download.fedoraproject.org/pub/epel/7/x86\_64/Packages/l/ - libdbusmenu-16.04.0-2.el7.x86\_64.rpm - libdbusmenu-gtk2-16.04.0-2.el7.x86\_64.rpm - libdbusmenu-gtk3-16.04.0-2.el7.x86\_64.rpm - libindicator-gtk3-12.10.1-5.el7.x86\_64.rpm - libindicator-12.10.1-5.el7.x86\_64.rpm - libappindicator-12.10.0-11.el7.x86\_64.rpm - libappindicator-gtk3-12.10.0-11.el7.x86\_64.rpm - https://www.google.com/chrome/?system=true&standalone=1 - CLICK "Download Chrome" (auto-determines best package)

**liberation-fonts**

**# yum install liberation-fonts-1.07.2-16.el7.noarch.rpm liberation-fonts-common-1.07.2-16.el7.noarch.rpm fontpackages-filesystem-1.44-8.el7.noarch.rpm liberation-narrow-fonts-1.07.2-16.el7.noarch.rpm liberation-sans-fonts-1.07.2-16.el7.noarch.rpm liberation-serif-fonts-1.07.2-16.el7.noarch.rpm --nogpgcheck -y**

[libdbusmenu-16.04.0-2.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libdbusmenu-16.04.0-4.el7.x86_64.rpm)

[libdbusmenu-gtk2-16.04.0-2.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libdbusmenu-gtk2-16.04.0-4.el7.x86_64.rpm)

[libdbusmenu-gtk3-16.04.0-2.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libdbusmenu-gtk3-16.04.0-4.el7.x86_64.rpm)

[libindicator-gtk3-12.10.1-5.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libindicator-gtk3-12.10.1-6.el7.x86_64.rpm)

[libindicator-12.10.1-5.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libindicator-12.10.1-6.el7.x86_64.rpm)

[libappindicator-12.10.0-11.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libappindicator-12.10.0-13.el7.x86_64.rpm)

[libappindicator-gtk3-12.10.0-11.el7.x86\_64.rpm](http://mirror.centos.org/centos/7/os/x86_64/Packages/libappindicator-gtk3-12.10.0-13.el7.x86_64.rpm)

# yum install libdbusmenu-16.04.0-4.el7.x86\_64.rpm libdbusmenu-gtk2-16.04.0-4.el7.x86\_64.rpm libdbusmenu-gtk3-16.04.0-4.el7.x86\_64.rpm libindicator-gtk3-12.10.1-6.el7.x86\_64.rpm libindicator-12.10.1-6.el7.x86\_64.rpm libappindicator-12.10.0-13.el7.x86\_64.rpm libappindicator-gtk3-12.10.0-13.el7.x86\_64.rpm -y --nogpgcheck

=================================================================================================================

Package Arch Version

=================================================================================================================

Installing:

libappindicator x86\_64 12.10.0-13.el7

libappindicator-gtk3 x86\_64 12.10.0-13.el7

libdbusmenu x86\_64 16.04.0-4.el7

libdbusmenu-gtk2 x86\_64 16.04.0-4.el7

libdbusmenu-gtk3 x86\_64 16.04.0-4.el7

libindicator x86\_64 12.10.1-6.el7

libindicator-gtk3 x86\_64 12.10.1-6.el7

**[CMD]**

# vi /etc/yum.repos.d/Google-chrome.repo

#Add

[google-chrome]

name=google-chrome

baseurl=http://dl.google.com/linux/chrome/rpm/stable/x86\_64

enabled=1

gpgcheck=1

gpgkey=https://dl.google.com/linux/linux\_signing\_key.pub

# yum install google-chrome-stable

# mv /etc/yum.repos.d/\*.repo /etc/yum.repos.d/BACKUP

# yum clean all

# yum repolist

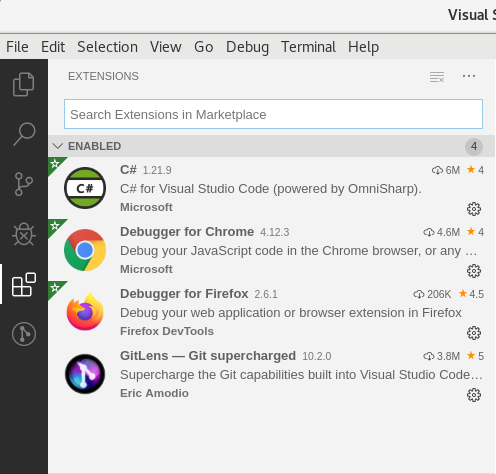
## VISUAL STUDIO Extensions

#GitLens - Visual Studio addon

Ctrl+Shift K - search Gitlens

# Visual Code Chrome & Firefox Extensions

Visual Studio Add Ons



## NPM Modules (as specific user)

As per Nick’s email dated 19/12/2019 @ 19:11 hrs.

* Update a machine (possibly lnlxtst01?) to have latest versions of some key tools, in particular:
  + Angular 8.2
  + Typescript 3.4

**[CMD]**

npm install @angular-devkit/build-angular

npm install @angular-devkit/build-ng-packagr

npm list | grep -i angular-devkit/core

npm install @angular-devkit/core

npm install @angular-devkit/schematics

npm install @angular/animations

npm install @angular/cdk

npm install @angular/cli

npm list | grep -i angula

npm install @angular/common

npm install @angular/compiler

npm install @angular/compiler-cli

npm install @angular/core

npm install @angular/forms

npm install @angular/language-service

npm install @angular/material

npm install @angular/platform-browser

npm install @angular/platform-server

npm install @angular/pwa

npm install @angular/router

npm install @angular/service-worker

npm install @angular/upgrade

npm install @ckeditor/ckeditor5-autosave

npm install @ckeditor/ckeditor5-build-inline

npm install @ng-bootstrap/ng-bootstrap

npm install @ng-select/ng-select

npm install @ngx-pwa/local-storage

npm install angular-in-memory-web-api

npm install Async

npm install async

npm install bourbon

npm install codelyzer

npm install core-js

npm install highcharts

npm install karma-chrome-launcher

npm install ngx-bootstrap

npm install ngx-toastr

npm install screenfull

npm install sweetalert2

npm install ts-node

npm install tsickle

npm install typescript

npm install zone.js

## FOR NPM UPGRADE I DID THE FOLLOWING

NPM does’t handle upgrades very well (if you just run npm upgrade it seems not throw errors rather than do anything) Hence having to more or less dictate and say INSTALL a specific version (could be why Gordon had it setup the way it was)

# vi /tmp/steve (Add the npm install lines form above)

# for i in `cat /tmp/steve | awk '{print $3}'`; do npm list | grep -i $i | grep -v ERR 2>&1 >> /tmp/steve1; done

# cat /tmp/steve1 | sed s'/ //g' | sed s'/├─//g' | sed s'/─//g' | sed s'/┬//g' | sed s'/│//g' | sed s'/└//g' | grep -v deduped | grep -v UNMETPEERDEPENDENCY | sed s'/extraneous//g'

On server to be upgraded

# vi /tmp/steve1 (paste in output of above)

# for i in `cat /tmp/steve1`; do npm install $i; done

## DEFINITIVE LIST OF NPM MODULES

### On LNLXTST01

# npm list --depth=0 | grep ├── | sed s'/ //g' | sed s'/├──//g'

Take NPM list – and manually run through on POC servers:

## Postman (Latest)

### LEGACY/LATEST

<https://www.getpostman.com/downloads/>

Linux .tar.gz

As user

**[CMD]**

# [ECAReleaseBuilder@lnlxpoc02 ~]$ tar -xvf /var/tmp/Postman-linux-x64-7.14.0.tar.gz

## Ca-Certificates – Write enable for Devs

### LEGACY/LATEST

**[CMD]**

# chmod 757 /etc/pki/ca-trust/source/anchors/

# chmod -R 757 /etc/pki/ca-trust/extracted

## Ca-Certificates – Write enable for Devs

### LATEST

Requirement to add RW access – but with varying levels of access based upon roles 0 easiest way to retrofit this was use ACL’s (not ideal but request came late on into new builds)

**Users**

|  |  |
| --- | --- |
| **User** | **ACL Group/Role** |
| Nick Gray | DevMan |
| Rune Stamselberg | DevMan |
| Wendy Gasperazzo | DevLead |
| Mike McAllistair \* | DevLead |
| Alexander Paskhin # | dev |
| Sajed Hussain # | dev |
| Remy Roux dit Buisson \* | dev |
| Istvan Kun \* | dev |
| Glenn Knight | dev |
| Sergej Sapalov | dev |
|  |  |
|  |  |

**Roles**

Devs – devs Group – Will have RW to all other devs home areas (excluding DevMan/DevLead home areas)

Mike/Wendy – DevLead Group – Will have RW to all Devs/Each Others (excluding DevMan home areas)

Nick/Rune – DevMan group – will have RW to everything

This was added under the Ansible Playbook for new builds – and is not part of the retrofit/copy of the original gnome builds.

|  |  |  |
| --- | --- | --- |
| Related documentsRelated documents | | |
|  |  | |
|  |  | |
|  | |  |