**LINUX BUILD NOTES**

(Ref PDF provided by **© Gordon Reffell 2019** )

**Create Linux DEV machine VSCode**

The latest Linux dev machines were created using **CentOS 7.6 1810**, a downstream version of Red Hat Enterprise Linux (RHEL).

The installation steps and configuration of the following software is highly dependent on the particular Linux distribution and on the version of the distribution.

**yum packages to install**

- @^minimal #

- @core #

- policycoreutils-python #

- bind-utils #

- nmap #

- @x11 # XWindow Server

- gnome-classic-session #

- @fonts #

- gnome-terminal # Terminal for gnome Desktop

- control-center # Gnome Control Center

- liberation-mono-fonts #

- firefox #

- vinagre # VNC/SSH/RDP Gnome Client

- policycoreutils-gui # SElinux Policy gui

- setroubleshoot # SELinux Plugin for troubleshooting ….

- gedit # Gnome text editor

- evince # Gnome document viewer (XML/PDF etc)

- tcpdump #

- wireshark # Q if needed (in what instances are this used)

- iptraf-ng # Q (visual network monitoring utility)

- xorg-x11-fonts-Type1 #

Gather packages installed on current systems

**CMD** - **rpm -qa --qf '%{name}\n' | sort -u >> mykickstart.ks**

**File created on lnvh7/**

yum packages to uninstall – chrony - gnome-boxes

Install a remote desktop server - either VNC or XRDP - needed for desktop access from Windows - both unpleasant to set up - both get problems in production - unpleasant to support - work out best config for each new distribution version

Install unzip - yum -y install unzip

# **3RD Party Software**

**COPY FROM SPACEWALK SERVER LEGACY 3rd-PARTY RPMS**

# [root@lnlxspw01 ]# **cd /var/satellite**

# [root@lnlxspw01 satellite]# **pwd**

/var/satellite

[root@lnlxspw01 satellite]# scp -r 3rd-Party-Legacy ***hostname\_server***:/var/tmp

## **Nodejs/npm**

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/dist/v12.13.1/node-v12.13.1-linux-x64.tar.xz>

**? binaries/Libraries in place – but no RPM – Q how it was installed.**

**[CMD]**

**# cd /var/tmp/3rd-Party/Legacy**

**# pwd**

**# yum install nodejs-10.15.3-1nodesource.x86\_64 --nogpgcheck**

## **DotNetCore + AspNetCore RUNTIMES**

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/>

**# cd /var/tmp/3rd-Party/Legacy**

**# pwd**

**# 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

## **DotNetCore SDK**

Install DotNetCore SDK - find version number of SDK that matches latest runtime - just find number, don't download for now - HAS STRANGE VERSION NUMBERING!! - https://dotnet.microsoft.com/download/archives - CLICK on .NET implementation eg ".NET Core downloads" (NOT ".NET Framework downloads") - CLICK on current version eg "2.2" (NOT preview version) - https://dotnet.microsoft.com/download/dotnet-core/2.2 - .NET Core \ SELECT "Current" version (not "Preview" or "LTS") - find latest version of SDK that matches the runtime version above - eg "SDK 2.2.203" - DON'T DOWNLOAD FROM HERE!! (JUST FIND VERSION NUMBER) - download SDK specially created for RHEL - https://packages.microsoft.com/rhel/7/prod/ - download .RPM for SDK - eg dotnet-sdk-2.2.203-x64.rpm

<https://packages.microsoft.com/rhel/7/prod/dotnet-sdk-2.2.203-x64.rpm>

**# cd /var/tmp/3rd-Party/Legacy**

**# pwd**

**# yum install dotnet-sdk-2.2.203-x64.rpm –nogpgcheck -y**

Package Arch Version

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

Installing:

dotnet-sdk-2.2 x86\_64 2.2.203-1

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/#

**# cd /var/tmp/3rd-Party/Legacy**

**# pwd**

**# 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

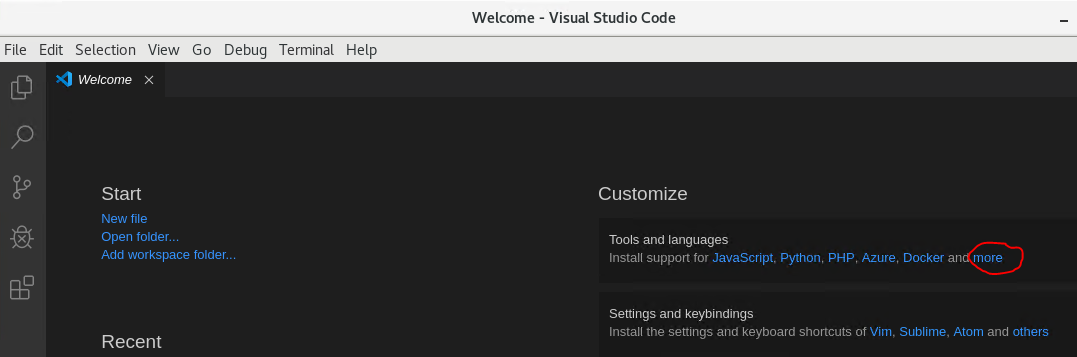
Install C# extension for VSCode - <https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp>

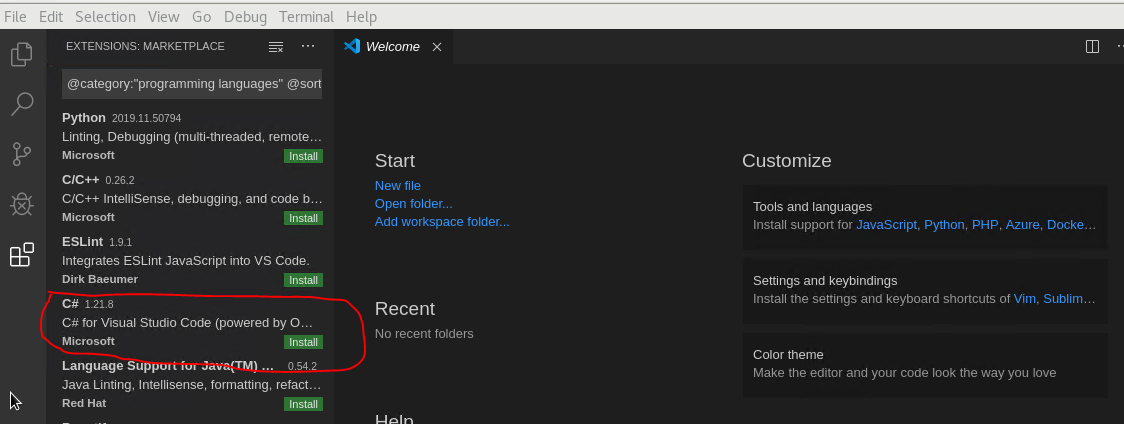
- more files downloaded on first use - dotnet new console -o helloworld - code helloworld - NOW downloads/installs extra files!

<https://github.com/OmniSharp/omnisharp-vscode>

**Query how to install**

Installed from within the Visual Studio Application on the Desktop





**NPM MODULES**

Install Angular CLI + TypeScript - npm -g install @angular/cli - npm -g install typescript

**GLOBAL INSTALL**

***As root user on the target\_server***

# npm -g install @angular/cli@7.3.9

# npm install -g typescript@3.2.4

**LOCAL INSTALL**

# su - $USER **# Name of developer**

[SajedH@lnlxsh01 ~]$ npm install @angular/cli@7.3.9

[SajedH@lnlxsh01 ~]$ npm install typescript@3.4.5

[SajedH@lnlxsh01 ~]$ cp /var/tmp/3rd-Party-Legacy/NickG.package.json package.json

[SajedH@lnlxsh01 ~]$ npm install

**List of installed NPM MODULES from gnomeng03 server**

# npm list --depth=1 >> /tmp/npmList.out

# cat /tmp/npmList.out | sed 's/[^a-zA-Z0-9@. ]//g' | sed 's/[deduped]//g'

# /tmp/npmListCleaned.out

# scp /tmp/npmListCleaned.out admin@192.168.130.68:/var/tmp/gnomeng03.npmListCleaned.out

Then on target server:

**[CMD]**

# for i in `cat /var/tmp/gnomeng03.npmListCleaned.out`; do npm install $i; done

**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

Create /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

Install latest Node-Sass binding.node file - Angular uses SASS - needs latest version to work with latest version of NODE - get all versions 57 and after - https://github.com/sass/node-sass/releases/ - linux-x64-\*\_binding.node - there are only a few AND THEY'RE SMALL so get all - eg 57, 59, 64, 67 - copy to <$HOME>/<APPLICATION>/node\_modules/node-sass/vendor/linux-x6457/binding.node etc

**NOTHING TO DO BASED UPON NICKG’s gnome host compared vs. new hostS**

~/node\_modules/node-sass/vendor/linux-x64-64 $ ls -altr

-rw-rw-r--. 1 NickG NickG **3545776** Jun 5 2019 binding.node

**Which matches what is already installed on teh new serveres**

[AlexP@lnlxap01 linux-x64-64]$ pwd

/home/AlexP/node\_modules/node-sass/vendor/linux-x64-64

[AlexP@lnlxap01 linux-x64-64]$ ls -altr

-rw-rw-r--. 1 AlexP AlexP **3545776** Dec 13 12:21 binding.node

**Think developers have to do this however**

Install key .NET Standard NuGet packages - create dummy project to force download to central folder - IMPORTANT: can only run "dotnet add package" from WITHIN a project!! - dotnet new classlib -o MyLibStd (defaults to --framework netstandard2.0) - cd MyLibStd - dotnet add package System.Runtime.Caching (for TaxcelEngine) - dotnet add package System.Data.SqlClient (adds related packages, eg System.Buffer + System.Vectors) - dotnet add package MySql.Data - dotnet add package Microsoft.Data.Sqlite - dotnet add package automapper - dotnet add package automapper.extensions.microsoft.dependencyinjection (adds related packages) - dotnet add package microsoft.entityframeworkcore (adds related packages) - dotnet add package newtonsoft.json - dotnet add package njsonSchema

Install key .NET Core NuGet packages - create dummy project to force download to central folder - IMPORTANT: can only run "dotnet add package" from WITHIN a project!! - dotnet new classlib -f netcoreapp2.0 -o MyLibCore - cd MyLibCore

© Gordon Reffell 2019

- dotnet add package System.Data.SqlClient (downloads System.Buffer + System.Vectors) - dotnet add package System.Configuration.ConfigurationManager (.NET Core package for System.Runtime.Caching)

Install Git - yum -y install git

**Install Postman** –

https://www.getpostman.com/downloads/ - looks for version for Linux x64 - download Postman-linux-x64-<VERSION>.tar.gz

<https://dl.pstmn.io/download/latest/linux64>

ON NickG’s machine

***On NickG’s machine PREVIOUS VERSION WAS***

***# cd $HOME/Postman/app***

***# cat version***

***v1.8.8***

***Told we can just download/install latest***

# su - $USER

# tar -xvf /var/tmp/RPMs/Postman-linux-x64-7.11.0.tar

Creates $HOME/Postman directory

Install **ApacheWebServer** - yum -y install httpd

Install MariaDB (=MySQL) - yum -y install mariadb-server

Install JavaSDK - yum -y install java-sdk

python-javapackages-3.4.1-11.el7.noarch

java-1.8.0-openjdk-devel-1.8.0.222.b03-1.el7.x86\_64

tzdata-java-2019b-1.el7.noarch

java-1.8.0-openjdk-headless-1.8.0.222.b03-1.el7.x86\_64

java-1.8.0-openjdk-1.8.0.222.b03-1.el7.x86\_64

javapackages-tools-3.4.1-11.el7.noarch

## **INSTALL DBEAVER/BeyondCompare**

Install DBeaver - https://dbeaver.io/download/ - download "Linux RPM package 64 bit (installer)

<https://dbeaver.io/files/dbeaver-ce-latest-stable.x86_64.rpm>

Install BeyondCompare - yum -y localinstall /mnt/DVD/bcompare-\*.x86\_64.rpm

<http://www.scootersoftware.com/download.php>

**AS root**

**# cd /var/tmp/RPMs**

**# rpm -ivh /var/tmp/RPMs/dbeaver-ce-6.2.5-stable.x86\_64.rpm bcompare-4.3.2.24472.x86\_64.rpm -y --nogpgcheck**

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

# yum clean all

# yum repolist

**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

**NODEJS/NPM – Local install in users $HOME**

As directed by Gordon – use previous package.json from $HOME directory (in this insytance we are using NickG’s file

# su - $USER

# cp /var/tmp/RPMs/NickG.package.json ./package.json

# npm install

**#TinyMCE - NPM module**

# npm install acorn-dynamic-import@4.0.0

# npm install [acorn@^6.0.0](mailto:acorn@%5e6.0.0)

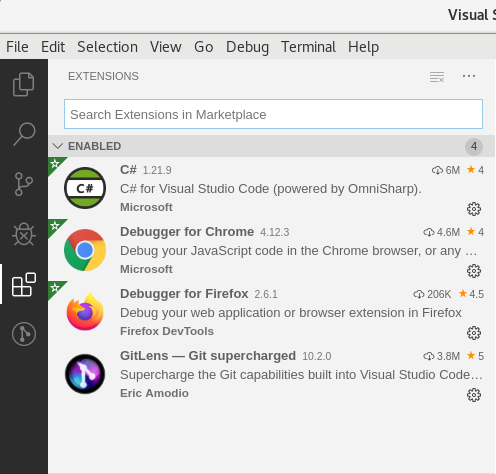
# npm install tinymce

#GitLens - Visual Studio addon

Ctrl+Shift K - search Gitlens

# Visual Code Chrome & Firefox Extensions

Visual Studio Add Ons



**ADD NEW USER TO GITECA ON GIT01 server**

As new user on new server:

# ssh-keygen (enter,enter, accept defaults)

# cd .ssh

# cat id\_rsa.pub

**example**

[IstvanK@lnlxik01 ~]$ cd .ssh

[IstvanK@lnlxik01 .ssh]$ cat id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCt5MtLjfuZtDpR5fuUdLTMq2CVUQ+1NTDBAp0YRN1PcBsAT4XbT772pPRsg7UxVq6RzxgWyKSOQT/t75Cv6uvBRe6NXsL1lDBLJnYAApYyI1HYidnonUbFyrUwuMENYuRdz0RSKkAvfv8U505o0QuY3m7GRxNIg9i39lIBYg7GquyglWYtyqWsC9eQQrZUE5XogccNcyJDUYsqQIFnkqSACoE5zhHd06DijeX47vMKQFF9Xfvv5Y8DLtZNiPESka+ySBwEzwrF4IoWkZUdlcLkMGeWBaoacxb+Wt7rswOS3vFduSxdf9DtFgLVfRtwsm7hQQUGdSC3ovXeoawYFXzN IstvanK@lnlxik01

Select/copy all the text start @ **ssh-rsa all the way to lnlxik01**

**On git01 server**

As root

# cd /etc/ssh/gitusers-keys

# vi giteca # open the file with the vi command line editor

Either use cursor keys to move to end line, or Esc, Shift+g (last line of file) You then need to open (so letter o) which will append a line to end of file in write mode

Then paste the text in

To save the file do:

Esc, Esc, :wq and hit enter so Esc, Esc, : and type wq