**Manual local RPM Build environment setup**

RPM build requires an environment where build files are packaged.

sudo yum install -y rpmdevtools rpmlint

Once the packages are installed , a rpm build environment can be created with

rpmdev-setuptree

This will create a rpmbuild directory inside user’s home directory with necessary rpm build directories

[caso@jen-docker-slave1 ~]$ ls rpmbuild/

BUILD  BUILDROOT  RPMS  SOURCES  SPECS  SRPMS

Copy the metro code in to SOURCES directory. I have added 2 extra files inside the nuage-metro directory to install pip packages, install-metro-deps.sh and pip-requirements.txt

Note that path SOURCES/metro-2.1.2/opt/nuage-metro-2.1.2 reflects the install path /opt/nuage-metro-2.1.2 during the rpm install.

[caso@jen-docker-slave1 opt]$ pwd

/home/caso/rpmbuild/SOURCES/metro-2.1.2/opt

[caso@jen-docker-slave1 opt]$ ls

nuage-metro-2.1.2

[caso@jen-docker-slave1 opt]$

Next step is to create a tar ball from the nuage-metro directory as this will be copied over to the /opt/nuage-metro-xxx directory during RPM install.

tar -zcvf metro-2.1.2.tar.gz metro-2.1.2/

[caso@jen-docker-slave1 ~]$ dir rpmbuild/SOURCES/

metro-2.1.2  metro-2.1.2.tar.gz

[caso@jen-docker-slave1 ~]$

RPM spec file specifies lot of details about the RPM and its dependencies. Create a spec file with

rpmdev-newspec <spec file name>

Update the rpm spec file inside SPECS directory with metro details,

Example metro.spec

[root@cas-cs2-009 RPM\_Build\_Dev]# cat rpms/metro.spec

Name: metro

Version: v0.0.0

Release: 1%{?dist}

Summary: Nuage metro packages

License: GPL

URL: https://github.com/nuagenetworks/nuage-metro

Source0: metro-%{version}.tar.gz

BuildArch: noarch

BuildRoot: %{\_tmppath}/%{name}-buildroot

Requires: git gcc sshpass openssl-devel

Requires: python2-pip python-ipaddr python-netaddr python-netifaces python-devel python-devel

Requires: PyYAML python-jinja2 python-paramiko pycrypto python-setuptools

%description

Metro playbooks and dependencies

%prep

%setup -q

%install

mkdir -p $RPM\_BUILD\_ROOT

cp -R \* $RPM\_BUILD\_ROOT

exit 0

%clean

rm -rf $RPM\_BUILD\_ROOT

%files

%defattr(-,root,root,-)

/opt/nuage-metro-%{version}

%pre

%post

# gets executed after installatin of the files

if [ -e /opt/nuage-metro-%{version} ]

then

pip install -r /opt/nuage-metro-%{version}/rpms/pip-requirements

fi

echo | sh /opt/nuage-metro-%{version}/rpms/VMware-ovftool-4.2.0-4586971-lin.x86\_64.bundle.txt --eulas-agreed --regular

[root@cas-cs2-009 RPM\_Build\_Dev]#

Above specifies the metro code version, which is set to 0.0.0 for dev branch. Also this specifies the RPM dependencies for metro except packages available only via pip such as vspk/netmiko. Ansible is installed through pip as this allows to specify the exact version. With yum install, latest available ansible will be picked instead of 2.2.1.0 which is not present in the latest repos.

After the install, rpm will place the nuage code in the /opt/nuage-metro-version folder as specified in the spec file. It will also run the commands under the %post section on the target server installing pip dependencies and vmware ovf tool.

Once SOURCE and SPECS are in good shape as above rpmbuild can be started. This will create the rpm package in RPMS folder.

[caso@jen-docker-slave1 rpmbuild]$ rpmbuild -v -bb SPECS/metro.spec

Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.mJoFcX

+ umask 022

+ cd /home/caso/rpmbuild/BUILD

+ cd /home/caso/rpmbuild/BUILD

+ rm -rf metro-2.1.2

+ /usr/bin/gzip -dc /home/caso/rpmbuild/SOURCES/metro-2.1.2.tar.gz

+ /usr/bin/tar -xf -

+ STATUS=0

+ '[' 0 -ne 0 ']'

+ cd metro-2.1.2

+ /usr/bin/chmod -Rf a+rX,u+w,g-w,o-w .

+ exit 0

Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.pPeL0n

+ umask 022

+ cd /home/caso/rpmbuild/BUILD

+ '[' /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64 '!=' / ']'

+ rm -rf /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

++ dirname /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

+ mkdir -p /home/caso/rpmbuild/BUILDROOT

+ mkdir /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

+ cd metro-2.1.2

+ mkdir -p /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

+ cp -R opt /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

+ '[' noarch = noarch ']'

+ case "${QA\_CHECK\_RPATHS:-}" in

+ /usr/lib/rpm/check-buildroot

+ /usr/lib/rpm/redhat/brp-compress

+ /usr/lib/rpm/redhat/brp-strip /usr/bin/strip

+ /usr/lib/rpm/redhat/brp-strip-comment-note /usr/bin/strip /usr/bin/objdump

+ /usr/lib/rpm/redhat/brp-strip-static-archive /usr/bin/strip

+ /usr/lib/rpm/brp-python-bytecompile /usr/bin/python 1

+ /usr/lib/rpm/redhat/brp-python-hardlink

+ /usr/lib/rpm/redhat/brp-java-repack-jars

Processing files: metro-2.1.2-1.el7.centos.noarch

Provides: metro = 2.1.2-1.el7.centos

Requires(interp): /bin/sh

Requires(rpmlib): rpmlib(CompressedFileNames) <= 3.0.4-1 rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PartialHardlinkSets) <= 4.0.4-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1

Requires(post): /bin/sh

Requires: /bin/bash /bin/sh /usr/bin/env /usr/bin/python

Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

Wrote: /home/caso/rpmbuild/RPMS/noarch/metro-2.1.2-1.el7.centos.noarch.rpm

Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.myDfbQ

+ umask 022

+ cd /home/caso/rpmbuild/BUILD

+ cd metro-2.1.2

+ rm -rf /home/caso/rpmbuild/BUILDROOT/metro-2.1.2-1.el7.centos.x86\_64

+ exit 0

This creates the rpm

[caso@jen-docker-slave1 ~]$ ls rpmbuild/RPMS/noarch/

metro-2.1.2-1.el7.centos.noarch.rpm

[caso@jen-docker-slave1 ~]

**Install RPM on a client machine**

Client machine should already have the EPEL repos installed. If not

[root@rpm-test3 ~]# yum install epel-release

Then copy the rpm to the ansible deployment machine and run yum install for the local rpm file.

[root@jen-docker-slave2 ~]# yum install metro-2.1.2-1.el7.centos.noarch.rpm

Loaded plugins: fastestmirror

Examining metro-2.1.2-1.el7.centos.noarch.rpm: metro-2.1.2-1.el7.centos.noarch

Marking metro-2.1.2-1.el7.centos.noarch.rpm to be installed

Resolving Dependencies

--> Running transaction check

---> Package metro.noarch 0:2.1.2-1.el7.centos will be installed

--> Finished Dependency Resolution

Dependencies Resolved<snip >

Extracting VMware Installer...done.

The product is ready to be installed. Press Enter to begin

installation or Ctrl-C to cancel.

Installing VMware OVF Tool component for Linux 4.2.0

Configuring...

[######################################################################] 100%

Installation was successful.

Verifying : metro-2.1.2-1.el7.centos.noarch 1/1

Installed:

metro.noarch 0:2.1.2-1.el7.centos

Complete!

[root@jen-docker-slave1 nuage-metro-2.1.2]#

After the install, rpm will place the nuage code in the /opt/nuage-metro-version folder. Also it will install all the pip dependencies and vmware ovftool.

Now the users can go to /opt/nuage-metro-version/ , update build\_vars.yml and run the playbooks.

Updated metro code can be downloaded via git clone now or through remove/reinstall of the RPM package.

**Uninstall RPM on a client machine**

To remove metro packages and all dependencies installed,

pip uninstall -r /opt/nuage-metro-v0.0.0/rpms/pip-requirements

yum remove metro

yum autoremove

/opt/nuage-metro-v0.0.0/rpms/VMware-ovftool-4.2.0-4586971-lin.x86\_64.bundle.txt --uninstall-component vmware-ovftool

**Automated RPM builds with Jenkins**

There are two Jenkins jobs to build metro RPMs. Related shell script and rpm spec files are under the rpms folder inside metro.

Following can launch rpm build on dev branch

./rpms/rpm\_build.sh dev

Dev branch will always have a single rpm build and versioned v0.0.0

For master branch,

./rpms/rpm\_build.sh master

Master branch will get the version number from the release tag.

Following is used to get the latest tag for the master branch for the purpose of rpm naming

git describe --tags $(git rev-list --tags --max-count=1)

Caveats :

1. Jenkins node that runs build jobs are supposed to be setup with keyless ssh access to the ftp server (135.227.181.233) . This allows Jenkins shell scripts to copy the built RPM files to the ftp server as well as bring Vmware OVF tool during the build time in to the build server. Currently Jenkins jobs are scheduled on to BM server cs2-009.
2. Jenkins jobs for creating automated builds should be run as caso or any other user having a home directory and logging shell. This is due to the fact rpm build process needs to create directory and run shell commands.
3. Sometimes rpmbuilds folder on Jenkins server may needs to be reset to user/group of caso with chown –R caso.caso ~/rpmbuilds
4. Sometimes home/rpm folder on ftp server may needs to be set chmod –R 755 /home/rpm