DOCKER INSTALLATION IN RHEL:

1. Update the yum package index

```
$ sudo yum makecache fast
```

2. Install the latest version of Docker EE, or go to the next step to install a specific version.

```
$ sudo yum -y install docker-ee
```

3. On production systems, you should install a specific version of Docker instead of always using the latest. List the available versions. This example uses the sort -r command to sort the results by version number, highest to lowest, and is truncated.

```
$ yum list docker-ee.x86_64  --showduplicates |sort -r
  docker-ee.x86_64    17.03.0.el7          docker-ee-stable
```

The second column is the version string. The third column is the repository name

```
$ sudo yum -y install docker-ee-<VERSION_STRING>
```

4. Start Docker.

```
$ sudo systemctl start docker
```

5. Verify that Docker EE is installed correctly by running the hello-world image.

```
$ sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

Getting Started with Alpine:

Alpine is a lightweight linux distribution based on musl libc and busybox. There is a docker image based on Alpine which is an easy way of getting started with Alpine

Alpine Docker Image:

Based on Alpine kernel, this is a lightweight image of 5MB

1. Pull the alpine image

```
$ docker pull alpine
```

[In my machine I pulled Alpine 3.5.2 version]

2. Check IP Address of the container

```
$ docker run alpine ifconfig
```

3. Launching a bash shell

```
$ docker run -i -t alpine /bin/bash
```

This will give an error, as bash is not supported in alpine

```
exec: "/bin/bash": stat /bin/bash: no such file or directory
docker: Error response from daemon: Container command not found or does not exist..
```

4. Getting inside the container

```
$ docker run -it alpine /bin/sh

/ #
```

Detaching from the container without stopping Ctrl-P Ctrl-Q

5. Check the docker container is still running

```
$ docker ps -a
```

```
CONTAINER ID    IMAGE   COMMAND      CREATED             STATUS              POST    NAMES
8647ce2b84a5    alpine  "/bin/sh"    About a minute ago  Up About a minute           elegant_rosalind
```

Rename the container(optional):

```
$ docker rename <old_container_name> <new_container_name>
```

ex. $ docker rename elegant_rosalind myalpine

Start Alpine Container After Exiting From Container:

1. start docker engine.

```
$ sudo systemctl start docker
```

2. start the docker

```
$ docker start <container_name>
```

ex. $ docker start myalpine

3. Attach the container

```
$ docker attach <container_name>
```

ex. $ docker attach myalpine

Adding git command to Alpine Linux:

In Alpine Linux git command is not present by default so to install git command run the following commands

```
# apk update

# apk upgrade

# apk add --no-cache bash git openssh
```

Cloning Eclipse OMR git repository to your Alpine Linux container:

```
# git clone https://github.com/eclipse/omr.git
```

What Is Eclipse OMR?

____The Eclipse OMR project is a set of open source C and C++ components that can be used to build robust language runtimes that support many different hardware and operating system platforms.

current components are:

gc,compiler,jitbuilder,port,thread,util,omrsigcompat,omrtrace,tool,vm,example,fvtest

Adding make command to Alpine Linux:

In Alpine Linux make command is not present by default so to install make command run the following command

```
# apk add --no-cache bash make
```

Basic configuration and compile:

To build standalone Eclipse OMR, run the following commands from the top of the source tree. The top of the Eclipse OMR source tree is the directory that contains run_configure.mk.

```
#make -f run_configure.mk SPEC=linux_x86-64 OMRGLUE=./example/glue
```

```
#make
```

Adding gcc command to Alpine Linux:

In Alpine Linux gcc command is not present by default so to install gcc command run the following command

```
# apk add --no-cache gcc musl-dev
```

Adding musl-gcc command to Alpine Linux:

In Alpine Linux musl-gcc command is not present by default so to install musl-gcc command first clone the musl using below command

```
#git clone git://git.musl-libc.org/musl
```

Change the present working directory to musl

```
#cd musl
```

Run the below command

```
#./configure && make install
```

Adding c++ command to Alpine Linux:

In Alpine Linux c++ command is not present by default so to install c++ command run the below commands.

```
#apk add --update g++

#rm /var/cache/apk/*
```

Adding perl command to Alpine Linux:

In Alpine Linux perl command is not present by default so to install perl command run the following commands

```
#apk update

#apk upgrade

#apk add bash wget curl perl make g++ libev-dev patch git openssl-dev openssl
```

Issue:

Fatal error: execinfo.h: No such file or directory #include <execinfo.h>

Solution:

In Alpine Linux execinfo.h is not present by default so to add this header file run the following command

```
#apk add libexecinfo-dev
```

execinfo.h is a GNU specific header, and doesn't exist under musl

Change ifdef to check for __GLIBC__ instead of __linux__ to prevent errors when building under other libc's

[Note: Still there are some more issues to fix]

Issue:

error: __sighandler_t is not defined in this scope

Solution:

```
# sed -i "s/struct sigaction {/#ifndef __sighandler_t \ntypedef void
 (*__sighandler_t)(int);\n#endif\nstruct sigaction\n{/g" /usr/include/signal.h
#sed -i "s/union {void (*sa_handler)(int)/__sighandler_t sa_handler/g"
 /usr/include/signal.h
```

Issue:

error: 'sigmask' was not declared in this scope

error: 'SV_ONSTACK' was not declared in this scope

error: 'SV_INTERRUPT' was not declared in this scope

error: 'SV_RESETHAND' was not declared in this scope

error: 'sigmask' was not declared in this scope

error: invalid use of incomplete type 'const struct sigvec'

Solution:

```
#if defined(LINUX) && !defined(ALPINE)
```

Issue:

fatal error: numa.h: No such file or directory #include <numa.h>

Solution:

To disable the numa run the following commands from the top of the source tree. The top of the Eclipse OMR source tree is the directory that contains run_configure.mk.

```
#make -f run_configure.mk SPEC=linux_x86-64 OMRGLUE=./example/glue
  'EXTRA_CONFIGURE_ARGS=--disable-OMR_PORT_NUMA_SUPPORT' clean all
```

Issue:

error: unknown type name 'sigval_t' sigval_t val;

error: request for member 'sival_ptr' in something not a structure or union

Solution:

Replace sigval_t with union sigval in omrintrospect.c file

Issue:

error: redefinition of 'struct prctl_mm_map' struct prctl_mm_map

Solution:

Comment the header file #include<linux/prctl.h> in omrosdump_helpers.c file.

Issue:

error: 'HZ' undeclared (first use in this function) #define USER_HZ HZ

Solution:

Add the below code in the beginning of the file where you got the error

```
#define PROC_PARTITIONS PROC_FS_ROOT "partitions"

#define PROC_DISKSTATS  PROC_FS_ROOT "diskstats"

#ifndef HZ

#define HZ 100

#endif
```

Issue:

error: implicit declaration of function 'pthread_attr_getstackaddr'

Solution:

Add the below code in the file omrthreadinspect.c

#if _GLIBCXX_USE_C99

#if _GLIBCXX_USE_C99 || defined __UCLIBC__

Issue:

error: missing binary operator before token "(" #if __GLIBC_PREREQ(2,4)

Solution:

In alpine linux it will not support the version of the macro so remove the version number in the macro #if __GLIBC_PREREQ

Issue:

error: implicit declaration of function 'gettid'

error: unknown type name 'gettid'

Solution:

gettid( ) is not defined in alpine linux so undefine the gettid()

#if !__GLIBC_PREREQ && !defined(ALPINE)