openEuler riscv -OSC 安装和功能测试

准备工作

在安装好 openEuler 环境后，启动并登录，按照测试说明为 dnf 添加如下软件库：

http://119.3.219.20:82/openEuler:/Mainline:/RISC-V/standard_riscv64/

添加可使用 dnf config-manager --add-repo repository_url 指令，在添加完成后执行 dnf repolist all 可以得到以下的输出：

之后按照测试说明安装相关软件包，安装之前使用 date 指令查看时间，如果时间不正确则使用 date -s 指令设置正确时间。在安装过程中，如果安装过程中出现问题 noarch 认证问题，则执行 **dnf config-manager --set-enabled oe-noarch** 之后重试，最终会显示如下的结果：
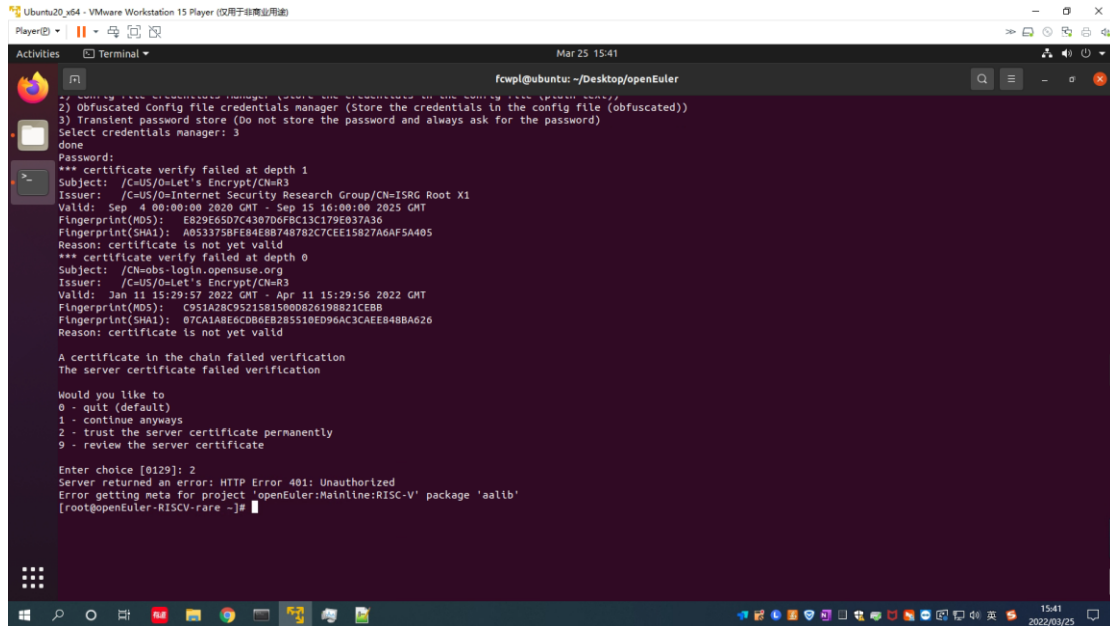

之后按照说明中的步骤，执行 **osc co openEuler:Mainline:RISC-V/aalib**，输入用户名和密码，在选择密码是否存储等问题之后会得到如下图中的输出：

      可以看到提示了未认证，最后程序退出了，这时需要到 **~/.config/osc** 下，将 **oscrc** 文件中的 **apiurl** 改为 **https://build.openeuler.org/**，去掉 **no_verify = 1** 的注释，并将 **[https://api.opensuse.org]** 改为 **[https://build.openeuler.org]**，注释中的可以不管，最终的文件内容如下：

[root@openEuler-RISCV-rare osc]# cat oscrc

[general]

URL to access API server, e.g. https://api.opensuse.org

you also need a section [https://api.opensuse.org] with the credentials

apiurl = https://build.openeuler.org

Downloaded packages are cached here. Must be writable by you.

#packagecachedir = /var/tmp/osbuild-packagecache

Wrapper to call build as root (sudo, su -, ...)

#su-wrapper = sudo

rootdir to setup the chroot environment

can contain %(repo)s, %(arch)s, %(project)s, %(package)s and %(apihost)s (apihost is the hostname

extracted from currently used apiurl) for replacement, e.g.

/srv/oscbuild/%(repo)s-%(arch)s or

/srv/oscbuild/%(repo)s-%(arch)s-%(project)s-%(package)s

#build-root = /var/tmp/build-root/%(repo)s-%(arch)s

compile with N jobs (default: "getconf _NPROCESSORS_ONLN")

#build-jobs = N

build-type to use - values can be (depending on the capabilities of the 'build' script)

empty   -   chroot build

kvm   -   kvm VM build   (needs build-device, build-swap, build-memory)

xen   -   xen VM build   (needs build-device, build-swap, build-memory)

experimental:

qemu -   qemu VM build

```
#        lxc    -    lxc build
#build-type =
# build-device is the disk-image file to use as root for VM builds
# e.g. /var/tmp/FILE.root
#build-device = /var/tmp/FILE.root
# build-swap is the disk-image to use as swap for VM builds
# e.g. /var/tmp/FILE.swap
#build-swap = /var/tmp/FILE.swap
# build-kernel is the boot kernel used for VM builds
#build-kernel = /boot/vmlinuz
# build-initrd is the boot initrd used for VM builds
#build-initrd = /boot/initrd
# build-memory is the amount of memory used in the VM
# value in MB - e.g. 512
#build-memory = 512
# build-vmdisk-rootsize is the size of the disk-image used as root in a VM build
# values in MB - e.g. 4096
#build-vmdisk-rootsize = 4096
# build-vmdisk-swapsize is the size of the disk-image used as swap in a VM build
# values in MB - e.g. 1024
#build-vmdisk-swapsize = 1024
# build-vmdisk-filesystem is the file system type of the disk-image used in a VM build
# values are ext3(default) ext4 xfs reiserfs btrfs
#build-vmdisk-filesystem = ext4
# Numeric uid:gid to assign to the "abuild" user in the build-root
# or "caller" to use the current users uid:gid
# This is convenient when sharing the buildroot with ordinary userids
# on the host.
# This should not be 0
# build-uid =
# strip leading build time information from the build log
# buildlog_strip_time = 1
# Enable ccache in build roots.
# ccache = 1
# extra packages to install when building packages locally (osc build)
# this corresponds to osc build's -x option and can be overridden with that
# -x '' can also be given on the command line to override this setting, or
# you can have an empty setting here. This global setting may leads to
# dependency problems when the base distro is not providing the package.
# => using server side definition via cli_debug_packages substitute rule is
#      recommended therefore.
#extra-pkgs =
# build platform is used if the platform argument is omitted to osc build
#build_repository = openSUSE_Factory
```

```
# default project for getpac or bco
#getpac_default_project = openSUSE:Factory
# alternate filesystem layout: have multiple subdirs, where colons were.
#checkout_no_colon = 0
# change filesystem layout: avoid checkout within a project or package dir.
#checkout_rooted = 0
# local files to ignore with status, addremove, ....
#exclude_glob = .osc CVS .svn .* _linkerror *~ #*# *.orig *.bak *.changes.vctmp.*
# limit the age of requests shown with 'osc req list'.
# this is a default only, can be overridden by 'osc req list -D NNN'
# Use 0 for unlimted.
#request_list_days = 0
# show info useful for debugging
#debug = 1
# show HTTP traffic useful for debugging
#http_debug = 1
# number of retries on HTTP transfer
#http_retries = 3
# Skip signature verification of packages used for build.
no_verify = 1
# jump into the debugger in case of errors
#post_mortem = 1
# print call traces in case of errors
#traceback = 1
# check for unversioned/removed files before commit
#check_filelist = 1
# check for pending requests after executing an action (e.g. checkout, update, commit)
#check_for_request_on_action = 1
# what to do with the source package if the submitrequest has been accepted. If
# nothing is specified the API default is used
#submitrequest_on_accept_action = cleanup|update|noupdate
# template for an accepted submitrequest
#submitrequest_accepted_template = Hi %(who)s,\n
# thanks for working on:\t%(tgt_project)s/%(tgt_package)s.
# SR %(reqid)s has been accepted.\n\nYour maintainers
# template for a declined submitrequest
#submitrequest_declined_template = Hi %(who)s,\n
# sorry your SR %(reqid)s (request type: %(type)s) for
# %(tgt_project)s/%(tgt_package)s has been declined because...
#review requests interactively (default: off)
#request_show_review = 1
# if a review is accepted in interactive mode and a group
# was specified the review will be accepted for this group (default: off)
#review_inherit_group = 1
```

```
[https://build.openeuler.org]
# set aliases for this apiurl
# aliases = foo, bar
# real name used in .changes, unless the one from osc meta prj <user> will be used
# realname =
# email used in .changes, unless the one from osc meta prj <user> will be used
# email =
# additional headers to pass to a request, e.g. for special authentication
#http_headers = Host: foofoobar,
#               User: mumblegack
# Plain text password
#pass =
user=HuangYuekai
credentials_mgr_class=osc.credentials.TransientCredentialsManager
[root@openEuler-RISCV-rare osc]#
```
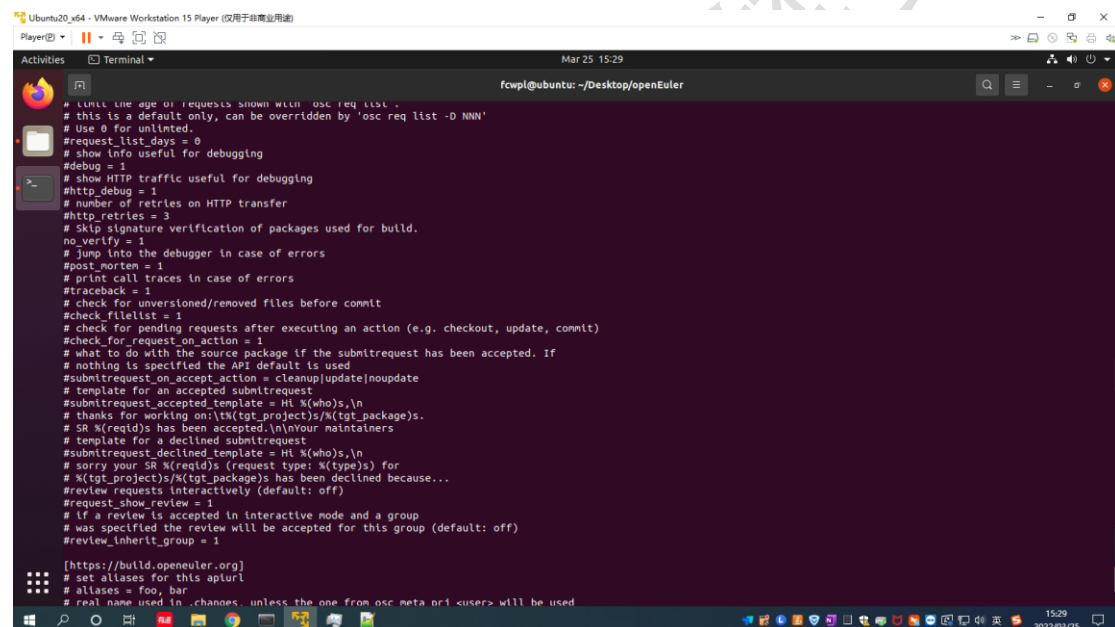

至此，准备阶段的工作就完成了。

测试阶段

1.再次运行 osc co openEuler:Mainline:RISC-V/aalib

调整完成后，再次运行 osc co openEuler:Mainline:RISC-V/aalib 指令，得到如下的输出，如果之前没有选择 2 则这里会提示再选择，选择 2 最终执行成功
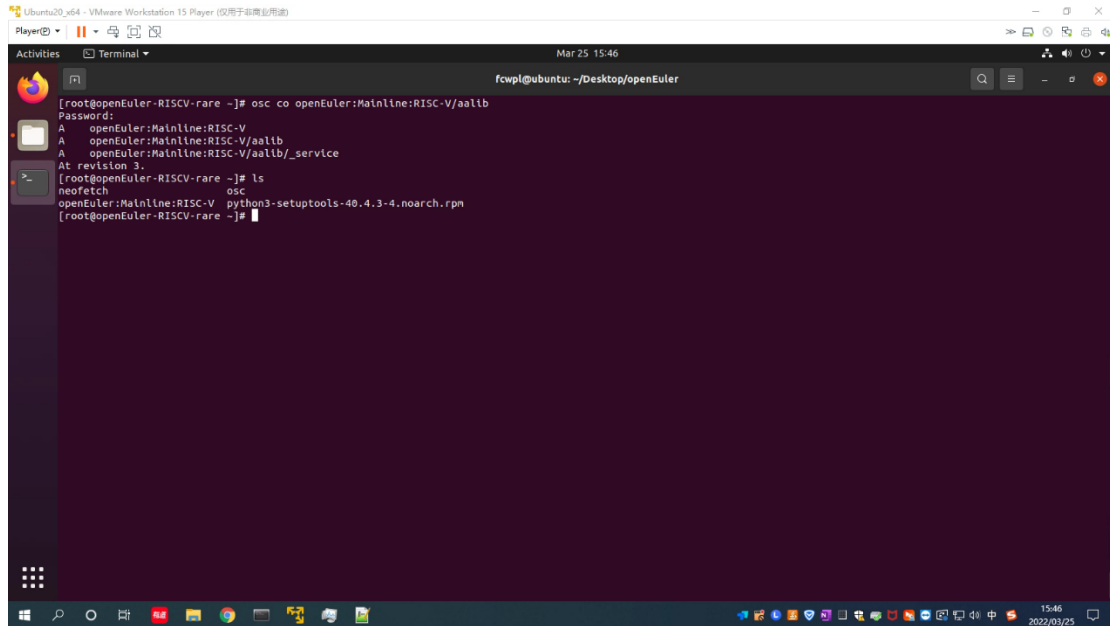
如果之前没有选择 2，则这里的输出如下：

[root@openEuler-RISCV-rare osc]# osc co openEuler:Mainline:RISC-V/aalib

Password:

*** certificate verify failed at depth 1

Subject: /C=US/O=Let's Encrypt/CN=R3

Issuer: /C=US/O=Internet Security Research Group/CN=ISRG Root X1

Valid: Sep   4 00:00:00 2020 GMT - Sep 15 16:00:00 2025 GMT

Fingerprint(MD5):   E829E65D7C4307D6FBC13C179E037A36

Fingerprint(SHA1):   A053375BFE84E8B748782C7CEE15827A6AF5A405

Reason: certificate is not yet valid

*** certificate verify failed at depth 0

Subject: /CN=build.openeuler.org

Issuer: /C=US/O=Let's Encrypt/CN=R3

Valid: Feb 25 02:41:36 2022 GMT - May 26 02:41:35 2022 GMT

Fingerprint(MD5):   F8EEDA0BB1AA2705BF4F5ECC129AD48D

Fingerprint(SHA1):   A684E5C35A32FE93B6D93DF65DEDC976ADD2954E

Reason: certificate is not yet valid

A certificate in the chain failed verification

The server certificate failed verification

Would you like to

0 - quit (default)
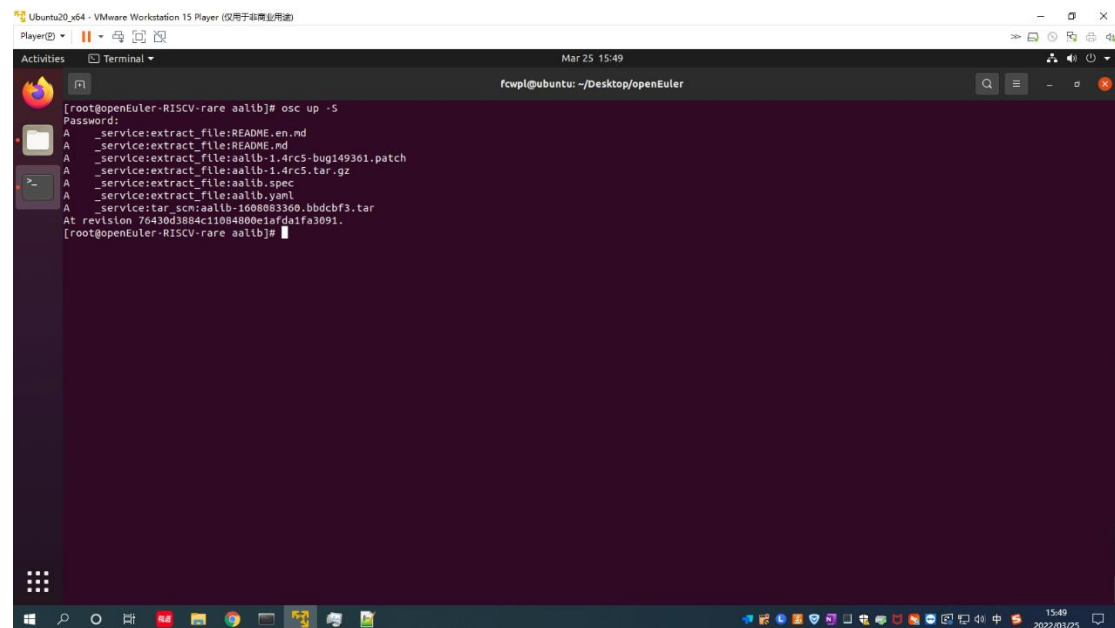
1 - continue anyways

2 - trust the server certificate permanently

9 - review the server certificate

Enter choice [0129]: 2

A   openEuler:Mainline:RISC-V

A   openEuler:Mainline:RISC-V/aalib

A   openEuler:Mainline:RISC-V/aalib/_service

At revision 3.

[root@openEuler-RISCV-rare osc]#

2.进入 openEuler:Mainline:RISC-V/aalib 文件夹，运行 osc up -S 命令

按照说明执行代码，可以得到如下的输出：

表明成功执行并下载相应软件包。

3.构建 aalib 包，记录最后几行构建成功的信息

按照测试说明，继续执行 **rm -f _service;for file in `ls | grep -v .osc`;do new_file=${file##*:};mv $file $new_file;done**，然后运行 **osc build**，得到如下的输出：

可以看到，输出结果提升 finished "build aalib.spec"，至此，**测试阶段全部完成**。