

Tutorial: Making code contributions to O-RAN open source project

The tutorial covers how to make code contributions to the O-RAN Software Community and O-RAN Spec Code repositories.

Setting Up Contribution Account

1. To make contributions to Linux Foundation Gerrit, a Linux Foundation ID is needed for identifying the contributor.
 - a. If without one, a Linux Foundation ID can be created simply by going to the URL: <https://identity.linuxfoundation.org>, selecting: "I need to create a Linux Foundation ID", and completing the registration form. Shortly after clicking on the "Create new account" button, a validation email from "The Linux Foundation <do-not-reply@linuxfoundation.org>" with subject "Validate your Linux Foundation ID email" should arrive in the email box used for LF ID registration. Inside of this email there is an embedded link and clicking on it will complete the validation process and lead to a page with a green bannered message: "You have successfully validated your e-mail address." Congratulations! You now have a Linux Foundation ID. Linux Foundation uses single sign-on so this LF ID is used for accessing the whole LF tool set.
2. Now let's set up the ssh key for the contribution account.
 - a. Decide a key pair to be used for making code contribution;
 - i. If there is none, or we want to generate a new pair, we can do so by running: **ssh-keygen -t rsa**. After the generation is completed, we would be able to find the private key at the location and file name specified during the interactive session. The public key is stored at the same location, with the same file name but an extension of **.pub**.
 - ii. For newly generate key, we would want to add it to the local key list by running **eval \$(ssh-agent)** and **ssh-add**. Then we can test the ssh key by using it to establish a ssh session with the O-RAN SC Gerrit (see example below).
 - b. Add the public key of the key pair to the Gerrit server.
 - i. Login to <https://gerrit.o-ran-sc.org> with the LF ID;
 - ii. Click on the user name at the top-right corner of the page, then select "**Settings**" from the pull-down menu;
 - iii. Find the section about "**SSH Keys**" on the Settings page, paste our public key into the "**New SSH Key**" box, then click on the "**ADD NEW SSH KEY**" button. We should now see this key appearing at the top of the SSH Keys section. We will also receive an email notification about a new SSH key just being added.
 - c. Below is an example session of generating a new key pair, add to local ssh server, and test it with the Gerrit server.

Sample key set up session

```
$ ssh-keygen -t rsa -C "your_email@youremail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/testaccount/.ssh/id_rsa): /tmp/testkey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /tmp/testkey.
Your public key has been saved in /tmp/testkey.pub.
The key fingerprint is:
SHA256:DaNfWTIWxeG6SSfSVT4FuRuzZI5vTYRrUEoCzFl2rRg your_email@youremail.com
The key's randomart image is:
+---[RSA 2048]-----+
|      o.++o+ooo.|
|      +.E+oo+. |
|      o +=++oo |
|      . *.B+ B..|
|      . S O .* B |
|      . + =. * . |
|      . o o o |
|      o . |
|      . |
+-----[SHA256]-----+
$ ls testkey*
testkey      testkey.pub
$ eval $(ssh-agent)
Agent pid 69281
$ ssh-add ./testkey
Identity added: ./testkey (your_email@youremail.com)
$ ssh -p 29418 testaccount@gerrit.o-ran-sc.org
The authenticity of host '[gerrit.o-ran-sc.org]:29418 ([34.215.66.175]:29418)' can't be
established.
ECDSA key fingerprint is SHA256:hd0AoKsvuL7gCETA8JLuLhhUBHcjwHysY6GvKSs+gfc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[gerrit.o-ran-sc.org]:29418,[34.215.66.175]:29418' (ECDSA) to the list
of known hosts.

****      Welcome to Gerrit Code Review      ****

Hi XXXX, you have successfully connected over SSH.

Unfortunately, interactive shells are disabled.
To clone a hosted Git repository, use:

git clone ssh://testaccount@gerrit.o-ran-sc.org:29418/REPOSITORY_NAME.git

Connection to gerrit.o-ran-sc.org closed by remote host.
Connection to gerrit.o-ran-sc.org closed.
```

Setting Up Local Development Environment

1. Install git (example below for Ubuntu dev machine)

```
sudo apt-get install git
```
2. Install git-review aka Gerrit client plugin (example below for Ubuntu dev machine)

```
sudo apt-get install git-review
```
3. Configuring git for code contribution with Linux Foundation ID and the email registered with the LF ID.
 - a.

```
git config --global user.email "example@example.com"
git config --global user.name "example"
```

Contributing Code

1. Clone the repo that your contribution is for: **git clone http://gerrit.o-ran-sc.org/r/<REPO_PATH>**
2. Change dir into the repo;
3. Sign for your code change (comment-thoralf_"review -s" does not do any signing. this step seems unnecessary // maybe it was supposed to be "git commit -s") : **git review -s**; This is where Gerrit plugin using ssh to verify with gerrit.o-ran-sc.org that your email is already registered with the server;
4. Follow the usual git code change work flow: make code change, git add, and git commit;
 - a. The message provided at git commit should be used as commit summary, --keep it short;
5. Update commit message with **git commit -s --amend**; This is where more detailed commit messages are added, and a signed-off by line inserted to commit message;
6. Submit code change with **git review**;

After the initial code change submit, if further changes (patches) are needed for the same submission, we can make the change and git add, use **git commit -s --amend** to commit and modify the commit message, then **git review** to submit for code review. Do not use just **git commit**, as it will create a new change ID and causing the new patch to be submitted as a new change instead of a patch atop of the existing code submission.