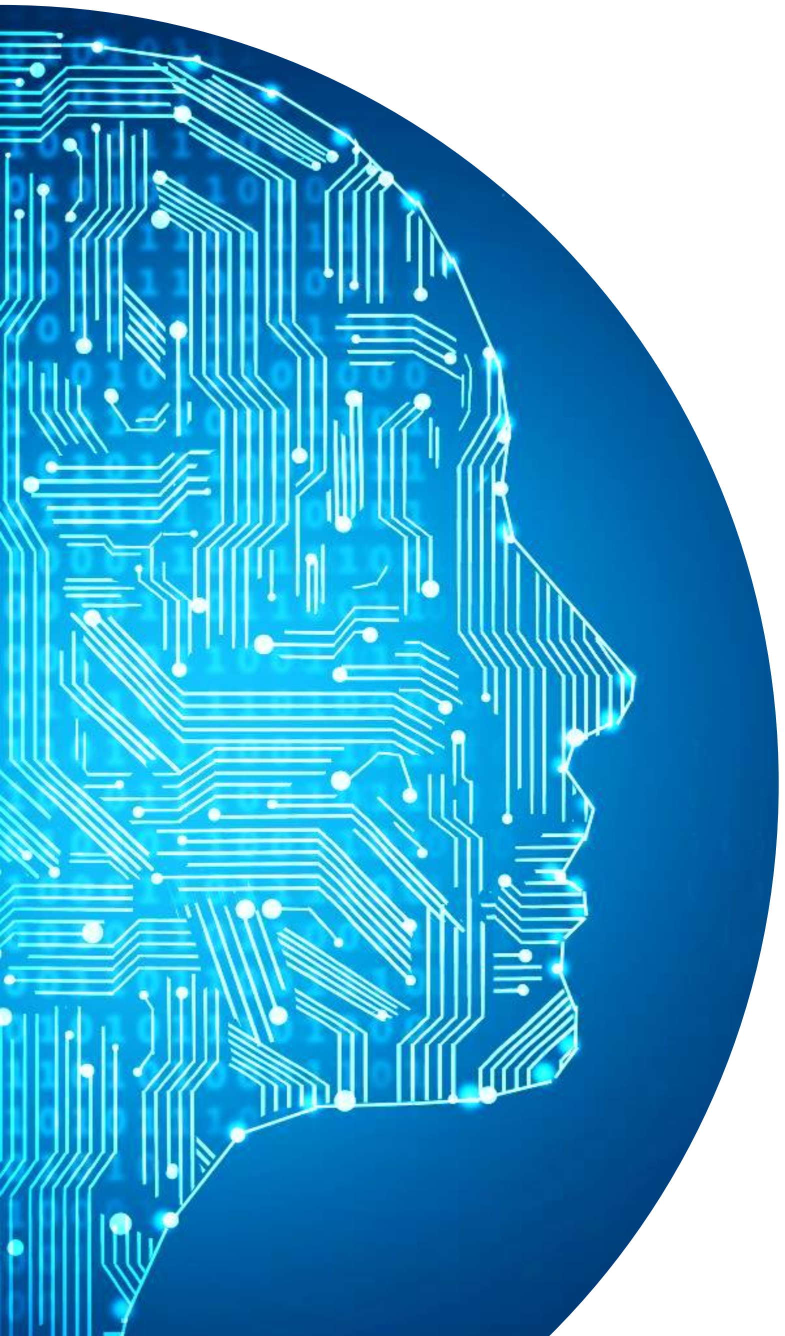
Open Source MANO by ETSI

How to contribute to OSM

Gulsum Atici (Canonical, MDL)
Gerardo García (Telefónica, OSM TSC chair)

OSM#15

12/06/2023



Agenda

- Get an EOL /Individual Contributor Account
- Prepare your environment
- Testing before committing
- Contribute your changes

Get an EOL / Individual Contributor Account

Get an EOL /Individual Contributor Account

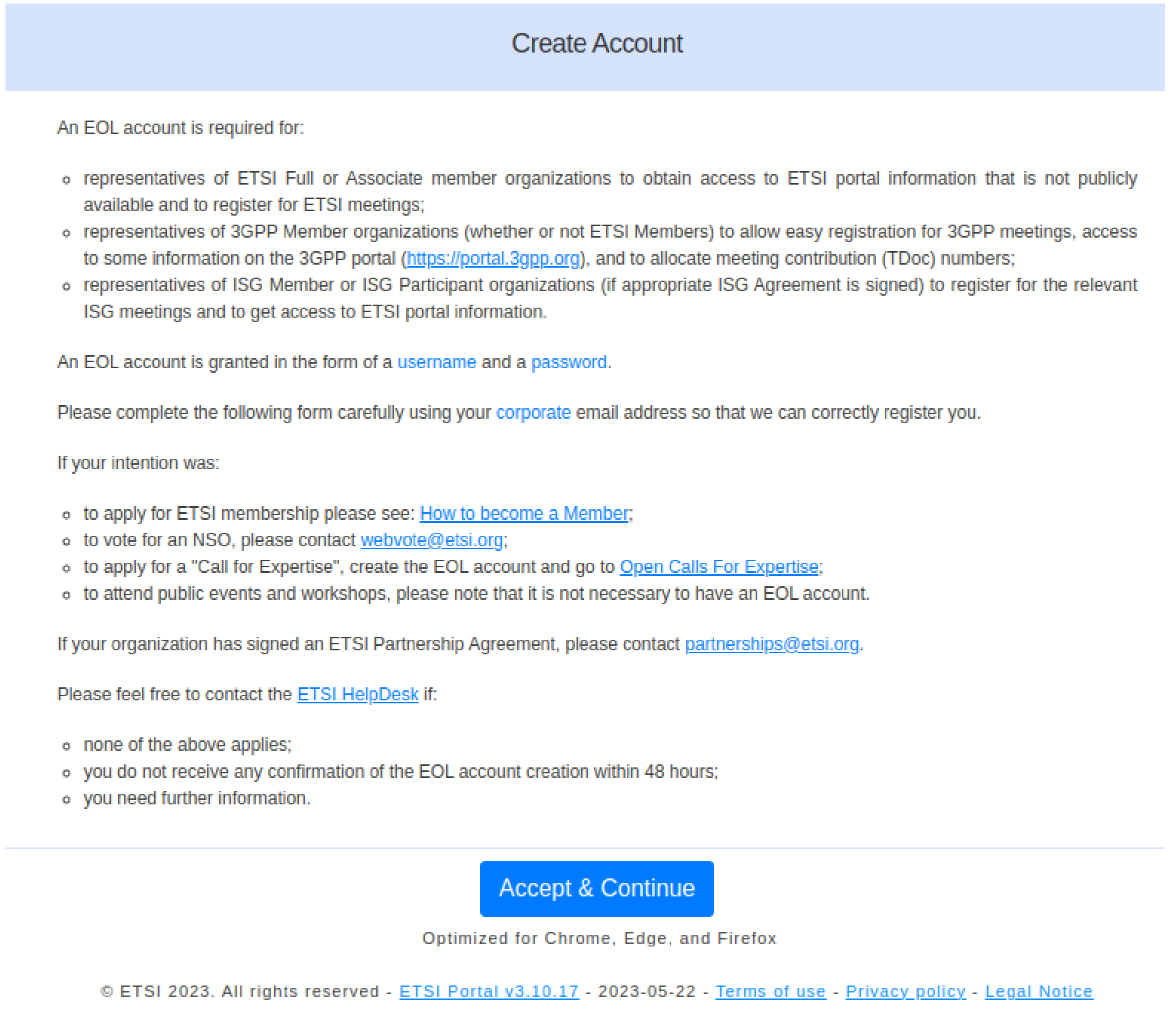
If you are contributing on behalf of your company, you should login with your **ETSI Online Account (EOL)**.

If your company is not yet an OSM Member or Participant, you can check here how to join OSM as an organization: https://osm.etsi.org/about/how-to-join

If your **company has already joined OSM** but you do not have an EOL account yet, **you can request an EOL account** at: https://portal.etsi.org/createaccount

If you are an individual contributor, you can create your OSM **individual contributor account** at: https://osm.etsi.org/register

If you need any help, contact us at: **OSMsupport@etsi.org**



Create Account

An EOL account is required for:

- representatives of ETSI Full or Associate member organizations to obtain access to ETSI portal information that is not publicly available and to register for ETSI meetings;
- representatives of 3GPP Member organizations (whether or not ETSI Members) to allow easy registration for 3GPP meetings, access to some information on the 3GPP portal (https://portal.3gpp.org), and to allocate meeting contribution (TDoc) numbers;
- representatives of ISG Member or ISG Participant organizations (if appropriate ISG Agreement is signed) to register for the relevant ISG meetings and to get access to ETSI portal information.

An EOL account is granted in the form of a username and a password.

Please complete the following form carefully using your corporate email address so that we can correctly register you.

If your intention was:

- to apply for ETSI membership please see: How to become a Member;
- to vote for an NSO, please contact webvote@etsi.org;
- to apply for a "Call for Expertise", create the EOL account and go to Open Calls For Expertise;
- to attend public events and workshops, please note that it is not necessary to have an EOL account.

If your organization has signed an ETSI Partnership Agreement, please contact partnerships@etsi.org.

Please feel free to contact the ETSI HelpDesk if:

- none of the above applies;
- you do not receive any confirmation of the EOL account creation within 48 hours;
- you need further information.

Optimized for Chrome, Edge, and Firefox

© ETSI 2023. All rights reserved - ETSI Portal v3.10.17 - 2023-05-22 - Terms of use - Privacy policy - Legal Notice

Open Source MANO

Prepare your environment

Configure git to use ssh key

- Create an SSH key (e.g. with ssh-keygen)
  - ~/.ssh/id_rsa
  - ~/.ssh/id_rsa.pub
- Add your public key to Gerrit
  - Go to https://osm.etsi.org/gerrit/#/settings/ssh-keys --> Add Public Key
- Configure the file .ssh/config. Add following content to the .ssh/config file.

```
Host osm.etsi.org
  Hostname osm.etsi.org
  User <your_etsi_user>
  IdentityFile ~/.ssh/id_rsa
  PubkeyAcceptedAlgorithms +ssh-rsa
  HostkeyAlgorithms +ssh-rsa
```

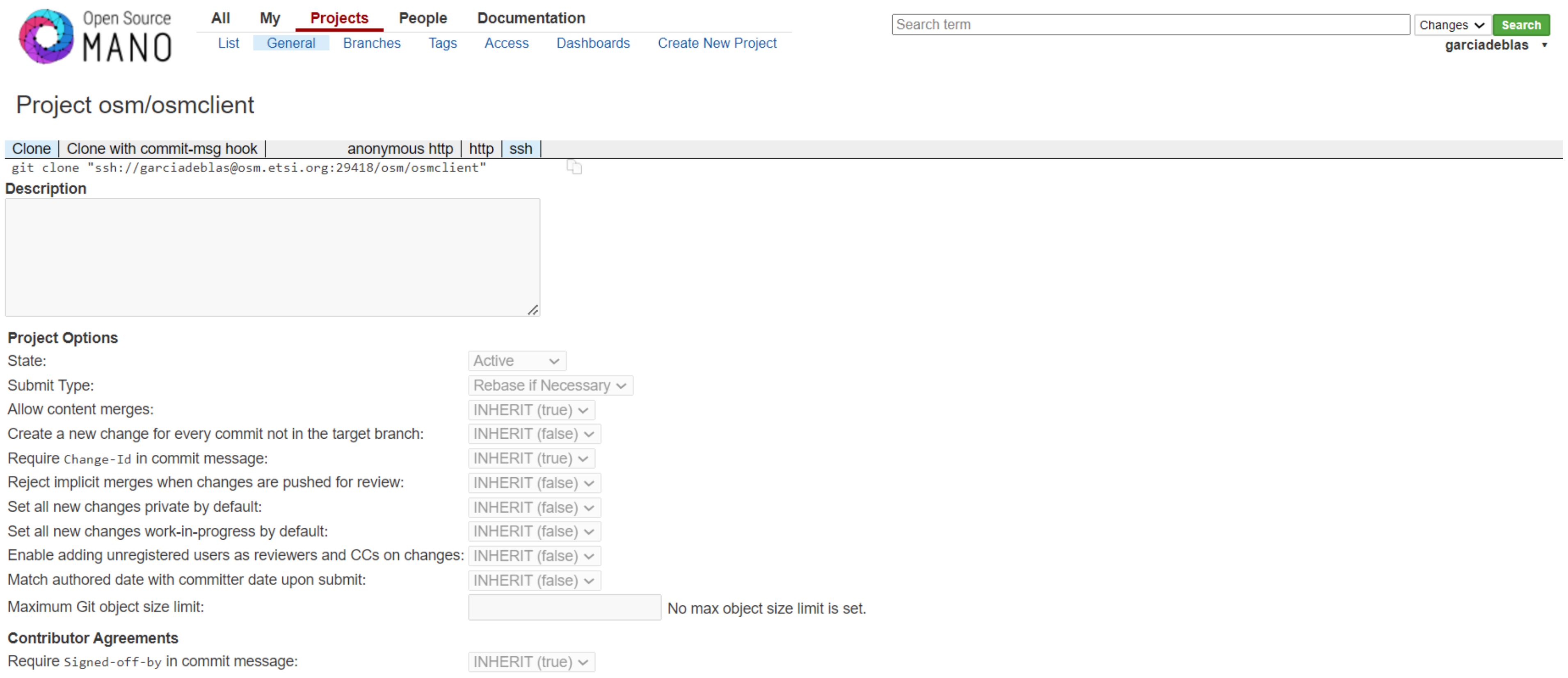
- Test your ssh connection
  - ssh -p 29418 osm.etsi.org

Configure git username and e-mail

- Configure your git user globally:
```
git config --global user.user <username>
```
- Configure your git name globally:
```
git config --global user.name <name>
```
- Configure your git email address:
```
git config --global user.email <email>
```
- Check your git configuration:
```
git config --list
```
- Sample Output:
```
git config --list
user.email=gulsum.atici@canonical.com
user.name=Gulsum Atici
user.user=aticig
```
- In case you are using git in the same computer for other open source projects, you can restrict your git variables to the local folder:
```
cd <LOCAL_FOLDER>
git config --local user.name <username>
git config --local user.email <email>
```

Clone the projects you are going to work on

- Go to Gerrit Projects (https://osm.etsi.org/gerrit/#/admin/projects/), select a Project, then clone the project with SSH and commit message hook


Open Source MANO
All My Projects People Documentation
List General Branches Tags Access Dashboards Create New Project
Search term
Changes
Search
garciadeblas

Project osm/osmclient

Clone | Clone with commit-msg hook | anonymous http | http | ssh |

```
git clone "ssh://garciadeblas@osm.etsi.org:29418/osm/osmclient"
```

Description

Project Options

State:	Active
Submit Type:	Rebase if Necessary
Allow content merges:	INHERIT (true)
Create a new change for every commit not in the target branch:	INHERIT (false)
Require `Change-Id` in commit message:	INHERIT (true)
Reject implicit merges when changes are pushed for review:	INHERIT (false)
Set all new changes private by default:	INHERIT (false)
Set all new changes work-in-progress by default:	INHERIT (false)
Enable adding unregistered users as reviewers and CCs on changes:	INHERIT (false)
Match authored date with committer date upon submit:	INHERIT (false)
Maximum Git object size limit:	No max object size limit is set.

Contributor Agreements

Require `Signed-off-by` in commit message:	INHERIT (true)

Clone the projects you are going to work on

The following script can help you to clone all OSM projects with SSH and the commit-msg hook

```
mkdir ~/OSM/
cd ~/OSM
BRANCH=master
# Update <EOL_USERNAME> accordingly
MY_EOL=<EOL_USERNAME>
for MDG in common devops IM LCM MON N2VC NBI NG-SA NG-UI osmclient PLA POL RO tests; do
  git clone ssh://${MY_EOL}@osm.etsi.org:29418/osm/${MDG}.git
  (cd "${MDG}" && curl https://osm.etsi.org/gerrit/tools/hooks/commit-msg > .git/hooks/commit-msg ;
chmod +x .git/hooks/commit-msg)
  git -C ${MDG} checkout ${BRANCH}
done
```

Work in your environment

- Check README.md file
- Recommended editor: VSCode
- Make sure that Python 3.10 is installed in your environment

```
python3 --version
Python 3.10
```

- If you do not have Python 3.10, please follow the links to install Python 3.10
  - For Windows 10 and 11: https://www.tomshardware.com/how-to/install-python-on-windows-10-and-11
  - For Ubuntu 22.04|20.04|18.04: https://computingforgeeks.com/how-to-install-python-on-ubuntu-linux-system
- Install dependencies (ideally in a virtual environment)

```
python3 -m pip install -r requirements.txt -r requirements-dev.txt -r
requirements-test.txt
```

Open Source MANO

Testing before committing

Testing your commits

- Every commit in OSM is validated in a Jenkins CI/CD pipeline:
  - Per-Project validation: check code format, code linting, and run unit tests
  - E2E sanity tests

- You should validate your commits locally before committing
  - Otherwise, Jenkins will score it with -1

- The recommended way is to run from the Project folder the following command: `./devops-stages/stage-test.sh`
  - This is equivalent in most of the projects to run `tox`

Testing your commits using E2E Robot Tests

- For complicated changes, you need to make sure that your code does not break anything
- Run Robot tests using opensourcemano/tests image:

```
docker run --rm=true --name tests -t --env-file envconfig.rc \
           -v ~/.config/openstack/clouds.yaml:/etc/openstack/clouds.yaml \
           -v ~/tests/reports:/robot-systest/reports \
           opensourcemano/tests:testing-daily \
           -t sanity
```

- Different tags can be used (there is one for each test suite)
- Check README of osm/tests project for more details: https://osm.etsi.org/gitlab/osm/tests/-/blob/master/README.md

Contribute your changes

Add Apache license to the header of new files

If you are adding a new file, add the apache license to the header of the file. Apache license allows users to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software under the terms of the license, without concern for royalties.

```
#######################################################################################
# Copyright ETSI Contributors and Others.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#######################################################################################
```

Add Release Notes

- For RO and common projects, release notes are mandatory for all the changes.
- Add release notes using tox
  - `tox -e release_notes "<your_text>"`
- Go the release notes file and update the related section (upgrade, security, feature, prelude, other, fixes ..) of file
- Add the release notes file to the commit
  - `git add release_notes/notes/<...>.yaml`

Commit changes to your local repository

- Add your changes to the staged area

```
git add example-file
```

- Commit the change to your local repository

```
git commit -s -m "Fix of bug XXXX fixed"
```

- The -s parameter is mandatory. It signs the commit by adding the following line at the end of your commit message:

```
Signed-off-by: Random Developer random@developer.example.org
```

Push your contribution to gerrit

- OSM uses **gerrit** as code review tool. All changes are pushed to gerrit, which is a kind of intermediary git repository, with web-based UI for review
- Do a pull with rebase before pushing your contribution in order to merge latest changes made by other users. This avoids conflicts in Gerrit.
  ```
  git pull --rebase
  Note: you can force git to use always the --rebase option with:
  git config --local pull.rebase true
  ```
- Push your change to the appropriate branch (e.g. master).
  ```
  git push origin HEAD:refs/for/master
  ```
- You can review your contribution on Gerrit web interface:
  https://osm.etsi.org/gerrit/#/q/status:open

Last but not least

- Gerrit will send you email notifications about the progress of your change
- Check Jenkins validation
  - If -1, check Jenkins output to see the errors
- Code submitted to Gerrit, will be merged only after getting
  - Code Review +2
  - Verified +1

Additional how-to's

- Developer guide: https://osm.etsi.org/docs/developer-guide
- How to amend a change and re-contribute it: https://osm.etsi.org/docs/developer-guide/05-git-review.html?highlight=amend#amending-a-change
- How to push and review changes using git-review: https://osm.etsi.org/docs/developer-guide/05-git-review.html?highlight=amend#using-git-review-to-push-and-review-changes
- How to run Robot tests: https://osm.etsi.org/gitlab/osm/tests/-/blob/master/README.md

Open Source MANO by ETSI

Thank You!

How to build a docker image of a component as it is done in Jenkins CI/CD pipeline (1/2)

- Build your project as it is done in Jenkins
- Clone devops repo at the same level that your project

```
git clone ssh://${MY_EOL}@osm.etsi.org:29418/osm/devops.git
(cd "devops" && curl
https://osm.etsi.org/gerrit/tools/hooks/commit-msg >
.git/hooks/commit-msg ; chmod +x .git/hooks/commit-msg)
```

- Start HTTP server

```
./devops/tools/local-build.sh --run-httpserver
```

- Build the debian packages associated to your project (e.g. NBI) and its dependencies

```
rm -f ~/.osm/httpd/*.deb
./devops/tools/local-build.sh --module NBI,IM,common stage-2
```

How to build a docker image of a component as it is done in Jenkins CI/CD pipeline (2/2)

- Build Docker image

```
./devops/tools/local-build.sh --module NBI stage-3
```

- Patch the image of the component

```
kubectl -n osm patch deployment nbi --patch '{"spec":
{"template": {"spec": {"containers": [{"name": "nbi", "image":
"opensourcemano/nbi:devel"}]}}}}'
kubectl -n osm scale deployment nbi --replicas=0
kubectl -n osm scale deployment nbi --replicas=1
```