Setting up Jobs in Nordix Zuul

This page describes how to setup the rebase and push-upstream jobs in Nordix Zuul.

Developers who are contributing code as reviews to OpenDev must use this process for push-upstream jobs. This process keeps the reviews on the Nordix *master* branch in lock step with reviews on the OpenDev *master* branch.

This process assumes that your userid on the Nordix Gerrit and the OpenDev Gerrit are the same. You can check if this is the case by clicking these links

- Username on Nordix Gerrit: https://gerrit.nordix.org/#/settings/
- Username on OpenDev Gerrit: https://review.opendev.org/#/settings/

The process does not work in cases where your Nordix and OpenDev userids differ as the tooling in Nordix needs to know the userid (username) in OpenDev to be able to push changes upstream in your name.

Contents

1 Initial Configuration

1.1 Permit the Nordix Zuul to upload changes to OpenDev on your behalf

You must store the Nordix Zuul public key in the OpenDev Gerrit in order to enable pushing of reviews directly from the Nordix Gerrit to the OpenDev Gerrit under your username. You need to carry out this step only once.

1. Log onto the **OpenDev Gerrit**
2. in SettingsSSH Public Keys, click "Add Key..." (or click this link https://review.opendev.org/#/settings/ssh-keys)
3. Paste the root@zuul-master public key below into the text box

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC+VjYzLmmfLr11/cZWrJtofbxLDMFeQc6le3oUJlKTypSl
/gEEYDsRBbfpH0ypGz5IE7oAX+H7bSjetStBEWH6lQIgzGc8peUP+rl
/AqcBkvcktecpr6wWYt0MWoz+mGENB7PZhzaz8ez6F7ph3J4mwSB41sITdRBIoxr/oTmqUXBdYca7JTax4j3N3coa8
/PifYjnIuI1rK5FsCOB56TWp7L
/lANB9jrLQd6XwDIRawXSA8OOpOBrGIASF3bizt4bOcVnIdhR03ZiPJECvXg6XFlE4iQ6ZCFR1gkzBmz+TP2TBv2MA1FsQJ6FvBI8vNqDP
dX3DuftBFa1LBuQCZAcxmJ+l5ctZU4AruObM2yRuV4dlJQQpacRLrC+8q7YYGhCUqleOwUGYERPfjtCW/6PirGib//mVcdF28Yw
/Yu0DPJCmtFDppKNEUo+dYbazQQ8TksxjuMK0dJuLVFNaIUXoDgWGYzo/HuNc0zUjqnHGfD5YVn5W+O+DtgO05yEkL0= root@zuul-
master
```

4. Click "Add"

1.2 Add Repository in Zuul Configuration

You must need to add your repositories into zuul configuration before you plan to setting up jobs for those repos/projects in zuul

> Please note that, only Nordix Zuul admin can update the zuul.conf & main.yaml files in zuul-master. You may need to contact the Nordix Zuul admin to complete below pre-requisite tasks before setting up any jobs in Nordix Zuul

1. Login to zuul-master server
2. Ensure the existence of nordix & opendev gerrit connections in /zuul/zuul/etc_zuul/main.yaml file
3. Update the project name against proper gerrit connection in */zuul/zuul/etc_zuul/main.yaml* file
4. Restart the zuul-scheduler docker container

2 Setup push-upstream jobs in Nordix Zuul

2.1 Update the zuul-config Project to Define push-upstream Zuul Job

1. Clone the Nordix zuul-config repository

```
git clone ssh://<your-userid>@gerrit.nordix.org:29418/zuul-config
```

2. Create or update the secrets.yaml file as per your push-upstream job requirements under the *zuul.d* directory in the cloned *zuul-config*
3. Update the jobs.yaml file for your push-upstream job under the *zuul.d* directory in the cloned *zuul-config*
4. Update the projects.yaml file to define the push-upstream jobs to run on *nordix-patchset-create* pipeline under the *zuul.d* directory in the cloned *zuul-config*
5. Add or update the ansible roles and playbooks specific to your push-upstream jobs under playbooks directory in the cloned *zuul-config*

2.2 Submitting a Review

> ⚠ If you just want to push code through Nordix you can skip this section and simply use 'push-upstream' comment in your review to trigger the job.

1. Prepare your review on the ***master*** branch of the Nordix repo that is a downstream of an OpenDev repo. You can also prepare pull from OpenDev git repos if you wish.
    a. You may work on the review on the master branch from scratch
    b. You may create the review on the master branch by squashing or cherry picking reviews from development branches.
2. Submit your review to the ***master*** branch of the repo on the Nordix gerrit.

    ```
    git push <nordix> HEAD:refs/for/master</optional gerrit topic>
    ```
3. Upstream your review to the master branch of the repository on the OpenDev gerrit by logging onto the Nordix gerrit and posting a comment with the word ***push-upstream***. This triggers the Jenkins job that upstreams your review to the OpenDev master branch.
    Note: Never +2 or commit/merge a review directly on the Nordix master branch as this will cause the OpenDev and Nordix master branches to diverge.
4. The review is approved and merged in OpenDev. When the review is merged in OpenDev, the merge triggers a push of the review in the Nordix cloned repo. The Nordix cloned repo is therefore always in lock step with the OpenDev upstream original repo.
5. When the review in upstream gets merged, the status of your original review on Nordix Gerrit will also be automatically closed since Gerrit associates reviews using ChangeId and when sync job brings your commit from upstream to Nordix Gerrit, it will be in Merged state. No further work is required on your side.

Note: Steps 1. to 3. may be repeated to update reviews where changes to reviews are required.

2.3 Uploading a Modified Review and Submitting it through Nordix to OpenDev

The review is created on the Nordix gerrit and is then transferred to the OpenDev gerrit.

1. Make whatever changes and additions you require to the review and check it in locally.

    ```
    git commit --amend
    ```
2. Push your commit to the Nordix gerrit as a patch set on an existing review.

    ```
    git push <nordix> HEAD:refs/for/master</optional gerrit topic>
    ```

    You can check that you review has a new patch set on the master branch of the Nordix gerrit.

3. Transfer the review from the Nordix gerrit to the OpenDev gerrit. Post a comment on the Nordix gerrit containing the magic word "push-upstream". You can now check that your review has a new patch set on the master branch of the OpenDev gerrit.

3 Setup rebase jobs in Nordix Zuul

3.1 Update the zuul-config Project to Define rebase Zuul Job

1. Clone the Nordix zuul-config repository

    ```
    git clone ssh://<your-userid>@gerrit.nordix.org:29418/zuul-config
    ```
2. Create or update the secrets.yaml file as per your rebase job requirements under the *zuul.d* directory in the cloned *zuul-config*
3. Update the jobs.yaml file for your rebase job under the *zuul.d* directory in the cloned *zuul-config*
4. Update the projects.yaml file to define the rebase jobs to run on *periodic & opendev-change-merge* pipelines under the *zuul.d* directory in the cloned *zuul-config*
5. Add or update the ansible roles and playbooks specific to your rebase jobs under playbooks directory in the cloned *zuul-config*

> ⚠ If you just want to rebase code through Opendev Gerrit, simply use 'remerge' comment in the Opendev Gerrit change to trigger the rebase job