OOM Gating

OOM gating has been introduced for **Dublin**.

It consists of a deployment followed by a set of tests on patchsets submitted to **OOM repository**.

The CI part is managed on gitlab.com and the deployment is executed on ONAP Orange lab.

The goal is to provide a feedback - and ultimately to vote - on code change prior to merge to consolidate the Master branch.

The developer can evaluate the consequences of his/her patchset on a fresh installation.

The gating is triggered in 2 scenarios:

- new patchset in OOM (so if you submit 8 patchsets, gating will be queued 8 times..)
- comment with the magic word *oom_redeploy* is posted in the Gerrit's comment section

Please note that it is just an indicator, as it is based on the OOM Master branch, the errors reported in the results may be due to already existing code and may not be related to the patch itself.
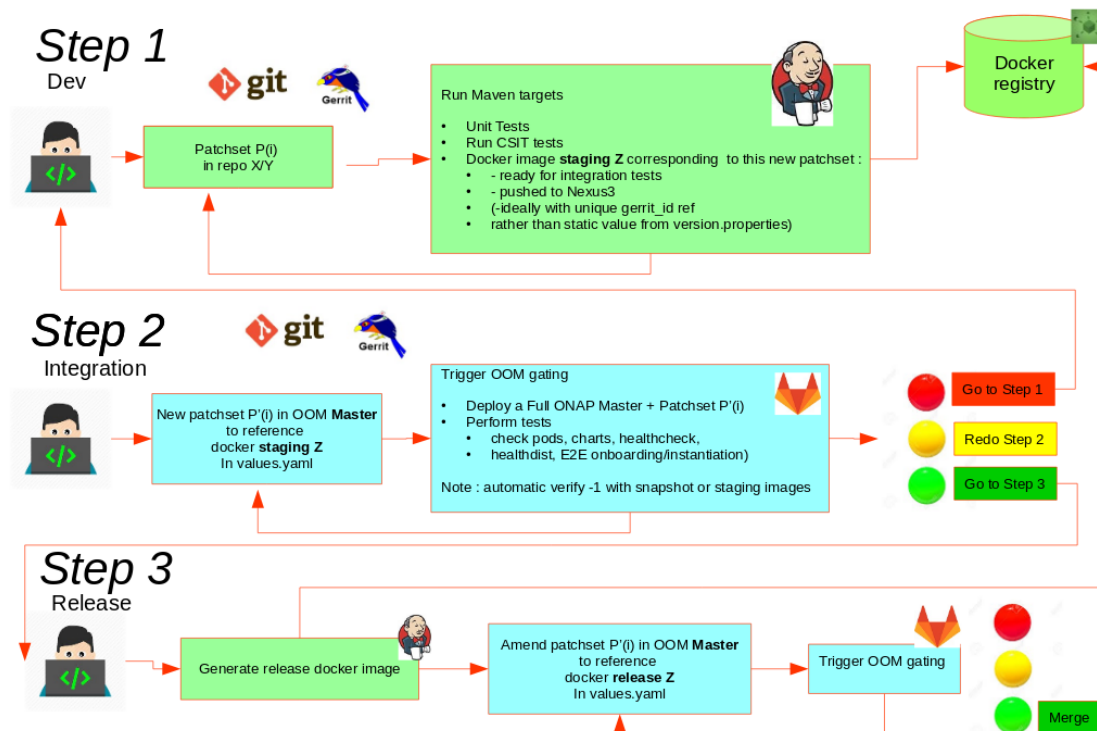
It is trivial to guess when the patch has nothing to do with the errors, but in some cases some pods or helm charts may be failed prior to the patchset modifying code from the same component.

The goal is to converge towards a rolling release, it means that Master would be always valid but it may take some time and would require some evolution in the management of docker versioning.

The Gating process can be described as follows:
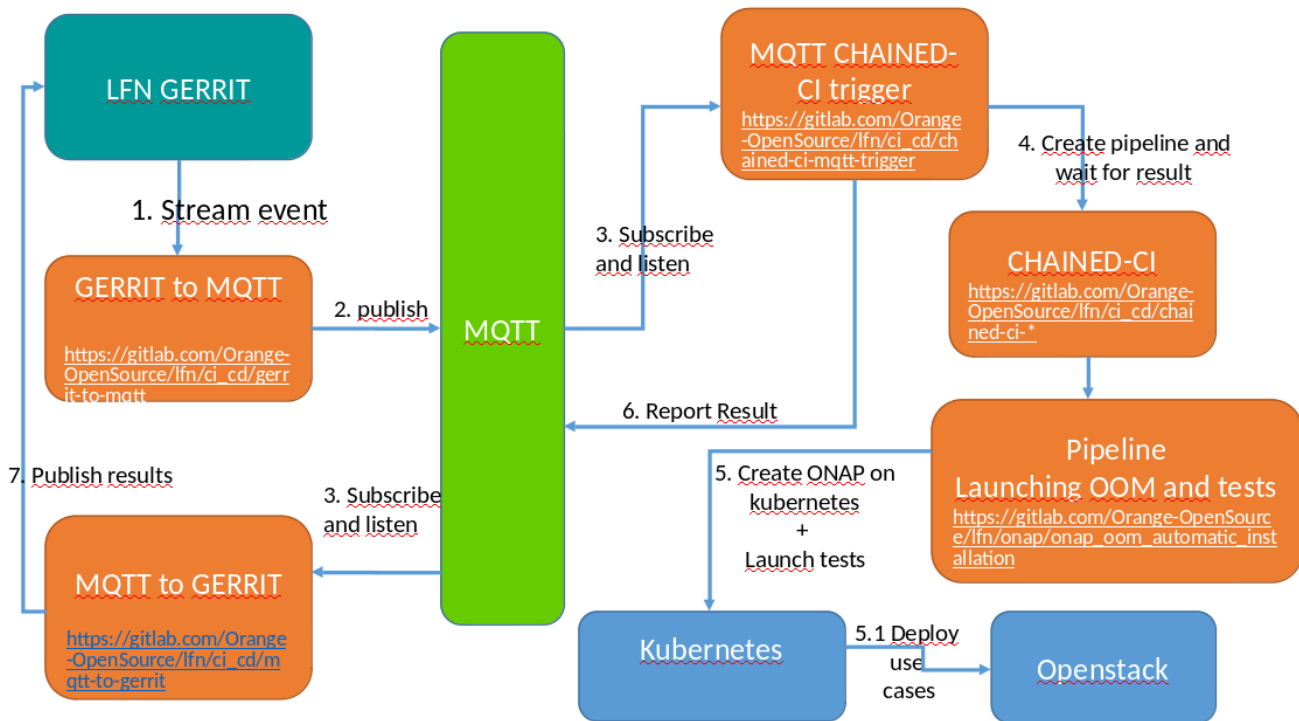
In order to simplify the integration and avoid depenencies towars manifest or branches that become quickly out of sync a 3 steps gating was introduced and can be described as follows:

Step 1
Dev

Patchset P(i)
in repo X/Y

Run Maven targets
- Unit Tests
- Run CSIT tests
- Docker image **staging Z** corresponding to this new patchset :
  - - ready for integration tests
  - - pushed to Nexus3
  - (-ideally with unique gerrit_id ref
  - rather than static value from version.properties)

Docker
registry

Step 2
Integration

New patchset P'(i) in OOM **Master**
to reference
docker **staging Z**
In values.yaml

Trigger OOM gating
- Deploy a Full ONAP Master + Patchset P'(i)
- Perform tests
  - check pods, charts, healthcheck,
  - healthdist, E2E onboarding/instantiation)

Note : automatic verify -1 with snapshot or staging images

Go to Step 1

Redo Step 2

Go to Step 3

Step 3
Release

Generate release docker image

Amend patchset P'(i) in OOM **Master**
to reference
docker **release Z**
In values.yaml

Trigger OOM gating

Merge

On patchset submission, the gating chain is triggered. It can be seen in the gerrit History windows.

1 : Gating process initiated
2 : Gating deployment started (timeout = 10800s = 180m
3 : Gating completed

A fourth notification is done to display the results in the gerrit review.

Please note that if you submit another patchset, it will retrigger the deployment and testing sequence. At this moment there is no filter so even a doc change triggers the pipeline.

At the end of the processing the ONAP jobDeployer reports the results

If you follow the links you will reach the xtesting-onap gitlab page (on gitlab.com).

Note you need a gitlab account to access these pages.

This link is unique and corresponds to the gating pipeline relating a specific patchset.

Gating pipeline History

Gating CI Console

Test

Download results

Patchset_id

Gerrit_id

You can download the artifact that corresponds to this specific test.

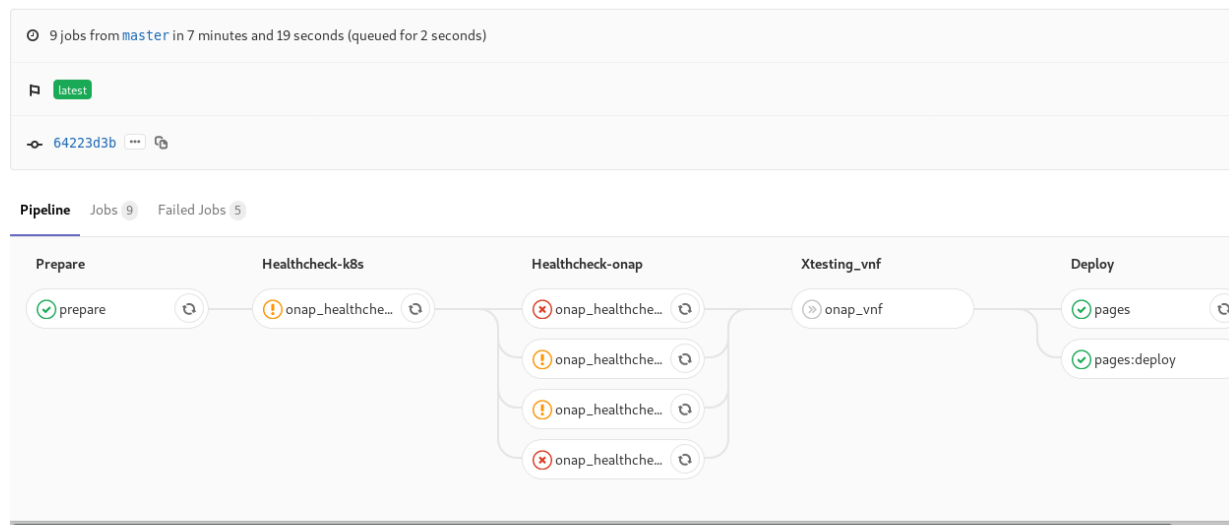If you're downloading all of the results, first you need to select the pipeline.

Link to the full pipeline

You should see the following menu

Pipeline Jobs 9 Failed Jobs 5

Prepare	Healthcheck-k8s	Healthcheck-onap	Xtesting_vnf	Deploy
✓ prepare ↻	⚠ onap_healthche... ↻	✗ onap_healthche... ↻	» onap_vnf	✓ pages ↻
		⚠ onap_healthche... ↻		✓ pages:deploy
		⚠ onap_healthche... ↻		
		✗ onap_healthche... ↻		

Then if you want to download the artifact with all the results, you must select the page stage, then click *download*.

Please see Where can I find the list of supported use cases? for the list of tests