

# ONAP CI/CD overview

Src : <https://docs.onap.org/projects/onap-integration/en/latest/integration-CICD.html>



Tasks

- Gerrit verification jobs
- Docker build execution chains
- CSIT test execution

- Gating
- Daily/Weekly deployment
- Azure Staging deployment

Resp

LF IT

Integration / Orange / DT / Nokia

Cloud resources

Vexxhost

- Azure
- Orange DC
- DT DC
- Nokia DC



- Jenkins files are in <https://github.com/onap/ci-management> under jjb > ...
- Verification jobs
  - New files (linter/verification jobs) creates when new projects are created
  - Files updated regularly (versions)
- Build chains (LF scripts) used for project but also simulator dockers and all xtesting dockers (testing dockers)
  - Xtesting docker rebuilt and repushed to nexus every day
  - In case of Nexus/Jenkins issues => it happened that the xtesting disappeared => issue in gating/daily/weekly
- CSIT tests
  - functional test
  - Recommendation since Guilin to bring such code back to projects
  - Integration not really involved except the historical +2 and regulare cleaning because tests are not all maintained
- Ci-management : Bartek and myself have the +2 (also Lasse but he stepped down from integration)
- Possibility to force rebuild through jenkins Web UI : <https://jenkins.onap.org/>

# History

- Historically some CI was done on windriver lab using a local jenkins deployed in windriver lab (broken since frankfurt)
- LF IT was not able to support full E2E CI chains
  - not enough cloud resources (ONAP is still huge and LF IT has no its own resource (only Vexxhost) + No share of the Vexxhost creds with the community (legal aspects))
  - No ONAP scripts only generic scripts provided through jenkins
- Orange introduced the Gating for EI Alto release
  - Consuming public ansible roles for the installation of kubernetes, ONAP, roles hosted in gitlab.com
  - Leveraging gitlab.com CI resources
  - Using jenkins built xtesting docker for the testing part
  - Azure cloud resources were added to match gating needs



CI/CD

- CI leveraging chained-ci project allowing to chain CI chains of different gitlab projects

Chain name

Test part  
(consuming xtesting dockers)

ONAP deployment

Openstack Tenant creation  
(for testing)

The screenshot displays the 'chained-ci UI' interface. At the top, there is a header 'chained-ci UI'. Below it, a 'Scenario filter' input field contains the text 'filter'. To the right, a 'Next update' indicator shows '37 / 60' with a note: '(Please set filter or optimize it to have a better update time)'. A 'New pipeline' button is also visible.

The main content area shows three pipeline stages, each with a 'Chain name', 'Branch', 'Duration', and 'Timestamp'. Each stage has a 'Play' button and a 'Refresh' button. The jobs for each stage are listed in a grid format.

Chain name	Branch	Duration	Timestamp	check	apps	infra_install	config
onap_daily_pod4_master	master	0 min	2021-09-06 10:20:47	<input type="radio"/> onap_test	<input type="radio"/> onap_deploy	<input checked="" type="radio"/> vnf_project_deploy	<input checked="" type="radio"/> config
onap_oom_gating_azure_4	master	0 min	2021-09-06 09:37:52	<input type="radio"/> onap_test	<input checked="" type="radio"/> onap_deploy	<input checked="" type="radio"/> build_clamp <input checked="" type="radio"/> build_so	<input checked="" type="radio"/> config
onap_oom_gating_azure_3	master	0 min	2021-09-06 08:30:17	<input checked="" type="radio"/> onap_test	<input checked="" type="radio"/> onap_deploy	<input checked="" type="radio"/> build_clamp <input checked="" type="radio"/> build_so	<input checked="" type="radio"/> config

Blue arrows from the text on the left point to the 'Chain name', 'Test part', 'ONAP deployment', and 'Openstack Tenant creation' labels, which correspond to the jobs in the pipeline stages.



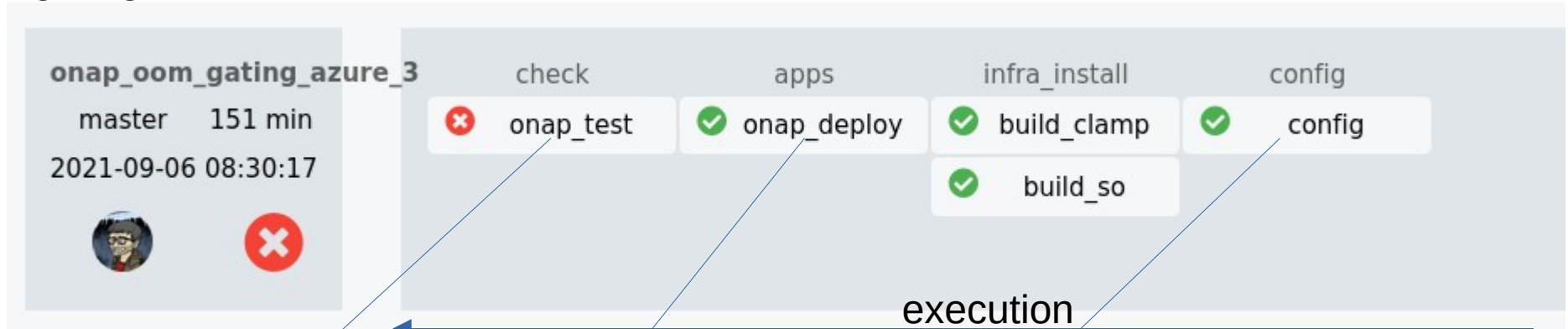
Each chain can be composed of several projects  
Chained-ci allows a mix and match approach assuming that each project generates outputs that can be consumed as inputs for the next project



The list of the projects :

- installation
  - vnf\_project\_deploy : create user/project on openstack
  - infra\_deploy : create VM on openstack
  - aks\_inframanager : create VM on AKS
  - k8s\_deploy
  - onap\_deploy
- Test
  - k8s\_test
  - onap\_test
- Build
  - build\_so (for SO gating – rebuild SO dockers including the patch)
  - build\_clamp (for clamp gating)

## A gating chain



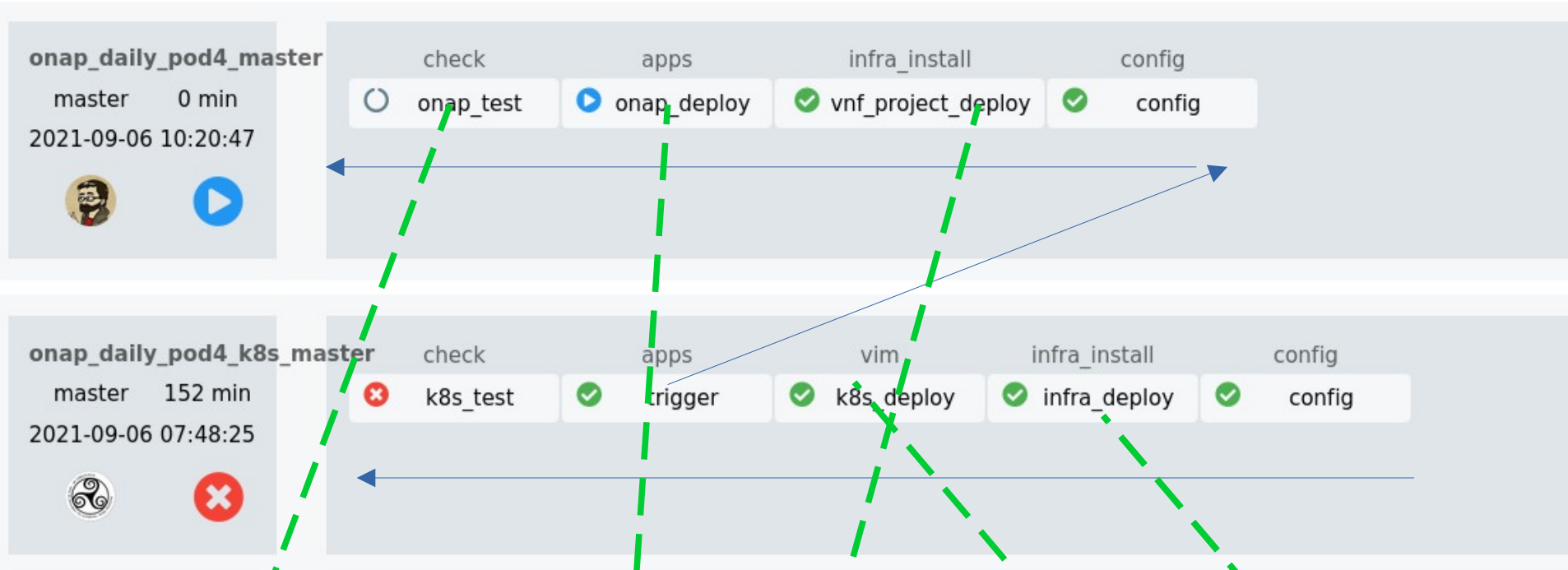
<https://gitlab.com/Orange-OpenSource/lfn/onap/xtesting-onap/-/pipelines/365761859>

[https://gitlab.com/Orange-OpenSource/lfn/onap/onap\\_oom\\_automatic\\_installation/-/pipelines/365714060](https://gitlab.com/Orange-OpenSource/lfn/onap/onap_oom_automatic_installation/-/pipelines/365714060)

[https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/chained-ci/-/jobs/1565053486](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/chained-ci/-/jobs/1565053486)



# A daily/weekly chain



Test ONAP

Deploy ONAP

Deploy k8s

Create user/project/Vms for ONAP tests

Create user/project/Vms for K8S

Brick	project
Installation k8s (kubespray)	<a href="https://gitlab.com/Orange-OpenSource/lfn/infra/kubespray_automatic_installation">https://gitlab.com/Orange-OpenSource/lfn/infra/kubespray_automatic_installation</a>
Installation k_s (RKE2)	<a href="https://gitlab.com/Orange-OpenSource/lfn/infra/rke2_automatic_installation">https://gitlab.com/Orange-OpenSource/lfn/infra/rke2_automatic_installation</a>
Installation ONAP	<a href="https://gitlab.com/Orange-OpenSource/lfn/onap/onap_oom_automatic_installation">https://gitlab.com/Orange-OpenSource/lfn/onap/onap_oom_automatic_installation</a>
Test ONAP	<a href="https://gitlab.com/Orange-OpenSource/lfn/onap/xtesting-onap">https://gitlab.com/Orange-OpenSource/lfn/onap/xtesting-onap</a>
Chained-CI	<a href="https://gitlab.com/Orange-OpenSource/lfn/ci_cd/chained-ci">https://gitlab.com/Orange-OpenSource/lfn/ci_cd/chained-ci</a>
...	...

All projects are under Apache v2 licence, not moved to git.onap.org because leveraging free gitlab-ci mechanisms, ONAP contributors are already members of these projects

# Ansible..

- All the projects are using ansible roles/playbooks to perform the different actions

Name	Last commit	Last update
📁 niter_plugins	initial commit	3 years ago
📁 inventory/group_vars	🚚 <a href="#">Use specific helm version for kubespray</a>	8 months ago
📁 roles/configure	🚚 Use specific helm version for kubespray	8 months ago
📁 scripts	👤 update chained-ci-tools scripts to latest version	1 year ago
📁 vars	🔒 update vaulted credentials	2 years ago
📁 .gitignore	🌟 Monitoring	2 years ago
📄 .qitlab-ci.orange.yml	🐛 try to make it work	2 years ago

# And chained-ci..

- Chained-ci is a software allowing to coordinate the different gitlab-ci the projects to be chained

[https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/chained-ci](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/chained-ci)

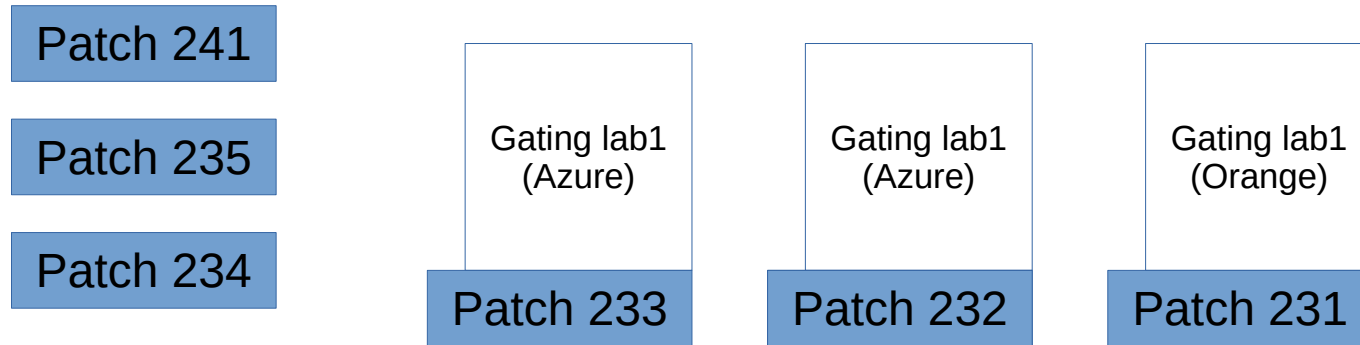
# Some more words on the Gating



We listen to OOM, SO and Clamp gerrit changes

When a change is detected we add it in a MQTT queue

When a Gating lab is available, it initiates a gating and remove the patch from the queue



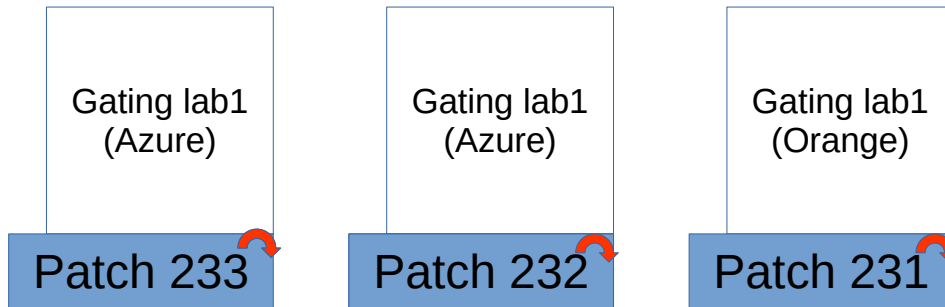
New patch arrives..nothing changes, all labs busy  
Message in gerrit 237 number 4 in the queue

Patch 237

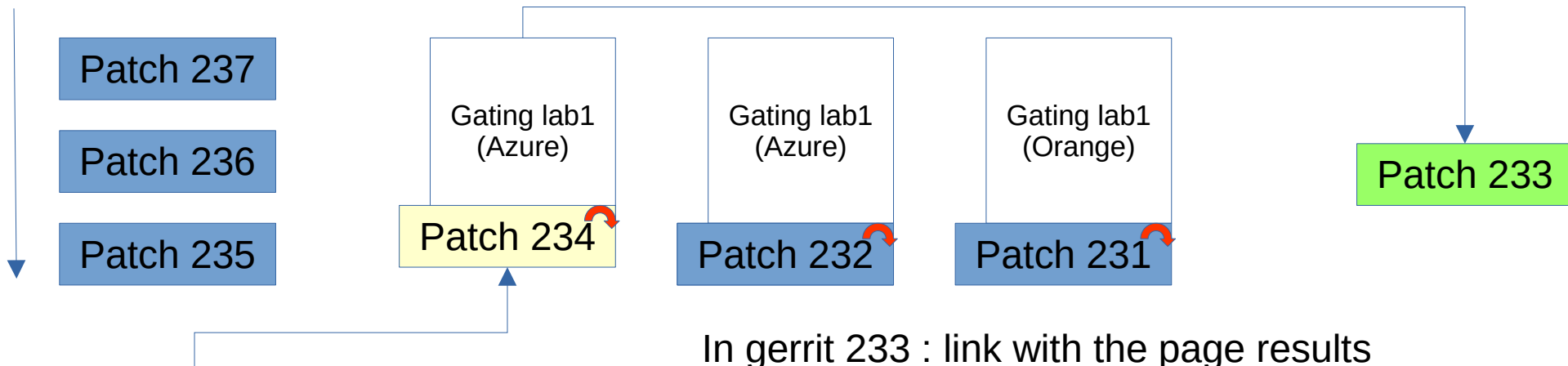
Patch 236

Patch 235

Patch 234



Let's assume that Patch 233 gating is finished...



In gerrit 233 : link with the page results  
In gerrit 234 : message gating started

The code to manage the MQTT/gerrit notification is also open source and available here :

- [https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/mqtt-to-gerrit](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/mqtt-to-gerrit)
- [https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/gerrit-to-mqtt](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/gerrit-to-mqtt)
- [https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/chained-ci-mqtt-trigger](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/chained-ci-mqtt-trigger)
- [https://gitlab.com/Orange-OpenSource/lfn/ci\\_cd/python-gerrit](https://gitlab.com/Orange-OpenSource/lfn/ci_cd/python-gerrit)

# How to improve the Gating

- Provide Cloud resources..we mentioned for El Alto at peak (~ RC) 9 simultaneous labs were needed have 0 queuingThe more cloud we have, the more resilient we are..(there are regular issues in infra..)
- With 2 labs, before M3 we reached up to #20 in queue...so 10 per lab assuming that a complete gating is ~1h30 ....feedback can be given 15h after patch submission..



# Recent work

- We worked on the creation of collections
  - [https://gitlab.com/Orange-OpenSource/lfm/infra/rke2\\_automatic\\_installation\\_collection](https://gitlab.com/Orange-OpenSource/lfm/infra/rke2_automatic_installation_collection)
  - [https://gitlab.com/Orange-OpenSource/lfm/infra/kubernetes\\_collection](https://gitlab.com/Orange-OpenSource/lfm/infra/kubernetes_collection)
  - [https://gitlab.com/Orange-OpenSource/lfm/infra/infra\\_collection](https://gitlab.com/Orange-OpenSource/lfm/infra/infra_collection)
  - <https://gitlab.com/Orange-OpenSource/lfm/onap/onap-testing-collection>
  - ....
- We also now can build a docker with all the needed collections as an alternative to chained-ci
- But these changes are still experimental