lf-releng-global-jjb

Release v0.88.3.dev0+cdd1193

Linux Foundation Releng

Jul 25, 2023

CONTENTS

Linux Foundation Release Engineering Global Jenkins Job Builder (JJB) Documentation.

Global-JJB is a library project containing reusable Jenkins Job Builder templates. Developed for LFCI to deploy management Jenkins jobs to an LF managed Jenkins instance, there are other jobs defined which may be helpful to projects that use the same build technology. The intention is to help projects save time from having to define their own job templates.

CHAPTER

ONE

RELEASE NOTES

1.1 Release Notes

1.1.1 v0.88.3

Deprecation Notes

- Remove WhiteSource job templates. WhiteSource tool for code scanning is no longer in use for LF projects.

1.1.2 v0.88.2

Bug Fixes

- Gradle build job needs to run docker login step to allow docker operations and access to Nexus3.
- Gerrit release verify and merge jobs need to have a {stream} variable in their name to allow the creation of the same job under different branches without conflicting.

1.1.3 v0.88.1

Bug Fixes

- Pass target file where the config file should be created. New version of packer accepts only .json or .hcl extension filenames.

1.1.4 v0.88.0

Prelude

As of packer version 1.7.0 HCL2 is the preferred way to write Packer templates. HCL2 preserves existing workflows while leveraging HCL2's advanced features like variable interpolation and configuration composability.

Upgrade Notes

- Upgrade packer version to v1.9.1. JSON format templates are deprecated and no longer work with packer version > 1.9.x. Project specific templates require to be upgraded to HCL2 format.

Deprecation Notes

- Support for '.json' templates will be removed from common-packer in subsequent releases. Therefore, the jobs are expected to work with older templates.

Bug Fixes

- Rewrite packer jobs to work with HCL2 format.

1.1.5 v0.87.1

Bug Fixes

- Add a JDK configuration step in the Gradle based jobs.

1.1.6 v0.87.0

New Features

- Add initial lf-gradle file for gradle based builds.

1.1.7 v0.86.9

Bug Fixes

- Pin urllib3 to <2.0.0 for the JJB cleanup

  The latest version of module breaks compatibility with python-jenkins.

  Error:

  ```
  ValueError: Timeout value connect was <object object at
  0x7fe57a4948a0>, but it must be an int, float or None.
  ```

  Reference:

  Launchpad#2018567 <https://bugs.launchpad.net/python-jenkins/+bug/2018567>

1.1.8 v0.86.7

Bug Fixes

- Pin urllib3~=1.26.15 in pypi distribution jobs

1.1.9 v0.86.6

Bug Fixes

- Pin urllib3~=1.26.15 in jjb-deploy job

1.1.10 v0.86.4

Bug Fixes

- Pin urllib3 to <2.0.0 for the RTD jobs

  The latest version of module breaks compatibility with RTDv3 jobs during tox install and run.

  Error:

  ```
  ValueError: Timeout value connect was <object object at
  0x7fe57a4948a0>, but it must be an int, float or None.
  ```

  Reference:

  > *Launchpad#2018567 <https://bugs.launchpad.net/python-jenkins/+bug/2018567>*

1.1.11 v0.86.3

Bug Fixes

- Pin urllib3 to <2.0.0 for the verify jobs

  The latest version of module breaks compatibility with python-jenkins.

  Error:

  ```
  ValueError: Timeout value connect was <object object at
  0x7fe57a4948a0>, but it must be an int, float or None.
  ```

  Reference:

  > *Launchpad#2018567 <https://bugs.launchpad.net/python-jenkins/+bug/2018567>*

1.1.12 v0.86.2

Bug Fixes

- Pin urllib3 to <2.0.0

  The latest version of module breaks compatibility with python-jenkins.

  Error:

```
ValueError: Timeout value connect was <object object at
0x7fe57a4948a0>, but it must be an int, float or None.
```

  Reference:

  Launchpad#2018567 <https://bugs.launchpad.net/python-jenkins/+bug/2018567>

1.1.13 v0.86.1

Bug Fixes

- Use the GERRIT_REFSPEC while pushing code and tags separately.

1.1.14 v0.86.0

Prelude

JJB 5x treats recursive parameters as an error.

JJB's 5.0.1 has a bug that return an error. TypeError: 'NoneType' object is not a mapping

JJB's YAML parser is re-written in JJB 5.x release for a fine-tuned control over YAML parsing, YAML objects and parameter expansion logic.

Known Issues

- This breaks the JJB test on the existing ci-man repositores when the macro is null value.
- This breaks backward compatibility with older version of JJB therefore care must be taken during upgrade.

Upgrade Notes

- Upgrade Jenkins-job-builder to 5.0.2 which has the fix for the issue.

  Ref: https://review.opendev.org/c/jjb/jenkins-job-builder/+/880589

- Upgrade Jenkins-job-builder to 5.0.1 as the default version.

  Ref: https://review.opendev.org/c/jjb/jenkins-job-builder/+/871965

Bug Fixes

- Remove recursive parameters set as defaults for packagecloud jobs
- Use the python3 module option instead of calling pip directly. This fixes the issue when pip is not available the PATH.

1.1.15 v0.85.0

Deprecation Notes

- Remove daily cron on maven-stage and maven-docker-stage jobs. Cron triggers stay configurable.

Bug Fixes

- Fix condition before pushing the object.
- Address the problem where the tag is not pushed to the mainline branch therefore causing the tag missing in the git history.

  To fix this check commit count between the HEAD and origin/${GERRIT_BRANCH} before the fetch and merge operation. This is done to ensure that the tag lands on the target branch. If the branch has already moved forward from the tagging point, then a spur commit is created for the tag.
- Un-pin tox version from 3.27.1 and remove tox-pyenv. Testing has demonstrated that tox-pyenv is no longer required to obtain correct Python runtime versions when running tests. Also, removed Python 3.8 from the VENV setup where it was being specifically requested.

  Due to unpinning of the tox version, tox.ini configuration files may need modifying to reflect a change in configuration syntax; where whitelist_externals needs to be replaced with allowlist_externals.

1.1.16 v0.84.0

New Features

- Introduce Docker Snyk CLI scanner jobs. These jobs can be triggered to download the latest version of Snyk's CLI scanner and trigger a scan for Docker based repos. These jobs produce a report which is published into Snyk's dashboard. These reports are fetched and reflected back into the LFX Security tool.
- Introduce Go Snyk CLI scanner jobs. These jobs can be triggered to download the latest version of Snyk's CLI scanner and trigger a scan for Go based repos. These jobs produce a report which is published into Snyk's dashboard. These reports are fetched and reflected back into the LFX Security tool.
- Introduce Maven Snyk CLI scanner jobs. These jobs can be triggered to download the latest version of Snyk's CLI scanner and trigger a scan for Maven based repos. These jobs produce a report which is published into Snyk's dashboard. These reports are fetched and reflected back into the LFX Security tool.
- Introduce Python Snyk CLI scanner jobs. These jobs can be triggered to download the latest version of Snyk's CLI scanner and trigger a scan for Python based repos. These jobs produce a report which is published into Snyk's dashboard. These reports are fetched and reflected back into the LFX Security tool.

Bug Fixes

- Add SNYK_CLI_OPTIONS parameter which can be used to pass additional Snyk CLI options as per https://docs.snyk.io/snyk-cli/cli-reference.
- The path and command for update-alternatives/alternatives was not being set correctly between CentOS7/8 and was incorrect under all tested ubuntu versions. It did not seem to cause jobs to break, so was perhaps not being detected in all cases.
- The latest (2.42.0.01) clm-maven-plugin introduced an error in our environment.

  Failed to execute goal com.sonatype.clm:clm-maven-plugin:2.42.0-01:index (default-cli) on project babel: Failed to invoke Maven build. Maven execution failed, exit code: 1 -> [Help 1]

  This fix will pin the clm-maven-plugin to the previous version (2.41.0-02)

1.1.17 v0.83.5

Bug Fixes

- The Nexus IQ script was outputting the wrong variable during execution, which could be misleading in the job console logs. Also added a warning message if the NEXUS_TARGET_BUILD variable has not been set/populated. Reports were not receiving module dependencies so the script has been amended to download them into the target directory, as per the Nexus IQ documentation.

1.1.18 v0.83.4

Bug Fixes

- Pin setuptools to the latest version before v66.0.0 to avoid PEP-440 non conforming versions errors while packages we use fix their version string formats to be compatible. To maintain congruency we are pinning setuptools to 65.7.0 in all places under global-jjb.

1.1.19 v0.83.3

Bug Fixes

- setuptools 66.0.0 enforces specific version number check. A lot of plugins will fails on this, and generates the following error pkg_resources.extern.packaging.version.InvalidVersion

  To temporarily fix this, we pin setuptools to <66.0.0

  Further information https://github.com/pypa/setuptools/issues/3772#issuecomment-1384342813 https://setuptools.pypa.io/en/latest/history.html#v66-0-0

1.1.20 v0.83.2

Bug Fixes

- Pin tox version on rtd-verify.sh script until tox>=4.0.2 and tox-pyenv>=1.1.0 compatibility issues and bugs are resolved.

1.1.21 v0.83.1

Bug Fixes

- Pin tox version until tox>=4.0.2 and tox-pyenv>=1.1.0 compatibility issues and bugs are resolved.

1.1.22 v0.83.0

New Features

- Replace the usage of plaintext sonarcloud api token with a Jenkins credential. The default value for the credential ID is 'sonarcloud-api-token' and we are standarizing it for all projects so this parameter does not require an override.

Upgrade Notes

- Parameter sonarcloud-api-token is NOT required anymore after upgrading to this GlobalJJB version, please remove it from all JJB projects.

1.1.23 v0.82.4

Bug Fixes

- Remove unnecessary quotes around the variable that processes glob patterns.

1.1.24 v0.82.3

Bug Fixes

- Replace Nexus IQ build Target from "${REQUIREMENTS_FILE}" to "${NEXUS_TARGET_BUILD}". The scanner is only including the requirements.txt file in its scan which should not contain other information than python package requirements. Instead, use a "${NEXUS_TARGET_BUILD}" parameter which the user can optionally provide to the scanner to indicate a file or directory to include in the scan. By default, this variable is configured to scan all files in the repo.

1.1.25 v0.82.2

Prelude

The SBOM generator script creates an spdx file in the root level. When the artifacts are staged the file gets overwritten.

Bug Fixes

- Create the spdx file as ${PROJECT}-sbom-${release_version}.spdx and then copy the spdx file under the namespace ${group_id_path} dir.
- Minor changes to improve openstack cleanup scripts.

1.1.26 v0.82.1

Known Issues

- line 48: tox: command not found

Bug Fixes

- Addresses a bug whereby the openstack orphaned objects/ports scripts exit early with an error when grep/awk do not match any orphaned objects. The fix allows jobs using the scripts to continue when no cleanups operations are required.
- The venv created for tox is unavailable when the semantics of the script are split across files, therefore ensure venv is created with –venv-file option and set.

1.1.27 v0.82.0

New Features

- Added a script to cleanup generic openstack objects.

Bug Fixes

- Addresses failures when cleaning up orphaned openstack ports. The main “openstack <object> list” command no longer accepts the “-c created_at” option, which has been moved to a property of the object and must now be queried with “openstack show object UUID”. Also, the created_at parameter sometimes returns “None” instead of a timestamp, and the existing version of the script does not catch this condition.

1.1.28 v0.81.6

Bug Fixes

- Remove line break in the docker_push_command variable causing the varible to not be set properly

1.1.29 v0.81.5

Prelude

PyPI verify jobs requires Python 3.x. The tox run picks up default version of python instead of the version made available through pyenv.

Known Issues

- Re-factor lf-activate-venv() to skip a return, while the venv is re-used, so that the PATH can be set.

Bug Fixes

- Fix the docker-push script which is broken due to a synxtax error causing the docker_push_command variable to not be set.
- Update the tox install and run script to Call lf-avtivate-venv().

1.1.30 v0.81.4

Bug Fixes

- Sonar CLI job needs to use the credential that matches the name of the project. That is, "sonar-token-{project-name}".

1.1.31 v0.81.3

New Features

- Add gerrit-cli-sonar and github-cli-sonar scanner job for non maven based repos. This job downloads a specific Sonar CLI version and runs sonnar-scanner on the code to produce a report which is pushed in SonarCloud.

1.1.32 v0.81.2

Prelude

OpenDaylight jenkins maven jobs with jdk17 and CentOS7 currently fails with a confusing message stating that the JAVA_HOME variable is not correctly set. This can happen in various cases, usually when there is a mismatch between the jdk used by maven and the folder pointed by JAVA_HOME. It appears that openjdk17 is not available with CentOS7 and that the folder indeed does not exist

Known Issues

- Current message (JAVA_HOME variable is not set) is confusing and can lead to erroneous interpretations.

Bug Fixes

- Add a folder existence check in related script before propagating JAVA_HOME variable to other scripts. If no folder was found, try to find an approaching solution and exit in case of failure with a more relevant error message.
- Install yq in the venv that is called by the builder scripts of RTDv3 and docker jobs.

Other Notes

- Adapt and refactor code consequently to be more agnostic to distribution and jdk installation specificities

1.1.33 v0.81.1

Known Issues

- git-review tries to copy commit-msg hook to submodules with incorrect source file path (.git/hooks/commit-msg) and fails - the path should be ../.git/hooks/commit-msg if a relative path is used since the copy command is run in the submodule directory
- lf-activate-venv creates a virtual environment in the current working directory where lf-activate-venv is run. This clutters the repository and all the files for the virtual environment are added for update.

Bug Fixes

- Set 'core.hooksPath' with the absolute path of the top-level hooks directory so that the correct source path can be used regardless of the working directory.
- Use the correct command depending on the $install_args value to avoid creating an additional virtual environment in the current working directory.

1.1.34 v0.81.0

Prelude

Update openstack images with the auto update image requires more recent version of git-review > 2.2.

New Features

- Add support for a new option to set venv file.

  lf-activate-venv –venv-file /tmp/.robot_venv robotframework

  Modify lf-activate-venv() to allow creation of a venv file and re-use the venv to improve job performance. When a dependency is already installed, pip skips the package therefore reduces the time it takes to create venv in every script.

  Precedence for venv file.

  a. **Re-use an existing venv file if one exists.**

1. Use venv file path from –venv-file
2. Use default venv file path "/tmp/.os_lf_venv"

b. Create new venv when 1. and 2. is absent

Note: The default file "/tmp/.os_lf_venv" is created by a pre-build script (../shell/python-tools-install.sh).

In the situation where a fresh venv is required remove "/tmp/.os_lf_venv" before calling lf-activate-venv().

Update all the required scripts that call lf-activate-venv().

Known Issues

- Error:

  Errors running git rebase -p -i remotes/gerrit/master fatal: –preserve-merges was replaced by –rebase-merges

  Ref: https://review.opendev.org/c/opendev/git-review/+/818219

Upgrade Notes

- The previous version of git-review is incompatible with the latest version of git due to renaming flags. This is fixed in git-review 2.2.0.

Bug Fixes

- Clean up conditions introduce in the shell scripts, while these checks are performed within lf-activate-venv().

1.1.35 v0.80.2

Known Issues

- Error: openstack: command not found

Bug Fixes

- lf-pyver() fails to include the currently selected version in the output of 'pyenv versions', which makes the version change every time the local version is set by pyenv with the version from lf-pyver().

  Fix the command to extract the list of Python versions to include all the numeric versions in the list.

- Use lf-activate-venv to install openstack deps

  Using python-tools-install.sh for the pre/post build is not recommended approach for installing python dependencies since this installs the dependencies with *–user* option (removed in I821a86ac3b54f284e8).

  Instead use lf-activate-venv to setup an venv and pull in the required dependencies and save the path of the virtualenv in a temp file that can be checked before attempting to create a venv.

1.1.36 v0.80.1

Known Issues

- yq: command not found

Bug Fixes

- Install yq through lf-activate-venv rather than using the python tools install script.

1.1.37 v0.80.0

Known Issues

- ERROR: Not installed on host: python3.8.13 ERROR: Can not perform a '–user' install. User site-packages are not visible in this virtualenv.

Bug Fixes

- Set the default version to 'python3' instead of '3.8.x' since some of the older images may not have the specifc version installed. The default version is only used when lf-env.sh is not available.

  CR I821a86ac3b54f2 sets and uses python 3.x version made available by pyenv therefore remove the –user option which is no longer required.

1.1.38 v0.79.4

Known Issues

- Addresses problems found while troubleshooting IT-24352

Upgrade Notes

- Use pyenv whih is the standard way to manage, set and use a python3 installation on the system.

  The required version of python3 for all jobs should be > 3.8.x, to avoid PyPI dependencies conflicts with outdated versions. However, the lf-activate-venv() uses the system default version python installed through packages. This can cause warning and build failures that source lf-env.sh.

  Update lf-activate-venv to use pyenv versions of python3 installed through the lfit.python-install galaxy ansible role.

- Upgrade Packer version to v1.8.2. The version is more recent v1.8.2 and has security updates.

  Ref: https://github.com/hashicorp/packer/releases/tag/v1.8.2

Bug Fixes

- Fix lf-activate-env code comment. The comment suggests using just the version number `--python <x.y>`, however as per the code the correct format as per the code is `--python python<x.y>`
- Added support for debian in update-java-alternatives.sh (addresses potential Ubuntu detection bug) Safer handling of unset/null SONARCLOUD_JAVA_VERSION variable preventing java runtime issues

1.1.39 v0.79.3

Bug Fixes

- Copy SBOM report to the project's m2repo so that is signed by SIGUL and pushed in the same staging package as the maven artifacts.

1.1.40 v0.79.2

Bug Fixes

- Set lf-activate-env to use Python 3.8 while running lftools deploy logs. This fixes the below warnings which when jobs try to use default version of python 3.6 which is EOL.

  CryptographyDeprecationWarning: Python 3.6 is no longer supported by the Python core team. Therefore, support for it is deprecated in cryptography and will be removed in a future release.

  PythonDeprecationWarning: Boto3 will no longer support Python 3.6 starting May 30, 2022. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.7 or later.

1.1.41 v0.79.0

Bug Fixes

- Update to the latest version of SBOM (v0.0.15) that allows the usage of a custom maven settings file to resolve transitive dependencies. Update thebom-generator script to pass the project's global settings file and update the sbom file name so is better identifiable.

1.1.42 v0.78.0

Upgrade Notes

- Upgrade NexusIQ Client to more recent version 1.140.0-01.

1.1.43 v0.77.4

Deprecation Notes

- Our Sigul bridges now have publicly accessible DNS names, so it is no longer necessary to create an entry in the hosts file. Since this process relies on up-to-date IP addresses being configured in each project's global env vars, it can cause avoidable errors. It is therefore being removed.

1.1.44 v0.77.3

Bug Fixes

- Fix URL path indent, add a default ARG to the Dockerfile to remove WARNING. Set the .asc files permissions to jenkins after the sigul has signed the files.

1.1.45 v0.77.2

Bug Fixes

- Update the sigul-sign-dir.sh to sign artifacts using docker. The docker image is built on CentOS Streams 8/9. The newer version of sigul 1.1.1 available for CentOS 8 is not backwords compatible with the version of sigul on CentOS 7.

  As a temporary workaround build a CentOS7 docker image with sigul installed and use it for signing artificats on platforms where sigul is not readly available.

  Note: the executor node needs to have docker installed, so it can't be a "vanilla" build node but must be a docker node.

1.1.46 v0.77.0

Upgrade Notes

- Jenkins Job Builder 4.1.0 is now the default version. global-jjb has been pegged to version 2.8.0 since v0.55.3 released on 2020-07-21. Since this release JJB has dropped support for Python 2.7 and version 4.1.0 of JJB has required fixes needed for dealing with versions of plugins that are now shipping for Jenkins that cause issues.

  Projects that set override their JJB version should either remove the pin and take what global-jjb defaults to, or reset their pin to 4.1.0

1.1.47 v0.76.4

Bug Fixes

- SBOM's path flag does not work as expected. We need to introduce a new flag called SBOM_PATH to isolate the path where SBOM is going to be extracted to and executed from. By default this is set to $WORKSPACE but some projects need to execute the sbom from a different location in their code. See https://github.com/opensbom-generator/spdx-sbom-generator/issues/227
- Optionally run a script before and/or after maven goals. This will help add dependencies and post process builds with more flexibility to the project's needs.

1.1.48 v0.76.3

New Features

- Post-build script capture-instance-metadata.sh will attempt to choose the best possible version of Python to use. If the `$PYTHON` variable is set, this will be used. If not, we check to see if `python3` is available, as this should point to the latest version. If this is also not available, we run with the basic `python` command.

1.1.49 v0.76.0

New Features

- Append build result to cost.csv file. This would be useful to capture stats for unattended jobs and potentially send out reports to PTL's on resource usage stats.

1.1.50 v0.75.1

Bug Fixes

- Activate the virtual environment. Python may not be available by default on all versions.

1.1.51 v0.75.0

New Features

- Add new conditional builder step which calls a specific version of SPDX SBOM generator which runs a scan to generate a software bill of materials report in a specific repo.

1.1.52 v0.74.0

New Features

- Process orphaned coe clusters for K8S jobs

  K8s (COE cluster) jobs by default creates stacks names that does not match JOB_NAME, therefore ignore them while processing orphaned stacks and handle them separatly when cleaning up the orphaned clusters.

  The stack naming scheme is limited to take first 20 chars from the JOB_NAME while the rest is randomly generated for uniqueness: https://github.com/openstack/magnum/blob/master/magnum/drivers/heat/driver.py#L202-L212 This breaks the openstack cron jobs.

1.1.53 v0.73.0

Prelude

Enable support for OpenJDK17.

New Features

- Add support for the latest version of JDK17 to be used with lf-update-java-alternatives. This allows job to switch between the required version of JDK.

1.1.54 v0.72.0

New Features

- Add gerrit-go-verify job that will support running unit test for Go projects. It will run the command 'go test ./...' inside the indicated GO_ROOT path.

1.1.55 v0.71.0

New Features

- Add support for wait flag on SonarCloud quality gates, this way jobs won't finish until the quality gate reports back the result during the analysis step, which will fail anytime the quality gate fails.

1.1.56 v0.70.0

Bug Fixes

- gerrit-maven-sonar-verify was using the "default" branch choosing strategy, which pulls master, rather than the "gerrit" strategy for pulling feature branches.

1.1.57 v0.69.1

Prelude

Improve global-jjb code and documentation to minimize non-inclusivity. https://developers.google.com/style/inclusive-documentation

New Features

- Add packer builder macros to create a 'ansible.cfg' file. This is required by packer build jobs to set ansible host configuration. The job or image specific configuration can be created using JCasC custom files in the ci-man repository.

  Example configuration: https://github.com/ansible/ansible/blob/stable-2.11/examples/ansible.cfg

Upgrade Notes

- This change requires creation of a custom managed file (ansible.cfg) on the Jenkins environment with the default line "remote_src = ~/.ansible/tmp".

Bug Fixes

- Fix shell script maven-sonar.sh to handle parameter $SCAN_DEV_BRANCH as string and not as boolean, so we can achieve the desired IF logic

Other Notes

- Rename 'whitelist' to 'allowlist'

1.1.58 v0.69.0

New Features

- Add new Maven SonarCloud verify job that will execute SonarCloud scans before a change gets merged.

1.1.59 v0.68.1

Bug Fixes

- The OpenStack JCasC yaml converter has learned how to properly differentiate between volumeFromImage and image boot sources.
- Pin pyparsing<3.0.0 which is required by httplib2 0.20.1. A new version of pip 21.3.1 is out that has removed the dependency pyparsing<3,>=2.4.2 as required by httplib2.

1.1.60 v0.68.0

Prelude

JDK8 is soon going to be EOL by Mar 31, 2022. with most of the LF projects already using JDK11 (LTS), upgrade default version to JDK11 (LTS).

Upgrade Notes

- Upgrade default version to JDK11 (LTS).

  https://www.oracle.com/java/technologies/java-se-support-roadmap.html

Bug Fixes

- The JCasC convert for OpenStack was improperly converting executor definitions. The script has learned the proper syntax.

1.1.61 v0.67.1

New Features

- Add `sonarcloud-java-version` parameter to LF Sonar builders, which allows setting the JDK version to use with the Sonar scanner (default: openjdk11).

1.1.62 v0.67.0

New Features

- Add "Unmaintained" as a valid lifecycle_state to be used in INFO.yaml files.

1.1.63 v0.66.1

Bug Fixes

- Add missing v3-standard Vexxhost flavors to create_jenkins_clouds_openstack_yaml.

1.1.64 v0.66.0

New Features

- Add v3-starter flavors to cloud lookup in create_jenkins_clouds_openstack_yaml script.
- Add support in lf-maven-sonar scan jobs to process short lived dev branches.

1.1.65 v0.65.3

Bug Fixes

- Openstack labels need to include the config name, in addition to any labels explicitly defined. This also changes the builder name to match the config name, rather than using the labels (which can be only one label, but is technically a space-separated list).
- If no volume_size is defined, the default behavior was to set one to 10GB. However, the proper way to handle this is to use an "Image" boot source rather than "Volume From Image".

1.1.66 v0.65.2

Bug Fixes

- When running create_jenkins_clouds_openstack_yaml.py, rather than quitting if there are no labels defined, we can instead use the agent name (e.g. "centos7-2c-1g") as the default label. This recreates the functionality of the groovy scripts previously used.

1.1.67 v0.65.0

New Features

- Parallel jobs are now natively supported by tox since version 3.7.0 thanks to the option "-p" / "–parallel". This new option offers more possibilities than detox and other options have also been introduced to tune tox behavior in parallel mode. This evolution allows more tox parallel mode configurations in yaml templates.

1.1.68 v0.64.0

New Features

- Create a bashate tox profile. Bashate is a shell linter inspired from PEP8. It is now enforced by CI to improve bash code style.
- Update to the sysstat script to add support for operating system Ubuntu 20.04 and Docker systems

Bug Fixes

- Fix the issues detected by bashate (E043, E010, E011, E002, E020, E006, E003).

1.1.69 v0.63.1

Bug Fixes

- Set S3 URL in the framename of the target attribute

  The description-setter plugin does not read the URL when the framename is unset.

  Issue: LF JIRA RELENG-3269

Other Notes

- Conventional Commit message subject lines are now enforced. This affects CI. Additionally, if developers want to protect themselves from CI failing on this please make sure of the following
  - you have pre-commit installed
  - that you have run pre-commit install –hook-type commit-msg
- yamllint is now enforced on commits via pre-commit. This affects both CI as well as developers that have pre-commit properly configured in their environment.

1.1.70 v0.63.0

Bug Fixes

- Provision global-settings to replace the default used by the Unified Agent.
- The choosing strategy for the docker-merge-{stream} jobs for gerrit was improperly configured to use. This is now corrected.

1.1.71 v0.62.0

New Features

- Add sonar-prescan-script jobs for maven, allowing maven sonar jobs to execute a shell script prior to the scan.
- Add artifact distribution type for self releases. This support allows customers to specify a particular name and path of an artifact in Nexus which will be downloaded locally, re-tagged and posted into the releases repository of the matching repo.

Bug Fixes

- Pin git review to 1.78

  The latest version of module trys to look for git hook recursively within the submodules.

  Error:

```
Running: git submodule foreach cp -p .git/hooks/commit-msg "$(git rev-parse --git-
↪dir)/hooks/"
Problems encountered installing commit-msg hook
The following command failed with exit code 128
     "git submodule foreach cp -p .git/hooks/commit-msg "$(git rev-parse --git-dir)/
↪hooks/""
-----------------------
Entering 'global-jjb'
cannot stat '.git/hooks/commit-msg': Not a directory
fatal: run_command returned non-zero status for global-jjb
```

  Remove workaround that has been resolved in v1.28 and use lf-activate-venv to install git-review

- Reorder functions and add function labels to make release-job.sh easier to read.
- Update the create script to include V3 flavors. Newer V3 flavors are faster and garuntees the to run on new hardware.

1.1.72 v0.61.1

Bug Fixes

- Pin cryptography to 3.3.2

  The latest version of module breaks compatibility with the latest version of pip.

  Error:

```
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/tmp/pip-build-vqk6fya9/cryptography/setup.py", line 14,
  in <module> from setuptools_rust import RustExtension
ModuleNotFoundError: No module named 'setuptools_rust'
```

  Reference:

  PYCA#5753 <https://github.com/pyca/cryptography/issues/5753> PYCA#5771 <https://github.com/pyca/cryptography/issues/5771>

1.1.73 v0.61.0

New Features

- Enable support for artifact type releases. Add initial schema for artifact release verification.

Upgrade Notes

- Upgrade Packer version to v1.6.6. v1.6.6 gives more debug messages which is not seen with 1.4.2.

Bug Fixes

- Add self release verify and merge jobs for GitHub based projects.
- Fixes an bug with 'job-cost.sh' that would cause builds to be marked as unstable if not run on AWS or OpenStack.
- The old version of rtd-verify is missing the "–init" flag, which causes it to not add new submodules. Additionally, the "–recursive" flag has been included to ensure proper loading of recursive submodules.
- Updates the 'capture-instance-metadata.sh' script to skip attempting to capture instance metadata needed for job-cost.sh if the build is being run on an unsupported cloud or platform.
- Updates the 'sudo-logs.sh' script to set ownerhsip to current build user and user's login group, instead of the explicit 'jenkins:jenkins'. This will allow sudoer log ownership to work on builders not using 'jenkins' as their build username.

1.1.74 v0.60.5

Bug Fixes

- Add regex to trigger packer jobs when common-packer templates are updated.
- Fix the release job script to handle LOG_DIR unbound variable and condition to check if the LOGS_SERVER or CDN_URL is being used.

1.1.75 v0.60.4

Bug Fixes

- Remove python 2.7 support

  As per the deprecation notice python 2.7 is not long supported. This causing job failures since the dependencies install are not maintained.

  DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality.

1.1.76 v0.60.3

Bug Fixes

- Rename "tag-gerrit-repo" function in self releases to "tag-git-repo". This change is in preparation for self releases support for GitHub based projects. Use the name "git-repo" to fit both Gerrit and Github projects.

1.1.77 v0.60.0

New Features

- Add pipeline-verify jobs. This adds a simple pipeline verification job that will lint any pipelines found, and check to make sure that they are not set up to run on master.

Bug Fixes

- Fix release merge jobs to work with AWS S3_BUCKET when CDN_URL is set.

  Log shipping is being migrated from Nexus2 Log to AWS S3 Buckets. This requires the release job scripts to handle CDN_URL and set the correct value for LOG_SERVER.

1.1.78 v0.59.0

Bug Fixes

- Activate lftools env before processing the logs-deploy script. This will allow us to fetch the latest version of pyOpenSSL and cryptography to resolve the following issue https://github.com/pyca/pyopenssl/issues/973

1.1.79 v0.58.1

Bug Fixes

- Pin idna to 2.9 to resolve package incompatibility with lftools 0.35.1.

1.1.80 v0.58.0

New Features

- Adds a create_jenkins_managed_files_yaml.py administrative script for generating JCasC yaml files from a directory for control of the Managed Configuration Files plugin

Bug Fixes

- Allow projects to override Semantic Version (SemVer).

  It's recommended to use Semantic Versions (SemVer) for releases. Refer to https://semver.org for more details on SemVer. For projects that do not follow SemVer can use a job build parameter (OVERRIDE_SEMVER_REGEX) with the release job. This build param overrides the default SemVer regex.

  The default SemVer regex is taken from https://github.com/fsaintjacques/semver-tool.

1.1.81 v0.57.4

Deprecation Notes

- Cmake builds upload to SonarCloud. Fix build env to use openjdk11 sonarcloud upload will stop working with java8 on October 1st 2020

Bug Fixes

- Fix the capture-instance-metadata script to cleanly exit on AWS nodes.
- When evaluating jenkins-config management changes, if a system does not have an openstack cloud defined, we should not fail the job. Instead we now just skip that configuration and verification
- openstack-protect-in-use-images.sh Projects that do not have images in jjb were failing this step of the build. code now runs with set -eu -o pipefail for the duration of the script. shellcheck disable comments no longer needed and removed. Code now only merges arrays if non-empty. Simplify dedupe array code.

- Fix rtd-verify.sh to better validate submodules. rtd-verify.sh was using "git submodule" commands to validate submodules, but Jenkins reads the .gitmodules file and executes a "git config" command for each submodule. Because of this, if a bad submodule was added to .gitmodules, it would pass verify but cause failures on subsequent builds after it was merged.

  This change closes that loophole by reading the .gitmodules file, and then running through the same "git config" command for each submodule that Jenkins runs when pulling in the main branch.

1.1.82 v0.57.3

Prelude

Provides method to notify administrators if important jobs are failing.

New Features

- Allows customization of email address and email prefix. failure-notification-prefix failure-notification

Upgrade Notes

- Updating to this version of JJB will _require_ updating jjb/defaults.yaml in the same patchset. The following will need to be defined: failure-notification: "foo@bar.org" failure-notification-prefix: "[Some Prefix]"

Bug Fixes

- Improve alpine compatibility by changing the mktemp call in python-tools-install.sh to be compatible with both GNU mktemp as well as BusyBox mktemp (which is used in alpine).

1.1.83 v0.57.1

Bug Fixes

- Check condition when ${NOMAD_DC} is undefined or unset

  Check if the environment variable $NOMAD_DC is not defined or unset, this avoids the script from exiting without capturing instance metadata.

1.1.84 v0.57.0

New Features

- Packer CI jobs now have the ability to specify which "builder" to use in the job. Default configuration is the for the "openstack" builder.

Upgrade Notes

- Requires common-packer v0.7.0 if using packer CI jobs.
- If your project is in AWS and uses the Packer CI jobs to build AMIs you must set `packer-builder: aws` when upgrading to this version of global-jjb or your jobs will default to openstack and fail.

Bug Fixes

- Optionally get docker container tag to be set to the value of *stream*.
- Fix email notifications for lfdocs-conf releases. The name of the repo does not match the configuration in the code. Add a new PROJECT_SLUG variable to use in case the name of the repo is different from its configuration.
- Parallel mode for tox environments is broken if the user passes a JJB bool value to the job-template. We now lowercase the PARALLEL variable when comparing in the bash script to ensure the user provided value is compared against the lowercase value.
- Run Sonar scans using JDK11. *java-version* will only set java for the maven build part of the job, the sonar scan will use *sonarcloud-java-version* which is set to *openjdk11*. Projects not compatible with JDK11 will be able to run their build with *java-version* set to their JDK preference. https://sonarcloud.io/documentation/user-guide/move-analysis-java-11/

1.1.85 v0.56.0

New Features

- Capture instance metadata for the VM's. This data is useful while working with the cloud provider and debuging VM issues on the infrastructure.

Bug Fixes

- Fix image update script to handle $image_type correctly.

  This fixes the issue when a specific image type overwrites other image types incorrectly in the .cfg and some images names getting excluded.
- Minor fixes to correct shellcheck warnings/errors.
- Retains the expected behaviour of Gerrit Trigger job configuration for the comment-event-parameter-mode when a project upgrades their JJB to 3.5.0 or newer.

  In JJB 3.5.0 support was added to configure the comment-event-parameter-mode however, while it's default mode in JJB matches the documented help text for the configuration in Jenkins the Gerrit Trigger plugin itself appears to default to PLAIN mode when the parameter is excluded. This patch retains what we expect to be the default behaviour.
- Update sigul-install.sh to check for sigul. If sigul is already installed, we can skip the installation. This is important due to kojipkgs being unreliable. We now have the sigul binary baked into the base image creation, so newer images should already have sigul on board. If they do and we still try to do this manual installation, we could still run into connection issues with kojipkgs.

1.1.86 v0.55.3

Bug Fixes

- Fix the race condition by checking the created_at timestamp and clean up ports that were created at least 30 minutes before.

  There a race condition in the openstack-cron job that causes the script to delete ports in DOWN state and are still in use by the VM, causing the ODL CSIT jobs to fail.

  JSD ISSUE: RELENG-3062

1.1.87 v0.55.2

New Features

- Introduce lf-pipelines-verify job to test the LF's global pipeline library.

Deprecation Notes

- lftools_activate should no longer be used and will be removed in a future release.

Bug Fixes

- Fix the jjb-deploy job to accept a JJB_VERSION parameter in the parameters field so that projects can select which version of JJB they want for the job.

  The change that ended up breaking jjb-deploy was caused by this Gerrit patch here:

  https://gerrit.linuxfoundation.org/infra/c/releng/global-jjb/+/62788

  The bug was introduced in the lf-env.sh when the lf-pip-install macro was removed. Prior to the removal the jjb-deploy job was installing the latest and greatest JJB version. However now jjb-deploy is now installing the fallback version of JJB_VERSION from the lf-env.sh script which is 2.8.0.

  Reference: RELENG-3073

- Fix quoting bug in rtdv3.sh script that caused failures to exit improperly.
- Resolves the lftools_activate failure below.

```
---> maven-deploy.sh
ImportError: cannot import name 'enquote_executable'
```

  Fix is to replace lftools_activate with lf-activate-venv by sourcing lf-env.sh lf-activate-venv is a more error resistant way to make sure a pip package is installed

- Release verify will not allow valid semver versions like "1.3.2-1" which should be accepted. Update semver regex to conform to https://semver.org/ This regex was taken from https://github.com/fsaintjacques/semver-tool which follows the semver guidances and works in bash.

1.1.88 v0.55.1

Bug Fixes

- Pin importlib-resources to 2.0.0 to resolve package incompatibility between it and the currnet virtualenv version (20.0.21). Until a compatibile virtualenv is released this dependency needs to be pinned.
- Fixed logic for discovering new branches When a new branch exists on gerrit, but has never been seen by RTD we need to trigger a job so that the branch is discovered we can then mark it active in the following step. API changed under out feet, where a 404 was returned before we now get a 200 with the string null. Change code to explicitly match the returned string of "null"

1.1.89 v0.55.0

New Features

- verify-upstream-global-jjb jobs have been improved in two ways: the upstream Gerrit name has been changed to lf-releng to match the precedent set by OpenDaylight, and we have added a Github version of the job.

Bug Fixes

- Fix Job key in commit message body rather than trailers section (AKA footer). The commit message produced by gerrit-push-patch which currently creates a commit message where the Job key appears in the commit message body rather than the trailer.

  For example:

```
An example commit message

Job: builder-job/123

Change-Id: 1234567
Signed-off-by: Jenkins <jenkins@example.org>
```

  This fixes it to:

```
An example commit with proper trailer

Job: builder-job/123
Change-Id: 123457
Signed-off-by: Jenkins <jenkins@example.org>
```

- The packer merge job has a boolean option that sets an UPDATE_CLOUD_IMAGE variable. This variable was always evaluating to true which caused issues with builds always executing the section of the build. This causes problems for builds that do not produce information that is expected by the section of code. In specific AWS / EC2 builds fail as the build engine outputs different name data than expected. The variable is now properly evaluated.

1.1.90 v0.54.0

New Features

- Add new templates to build CXX projects with GNU autotools. These support the "configure && make && make install" pattern used by many open-source projects. Supports generation of the configure script for projects that do not store generated files in version control. Includes gerrit and github versions of autotools-packagecloud-stage, autotools-sonarqube and autotools-verify.

Bug Fixes

- Echo error response from RTD without formatting. The script rtd-trigger-build.sh starts a build at ReadTheDocs and parses the response, a small JSON like this:

  {"build_triggered":false,"project":"unicorn","versions":[]}

  This change drops the invocation of jq that attempts to pretty-print the JSON to the log when the build is not triggered, because that call was failing for inexplicable reasons.

- Extend scripts that invoke the docker CLI to report the version by invoking *docker –version*. Also add echo command to report end of script where it was missing. Includes docker-build.sh, docker-login.sh, docker-push.sh and release-job.sh. No functional change to any script behavior, just a line of extra output.

1.1.91 v0.53.2

Known Issues

- Update regexp in lf-infra-ship-logs macro. This fixes a regression from update to deploy-logs.sh script in v0.53.0 which causes build logs links to disappear on the Jenkins jobs description setter.

  Tests: https://jenkins.opendaylight.org/releng/view/builder/job/builder-packer-verify/274

1.1.92 v0.53.1

Upgrade Notes

- Before upgrading to this version of global-jjb, must remove any uses of the job templates named above. These jobs did not yield any results, so it's unlikely they were being used, and dropping them entirely should not cause any loss of information.

Bug Fixes

- Fix update cloud image list job to handle newlines correctly and update an existing gerrit change request.

  A gerrit change submitted through git-review checks for an existing change-id on Gerrit and either updates the patchset or creates a new one. For this to work correctly the change-id should go into the commit footer and not in the GERRIT_COMMIT_MESSAGE. The embedded newlines are not processed correctly in the script and fails to separate the footer and the GERRIT_COMMIT_MESSAGE, instead it creates a new patches everytime rather than updating an existing one.

  The change is tested now existing CR's are updated rather than pushing a new CR everytime.

  https://git.opendaylight.org/gerrit/c/releng/builder/+/89136

- Drop jobs gerrit-python-xc-clm and github-python-xc-clm from the two python job groups. Those templates were replaced by gerrit-tox-nexus-iq-clm and github-tox-nexus-iq-clm. Not adding those templates to the job groups because successful use requires additional project configuration to report the package requirements.

1.1.93 v0.53.0

New Features

- Add "logs" prefix to s3_path("logs/$SILO/$JENKINS_HOSTNAME/$JOB_NAME/ $BUILD_NUMBER/"). We are not allowed to have an index.html file at the root level of the S3 bucket. Adding the additional prefix allows us to mirror the directory structure of Nexus where you are able to see both production and sandbox logs when browsing to https://logs.<project>.org/.
- Add templates gerrit-pypi-stage/github-pypi-stage to allow projects flexibility to control when Jenkins publishes a redistributable Python package, either on merge (with the merge template) or on command (with the stage template).
- New templates gerrit-cmake-packagecloud-stage and github-cmake-packagecloud-stage for building DEB/RPM package files and publishing them to a PackageCloud.io repository on posted comment. Adds builder macros lf-packagecloud-file-provider and lf-packagecloud-push. Adds script packagecloud-push.sh to call Ruby gem package_cloud. The new templates are lifted from the ORAN project.
- Verify build nodes named in YAML files against config files. Extend lf-jjb-verify anchor with boolean configuration variable check-build-node-labels that guards a conditional build step. If true, run script jjb-verify-build-nodes.sh to check build-node labels in YAML files within the jjb subdirectory against labels defined by config files in the jenkins cloud configuration directory. Disabled by default. Projects should enable and configure the job; e.g., for external build node labels.
- Add a packer-verify-build job. This job is made to be manually triggered in order to build the packer image, so that the full build process can be validated before merging. If this is done on the final patch that is merged, the merge job will not run another build (thereby avoiding building two identical images).
- Optionally tag repo during release process. Set to true by default. Allow projects to skip repo tag in cases where multiple release types happen whithin the same repo.

Bug Fixes

- Standardize yaml parsing tools used across global-jjb. Switch from using niet package to yq for parsing yaml files in release-job.sh.
- Fix issue where job fails if the global variable S3_BUCKET is not set. Set conditional parameter "on-evaluation-failure" to "dont-run".
- Schema type "bool" is not valid. Fix to "boolean"
- Fix script to remove outputting $ARCHIVE_ARTIFACTS. Outputting required that $ARCHIVE_ARTIFACTS be set for every job. Instead it will output ${pattern_opts:-}, which if $ARCHIVE_ARTIFACTS is not set, will be blank.
- Fix the rtd verify script failure. The script attempts to install lftools dev with –user and fails on the error.

  Error:

  > [Errno 13] Permission denied: '/usr/local/lib/python3.6'

  This code is actually required when a new commands are added to lftools, the docs verify job needs to run the new command and install dev version lftools version.

- Document the jjb-verify feature that checks build-node labels named in JJB YAML files against nodes in cloud config files. Change the feature's JJB configuration variables to share a prefix. Improve the shell script to be robust to errors such as a suffix of ".cfg" or an external label of "" (just two double quotes).
- Add @weekly cron default value to docker_merge_common anchor, no longer empty. The merge job for docker images is like the stage job for java artifacts, every docker image is a release candidate. Run the merge job regularly to check dependencies like base images and to push updated images to the Nexus3 staging registry.
- Allow rtdv3 jobs to run in parallel.
- Revise tox-run.sh to guard against glob matching no tox log files. In that case the pattern is passed to the cp command, which fails. Detect the cp failure ('cp: cannot stat ..') and exit the loop. This new and undesired behavior was introduced by adding -e in change 76a0761, so the script stops when the cp command fails.

1.1.94 v0.52.0

New Features

- New templates gerrit-cmake-sonarqube and github-cmake-sonarqube use the SonarQube Jenkins plug-in to analyze CXX code and publish the results. Modeled after the generic gerrit/github-sonar templates. The new templates lift the limitations of the existing templates gerrit/github-cmake-sonar which download and install a Sonar scanner, and cannot report unit-test code coverage statistics.
- New templates gerrit-tox-sonarqube and github-tox-sonarqube use the SonarQube Jenkins plug-in to analyze Python code and publish the results. Modeled after the generic gerrit/github-sonar templates. The new templates eliminate the need for mvn-settings in the job configuration and the need for a pom.xml file in the repo.

Bug Fixes

- Add missing config property "stream: master" to lf_cmake_sonar macro in lf-c-cpp-jobs.yaml to match existing "branch: master" property. This makes the template consistent with other Sonar templates.

  Remove double-quote chars that were added recently around $make_opts from the invocation of the build wrapper in cmake-sonar.sh. This makes the invocation consistent with the invocation of cmake in the same file.

  Update documentation to reflect correct property names and defaults.
- Workarounds for aging python 3.5 on Ubuntu 16.04 builders Ubuntu 16.04 calls a python2 version of tox we must upgrade it even if we are testing with python3 ubuntu 16.04 runs python 3.5 we must pin the zipp package more pins will be needed as packages drop support for python 3.5 ideally projects will move away from this distro.
- Update jq validation of returned json blocks to work properly with jq v1.5.
- Switch info-file-validate.sh to use lf-activate-venv moves job from python2 to python3
- Revise script nexus-iq-cli.sh to stop on error or unbound variable. This should fail the build if the scanner returns a non-zero code, for example if credentials are missing or wrong.
- Refactor templates in lf-c-cpp-jobs.yaml with common anchor/alias lf_cmake_common to reduce redundant configuration. This includes gerrit-cmake-sonar, github-cmake-sonar, gerrit-cmake-sonarqube, github-cmake-sonarqube, gerrit-cmake-stage, github-cmake-stage, gerrit-cmake-verify and github-cmake-verify. No functional change.

  Document maven settings parameters in g*t-cmake-stage templates.
- Revise templates gerrit-cmake-sonarqube and gerrit-tox-sonarqube to move the triggering comment string into a parameter that can be overridden in a job definition. Github comment remains hardcoded. The default is still "run-sonar".

Rename gerrit-cmake-sonarqube tempplate configuration parameter from sonar-prescan-script to pre-build to be consistent with existing CMake stage and verify templates. This is a breaking change for any early adopters of this new template.

- Reduce volume of output from wget and unzip by adding the -q flag to the invocations in cmake-sonar.sh There's little need to see wget progress messages or archive contents in the build log.
- Revise tox-run.sh to guard against unbound variable TOX_ENVS; stop on error or unbound variable (options -eu); and to print commands before executing (option -x). Add echo command at end.

1.1.95 v0.51.0

New Features

- New script has been added: scrape-job-cost. This script will be executed by cron as nexus on the Nexus Server for each project. It will extract cost data from the nexus directory for each Jenkins Builder (production & sandbox). The cost data for each builder will be appended to separate cost files located in ~nexus/cost on the nexus server. The files will be named sandbox-YYYY.csv and production-YYYY.csv.
- New templates gerrit-sonar-prescan-script and github-sonar-prescan-script accept an arbitrary shell-script body that can do work like install prerequisites, build and test to generate a code-coverage report for the Sonar Scanner to find and upload. This adds flexibility that the existing gerrit-sonar-prescan and github-sonar-prescan templates lack.

Known Issues

- Removes ref-update from rtd-merge jobs which is triggering unnecessary jobs to be queued in Jenkins. This ref-update was originally added to enable Jenkins to trigger builds when a release tag is pushed to update the docs however it's now triggering many unnecessary jobs wasting project CI resources.

Bug Fixes

- Removed broken code that removed leading/trailing white-space from variables. Use lf-activate-venv() to install openstack. Enabled 'set -euf pipefail' and updated code to handle errors. Updated out of date comments. Some minor cleanup of code for clarity.
- Call "lftools jenkins" after credentials are set to fix failures due to the call being made without credentials being set first. The previous method did not require credentials, so the failure was introduced when we switched to using lftools. The os_plugin_version variable is not needed before the JENKINS_USER and JENKINS_PASSWORD are set, so no other changes are necessary.
- A recent change has made the "null" string a bad value for FLOATING_IP_POOL. By making it an empty string, we recreate the old functionality of having the default floating IP pool set to "No value".
- Fix release file detection on commit with multiple parents
- Branch discovery and build polling implemented. If a Branch has not been seen by rtd we trigger a build with rtd and poll till that build is complete. we can then enable the branch and trigger a build against it, again polling all builds untill they are complete.
- Fix to disable cloning submodules for rtdv3 verify job
- Add configurable doc-dir defaults to "docs/_build/html" needed for relative path modifications if you change the tox-dir Modified tox-run.sh with "ARCHIVE_DOC_DIR" variable so that relative paths can be handed when uploading generated docs to the log server

- Use git choosing strategy default in tox and pypi merge jobs for gerrit. This makes those jobs consistent with maven and other merge jobs for gerrit that always build from tip of the target branch to create artifacts from the latest & greatest code. Building from tip (not from Gerrit commit/merge point) avoids confusion about content when changes are merged out of order. For example, a fix is submitted and merged, but the merge job fails. In the mean time, a different change that happened earlier in commit history gets merged (or the merge job is retriggered), causing a new artifact to be pushed. But that artifact does not have the expected fix.

  Add comments to release merge jobs why their choosing strategy is not default.

  Document the git commit choosing strategy for the release merge jobs.

- Extend release-job.sh to detect if distribution_type is missing from the release yaml file and show a meaningful error. The shell option pipefile causes the script to halt silently if niet fails to find that key, which utterly baffles users.

- Use choosing strategy Gerrit Trigger in container/jar and package cloud release merge jobs. This retains the current behavior in the simple merge case, and ensures that a job triggered by a "remerge" comment uses the release file at that commit. The previous choosing strategy, default, uses the tip of the target branch. That does not allow recovery from merge job failure if the target branch has advanced past the commit with the release file.

1.1.96 v0.50.0

New Features

- Generate javadoc from a project in a subdirectory, which lifts the assumption that all files are in the git repository root. Extend maven-javadoc publish and javadoc-verify templates with mvn-dir configuration parameter (like tox-dir), defaults to '.' to keep existing behavior. Extend maven-javadoc-generate.sh script to invoke mvn with the -f option and use the directory name when archiving the generated HTML. Document the new configuration parameter.

Upgrade Notes

- Projects using macros lf-maven-javadoc-publish or lf-maven-javadoc-verify (i.e., using job templates gerrit-maven-javadoc-publish, github-maven-javadoc-publish, gerrit-maven-javadoc-verify or github-maven-javadoc-verify) must ensure the maven -f option is not used in config parameter mvn-params.

- The lf-pip-install builder macro has been deleted. At some point we will also be deleting shell/pip-install.sh.

Bug Fixes

- Fix zgrep pattern for verify_packagecloud_match_release function. Function was looking for "Successfully Uploaded" pattern, which is not the correct response when using packagecloud gem to push packages to packagecloud.

  Fix lf-release docs example to reflect changes to package_name. Package_name should not include the version in the name.

- Disable the pip install warning that are not useful in the console logs.

  00:25:10 WARNING: The scripts easy_install and easy_install-3.6 are installed in '/home/jenkins/.local/bin' which is not on PATH.

  00:25:10 Consider adding this directory to PATH or, if you prefer to suppress this warning, use –no-warn-script-location.

- Revise shell script maven-javadoc-generate.sh to supply -f argument to maven with argument translated by call to readlink. This works around the javadoc:aggregate behavior of silently doing nothing if invoked "-f ."

- Extend shell scripts that invoke pip freeze to show python and pip versions also
- Revise python-tools-install.sh to drop creation of virtualenv in ~/.local. That is NOT a Python virtualenv and should not be created as such. Using *–user* installs python modules into ~/.local/lib/PYTHON_VERSION/site-packages. Making ~/.local a virtualenv messes with the paths in site-packages and causes runtime errors like this:

  ERROR: Can not perform a '–user' install. User site-packages are not visible in this virtualenv.

  Reverts part of change I4b2d778f3fd81565c5dd009d50c969696faba0d2
- Extend script python-tools-install.sh to invoke pip freeze. This shows installed package versions in the log and allows detecting changes like the Jan 2020 release of pip v20 that broke various assumptions and behaviors.
- Extend macro lf_tox_sonar with parameters macro lf-infra-maven-parameters so job definitions can configure mvn-opts, mvn-params and mvn-version values. Default values are defined so existing jobs are not affected. This change makes the python-tox sonar template consistent with the maven sonar template.

  Extend the PyPI package structure recommendation with a way to share the docs/ folder.
- The update-cloud-images.sh script was using virtualenv to install openstack which fails because the latest version of openstack requires python3. It fails quietly because that is no error handling in the script (all errors are ignored). The script now uses lf-activate-venv() and it runs with 'set -euo pipefail'.
- The jjb-deploy-job.sh was 'activating' a venv that was created by pip-install.sh in /tmp/v. This venv was based on python2. Now jjb-deploy-job.sh calls lf-activate-venv() to install jenkins-jobs in a venv based on python3.

1.1.97 v0.49.0

Prelude

This release supports breaking changes in the upstream OpenStack Jenkins plugin. Version 2.47+ of the OpenStack Cloud plugin for Jenkins adds two new SlaveOptions params, node_properties and config_drive. Failure to send these params results in a failed request.

A project may need more than one type of release job this patch changes the trigger to match on release file name for pypi and packagecloud release jobs.

New Features

- Extend tag feature of release jobs: Extend release-yaml schema files to allow git_tag entry. Extend lf-build-with-parameters-maven-release macro with GIT_TAG. Extend release-job.sh: detect and use optional git_tag string. detect and reject an existing lightweight tag that blocks push of a gpg-signed tag. change GERRIT_HOST to GERRIT_URL in method for obtaining LF umbrella project to allow testing in the sandbox. rename function from tag to tag-gerrit-repo. simplify tests for presence of Jenkins parameter values show more details about steps as INFO statements Include schema file contents into documentation; remove the copies.
- Add support for automation of promoting packagecloud packages.
- Add gerrit-branch-lock to gerrit ci-jobs, and make it so that it can be triggered from any change, for any branch.
- Generate Job cost information Each job it will archive a CSV file (cost.csv). It will contain a single CSV record containing the following fields: JobName , BuildNumber , Date , InstanceType , Uptime , Cost1 , Cost2 The Date field can be sorted as a string and is readable by your favorite spreadsheet. The Date/Time is GMT. The Uptime is uptime of the build agent (secs). The Cost1 field is the cost($$) of the build node & Cost2 is cost associated with the stack. If the job is not a Openstack job, then Cost2 is '0'. The project cost file will based on the year (cost-2019.csv).

- Add support for these new parameters, node_properties and config_drive. Add a new function test_version, which will do a numerical (float) comparison between installed plugin versions and user-supplied test version numbers.
- Add the lf-kubernetes-create and lf-kubernetes delete macros which allow ad-hoc kubernetes cluster creation with a selectable disk size for the docker volume.
- Make tox-dir configurable default is "." can now be set to "docs/"
- Project can now set their default docs landing page used for self serve release of docs. By default landing page is /latest/ on release landing page should generally be set to /stable/
- Release jobs now support non-ff merges.
- **Extend packagecloud release:** Release additionally requires log directory, ref and version. Add set_variables_packagecloud & verify_packagecloud_match_release to allow tagging the git repo. Update packagecloud jobs to include sigul signing.

Known Issues

- Release jobs still trigger when a release file is deleted.

Upgrade Notes

- The openstack-stack-delete.sh script installs the latest tagged version of lftools and it uses that to get the stack cost. Any version of lftools >= v0.29.0 will contain the required changes to get the stack cost.
- If you are using the INSTANCE_MIN_CAPMAX paramater in your cloud configs, you will need to change it to INSTANCE_MIN when using v2.47+ of the plugin.
- default-version is now an optional parameter 'latest' is the default behavior change to 'stable' when docs are released

Deprecation Notes

- Deprecated the config option INSTANCE_MIN_CAPMAX, replaced with INSTANCE_MIN when using v2.47+ of the OpenStack plugin.

Bug Fixes

- Currently a Job Config Tarball is added to the archive. The directory layout created by 'jenkins-jobs test' has changed and as a result the code that sorted/reformated the directory and generated the tarball was broken.

  The prefix path of the tarball is now 'job-configs'. The number of log messages has been improved and greatly reduced. The tarball generated contains directory names with spaces and upper-case characters.
- Revise the lf-docker-get-container-tag macro to use include-raw, not include-raw-escape, to silence JJB warning. Improve documentation of lf-docker-get-container-tag macro.
- Removed dependency on 'user' venv created by python-tools-install.sh from the sandbox cleanup job. It will call lf-acitivate-venv() instead. Removed references to jjb-install.sh & jjb-cleanup.sh.
- Removed dependency on 'user' venv created by python-tools-install.sh from the builder-jjb-verify job. It will call lf-acitivate-venv() instead.
- Make shell script builder step customizable in lf-docker-get-container-tag. This will allow ONAP to call their own script. Set to default to "../shell/docker-get-container-tag.sh" which is what teams were already using.

Other Notes

- Disable generation of pip package lists during builds: packages_end.txt.gz, packages_start.txt.gz & packages_diff.txt.gz. They are no longer valid.

1.1.98 v0.48.1

Bug Fixes

- Change multi-cloud image validation to look for an OS_CLOUD variable in the cloud.cfg instead of overloading CLOUD_CREDENTIAL_ID as that variable does not usually point to an ID that is in the openstack clouds.yaml file.

1.1.99 v0.48.0

New Features

- By default, the lf-venv-create() function only installs/upgrades 'pip' and and any packages that need to be tied to a specific version like. Currently the only version specific package is: 'jenkins-job-builder==2.8.0'. The current version of lf-venv-create() supports being called twice but that functionality has NOT yet been thoroughly tested. RELENG-2508 has been created to validate & optimize that functionality.
- Created new function lf-activate-venv(). This function creates a venv in /tmp and prepends the bin directory to the PATH. The 'pip install' command now specifies: '–upgrade-strategy eager'. lf-activate-venv() supports an optional –python flag to specify which python to use to create the venv, the default is python3.

  Two new functions: lf-git-validate-jira-urls() and lf-jjb-check-ascii(). They will be used to replace the git-validate-jira-urls.sh & jjb-check-unicode.sh scripts at some point. For now, they are not being used.

Bug Fixes

- Update the gerrit trigger regexes for the docker job templates to use the shorter and more readable versions of the regexes used in other verify and merge job templates. Clarify documentation for comment triggers.
- Multi-cloud image validation was not properly working as we force set the OS_CLOUD environment variable before validation of images. This has been rectified to dynamicaly modify the OS_CLOUD variable based upon the cloud.cfg that will be used to define the cloud in Jenkins.
- We now validate that a cloud.cfg file exists for any defined OpenStack cloud.
- Removes undocumented, and now unneeded, openstack-cloud variable from the jenkins-cfg-verify job definition
- Change compare-type to REG_EXP in macros lf_python_clm_xc and lf_pypi_common for config parameter gerrit_trigger_file_paths with the regular expression .* Previously was ANT, which didn't match anything with pattern .* so recheck/reverify comment triggers did not work. Clarify documentation of comment triggers for python jobs.
- Add .pypirc config file provider to builders in release job. Extend PyPI documentation with recommended directory layout.
- Change the comment-added-contains-event strings for the release-verify and release-merge templates to the same regexes used in other verify and merge templates, instead of custom versions that didn't seem to work; verify now uses '^Patch Sets+d+:s+(recheck|reverify)s*$' merge now uses '^Patch Sets+d+:s+remerges*$'

- Change pip's upgrade-strategy to "eager" for python-tools-install.sh. Pip changed its default upgrade-strategy to "only-if-required", which is not correctly upgrading requests (and potentially other packages in the future) to meet other packages' requirements. This results in errors in the build log. By using the upgrade-strategy "eager", pip is able to properly install what is needed.

Other Notes

- The lf-venv-add(), lf-venv-create() & lf-venv-activate functions have been removed. No-one is accessing it yet.

1.1.100 v0.47.0

New Features

- Refactor PyPI release-verify and release-merge templates to download distribution files from a PyPI staging index and upload the files to a PyPI release index. Remove all the builders that were previously used. Call pip upgrade to install latest version of setuptools and twine. Split the PyPI job groups because the verify & merge templates do not accept the same arguments as release-verify & release-merge templates. Remove stream, branch usage; just one PyPI release job per project now. Extend the PyPI release yaml file schema for log_dir and other values. All this makes the PyPI release ver/mrg templates highly similar to the release-job ver/mrg templates. Revise documentation appropriately. Move the PyPI release features to the lf-release-jobs.(rst,yaml) files. Extend the release-job.sh script to have pypi release functions, and drop the pypi-tag-release.sh script.
- Added two new functions: lf-venv-create(), lf-venv-add() Changed name of lf-activate() -> lf-venv-activate() Updated functionality of lf-venv-activate (support for paths & python versions

Bug Fixes

- Added '|| return 1' to appropriate commands. Functions do not support '-e' and a function will normally continue after a command fails.

Other Notes

- Improved Comments

1.1.101 v0.46.0

New Features

- Global job that triggers on any docs changes. Creates docs project if absent Creates subproject association with master doc project Triggers docs build
- Read-the-docs job will run but skip its verification bits unless the repo has .readthedocs.yaml file in the root of its repository. This allows projects to commit changes to their docs/ dir without having to configure the read the docs builds.

Upgrade Notes

- Extend lf_tox_sonar with sonarcloud, sonarcloud-project-key, sonarcloud-project-organization, sonarcloud-api-token, tox-dir and tox-envs properties; also with lf-infra-tox-parameters macro. Use new sonarcloud property as guard for conditional builder step, if true use lf-infra-maven-sonarcloud, else use lf-infra-maven-sonar.

Bug Fixes

- Add missing trigger "remerge" to the PyPI release merge template. Move trigger definitions into PyPI templates, instead of defining four separate trigger definition blocks and using each exactly once. Document the required comment text for the triggers.
- Add macro with lf-infra-wrappers block to set jenkins-ssh-credential parameter to the value in parameter jenkins-ssh-release-credential, which makes the PyPI release merge templates parallel to the release-job merge template. Both need privileges to push a tag on the Jenkins minion. Document the revised configuration parameter. Silence yamllint issues.
- Add missing underscore in two echo commands in release-job.sh. Move comment for shellcheck disable to new line in lf-env.sh. Extend documentation of the container self-release process. Exclude lf-rtdv3.rst from WriteGoodBear coala processing.
- Add a verification step to maven releases to make sure the version being defined in the releases file matches the actual version produced by the maven-stage job that created the release candidate. This is to prevent releases being pushed in Nexus with a version different from what the developer intended in the releases file.
- Add missing {branch} parameter to branch-pattern in gerrit trigger blocks in PyPI release-verify and release-merge templates. Jobs were starting on all defined branches, not limited to target. Change pypi-tag-release script to continue if tag exists, not stop.
- Update to lf-env.sh. Change lf-set-maven-options(): MAVEN_OPTIONS to maven_options to better support shellcheck.

1.1.102 v0.45.0

New Features

- Add lf-env.sh 'library' script in ~jenkins. Sourcing this 'library' script from a bash script provides access to a number of functions. The script is installed by the 'init' script at boot time so that is is accessible from any Jenkins build script. Hopefully over time other functions will be added to this library.

Upgrade Notes

- Add sonar-project-file parameter to ci sonar jobs. By enabling the caller to override the default project file name with an empty string, we enable the ability to provide project settings directly in the sonar-properties field. This removes the requirement for a "sonar-project.properties" file in the repo.

Bug Fixes

- Fix the auto update image script to compare the image type before updating an the image in the source repository. This fixes the bug that updates images although they are the same flavour but a diffirent type.
- Add DRY_RUN parameter to the PyPI verify and merge template common macro because it's required by the pypi_upload.sh script. Adjust verbosity of shell scripts - quiet down pip, add repository name to echo output.
- Correct doc .pypirc test URL to https://test.pypi.org/legacy/ Change doc .pypirc to use API tokens instead of username/password. Use pypi-test as the default repository name in the PyPI merge template (instead of "staging") to match established ONAP practice.

1.1.103 v0.44.1

New Features

- Archive 'sudo' logs. The log will be located in the 'sudo' sub-directory of the archive. The actual name of the log-file depends on the OS of the builder.

Bug Fixes

- Change Packer version back to 1.4.2. Packer 1.4.3 frequently has issues setting metadata after a build completes. Packer 1.4.4 is not expected until October, so this change is needed now to fix the broken builds.
- In the PyPI merge template, add cron parameter to support daily build and push to a staging repo, like the maven merge template. In PyPI release templates, change name of gerrit and github trigger file patterns parameter. This avoids accidental overriding by jobs that limit their actions to subdirectories. The release file patterns are hardcoded in a shell script. Remove params from RST doc. In all PyPI templates, add disabled option and disable-job parameter to be consistent with other python templates.

1.1.104 v0.44.0

New Features

- Add an additonal Sonar job that allows the caller to provide a builder that runs prior to the Sonar scan.
- Add template to update OpenStack cloud images.
- This job finds and updates OpenStack cloud images on the ci-management source repository.
- The job is triggered in two ways:
- 1. When a packer merge job completes, the new image name created is passed down to the job.
  2. Manually trigger the job to update all images.
- When the job is triggered through an upstream packer merge job, this only generates a change request for the new image built.
- When the job is triggered manually, this job finds the latest images on OpenStack cloud and compares them with the images currently used in the source ci-management source repository. If the compared images have newer time stamps are **all** updated through a change request.
- This job requires a Jenkins configuration merge and verify job setup and working on Jenkins.

- New templates to build and push Python source and binary distributions to a PyPI server. Includes: `{project-name}-pypi-verify-{stream}`, `gerrit-pypi-verify`, `github-pypi-verify`, `{project-name}-pypi-merge-{stream}`, `gerrit-pypi-merge`, `github-pypi-merge`, `{project-name}-pypi-release-verify-{stream}`, `gerrit-pypi-release-verify`, `github-pypi-release-verify`, `{project-name}-pypi-release-merge-{stream}`, `gerrit-pypi-release-merge`, `github-pypi-release-merge`,

Upgrade Notes

- Packer merge jobs have a new build parameter when checked also updates the cloud image.
- **lf-infra-packer-build** macro now requires 1 new variables to be passed.
- 1. **update-cloud-image:** Set to true when images need to be updated on Jenkins.

Bug Fixes

- Changed the trigger to run sonar from stage-release to run-sonar. This makes it more concistent with the other parts.
- Builders may have diffrent pyenv versions installed. Programically pick the latest pyenv version. Since we change pyenv version when building images, we do not know which pyenv version are avaliable.
- Run WhiteSource scan jobs weekly on Sunday.
- Pip install pyenv from python2 should force more-itertools to 5.0.0 In a fresh python2.7 venv "pip install pyenv" correctly pulls down more-itertools [required: Any, installed: 5.0.0] If for some reason a higher version is already installed this will downgrade more-itertools to a py2 compatible version
- Allow java-opts to be defined in WhiteSource scans. This avoids java heap failures.

1.1.105 v0.43.1

Upgrade Notes

- CONTAINER_PULL_REGISTRY and CONTAINER_PUSH_REGISTRY need to be defined in Jenkins global environment varaibles.

Bug Fixes

- Pin python-cinderclient to 4.3.0.

  A new version of python-cinderclient 5.0.0 is released which breaks the openstack jobs.
- openstack –os-cloud vex limits show –absolute Error: No module named v1.contrib
- Debug info shows: File "/home/jenkins/.local/lib/python2.7/site-packages/openstackclient/volume/client.py", line 40, in make_client from cinderclient.v1.contrib import list_extensions ImportError: No module named v1.contrib
- Update echo commands in release-job.sh for easy identification.
- Update release-container-schema to use CONTAINER_PULL_REGISTRY and CONTAINER_PUSH_REGISTRY from Jenkins global variables. These will be used to pull and push operations for self release containers. Can be overwritten in the releases files.
- Fix warnings from tox (shellcheck).

- Remove quotes from optional var WSS_UNIFIED_AGENT_OPTIONS. When empty, the quotes will cause WS Unified Agent failures.
- Upgrade WS Unified Agent CLI to latest version 19.8.1

1.1.106 v0.43.0

New Features

- Add python tox merge job templates for gerrit and github. These templates are triggered by merge events to test the Python code on the branch that received the merge. Renamed tox-verify macro to tox-common.

Bug Fixes

- Import GPG signing key in release jobs before verifying Gerrit tag details.
- INFO validate job checks that repositories matches $PROJECT Catches projects that replace / with - in their INFO file Also ensures that repositories only has one entry. We are not supporting multiple projects with a single INFO.yaml file.
- Remove unused the MAVEN_CENTRAL_URL variable. The self-release job is designed to work with any Nexus repository info published in *staging-repo.txt.gz*, which makes the *MAVEN_CENTRAL_URL* redundant, hence remove the unused variable.
- Revise tox-install.sh script to invoke python using environment variable PYTHON instead of hardcoding "python", and remove the pip –quiet flag. Extend the RTD template to pass python-version, default python2.
- Use existing builder lf-infra-maven sonar, drop incomplete builder lf-tox-maven-sonar, to gain desired behavior of pushing code analysis results to Sonar. Use trivial goal 'validate' by default. The script maven-sonar.sh calls maven twice, first with build goals and then with Sonar goals. The incomplete builder did not supply the build goals.
- Release schema verification needs to happen first before we attempt to assign values to the variables. Validate version only after the schema validation has passed and the variables are assigned.
- Organize variable setup into functions. Maven release files expects different variables than container release files.
- Rename "version" variable in container release files to "container_release_tag" which is a better user friendly name given the fact that container versions are rather called tags. Internally, we still process it as "version" to allow reuse of the tag function.

1.1.107 v0.42.1

New Features

- Add DRY_RUN build param to do a test run the job with publishing artifacts.

Upgrade Notes

- Update lftools version to **v0.26.2**.

Bug Fixes

- Verify both repos before attempting release. We have run into a case where the repo on ODL nexus was good, and the repo on Sonatype nexus was missing. Cover this case by running the verify loop over each repo before attempting release.

1.1.108 v0.42.0

New Features

- Add support for distribution_type "container"
- Add function maven_release_file and container_release_file and the logic to choose the correct one. No functional change to maven_release_file.
- Add docker login step when docker releases are being processed.
- container_release_file downloads log_dir/console.log.gz and parses it to get a list of container name and version. Verifies pulls container and grabs the image_id then performs the merge then tags and pushes the container.
- Add lf-sonar-common job-template to lf-ci-jobs.yaml and add lf-infra-sonar to macros.yaml. The purpose of a new job template is to adopt using jenkins sonar plug-in along with the sonar-project.properties file versus pom.xml. Lastly the new job template will ensure that anything using lf-infra-tox-sonar is unaffected.
- Add support for "Build with Parameters" for projects that do not want to use a release file for maven builds.

Upgrade Notes

- release-verify and merge will need to run on a docker build-node for example centos7-docker-8c-8g Lftools will need to be updated to 0.26.0 so that -v is supported for lftools nexus release
- Update lftools version to **v0.26.1**.

Bug Fixes

- Fix missing extension in ID for release-schema.yaml.
- Make "distribution_type" mandatory in future release files.
- Rename "RELEASE_FILE" parameter to "USE_RELEASE_FILE" in release-jobs. This will match the actual varaible default value better and will not collide with the local "release_file" in the script.
- Fix "USE_RELEASE_FILE" if statement. We are now using a bool instead of a string. Changing the if statements to evaluate bools.
- Add pre-build-script parameter to python clm, tox and sonar templates. Gives flexibility to install prerequisite libraries, rearrange the source tree, etc.
- Pin more-itertools ~= 5.0.0, since version 6.0.0 requires Python 3.4 https://github.com/erikrose/more-itertools/releases

  Error: more-itertools requires Python '>=3.4' but the running Python is 2.7.5

- Restructure shell/release-job.sh into functions.

1.1.109 v0.41.0

Upgrade Notes

- Update lftools version to **v0.26.0**.
- Update packer version to 1.4.3. This new packer version fixed an issues in docker image, where its unable to install the packages into docker containers due to checking of wrong container OS. Fix in 1.4.3: builder/docker: Check container os, not host os, when creating container dir default [GH-7939]

Bug Fixes

- Project job sections that define gerrit_trigger_file_paths are overriding ones set in Default parameters of lf_release_common Hard Code gerrit_trigger_file_paths to fix this.

1.1.110 v0.40.4

Upgrade Notes

- Update lftools version to **v0.25.4**.

Bug Fixes

- Git fetch needs the dashed version git fetch "$PATCH_DIR/${PROJECT////-}.bundle". Fix if statement.
- Release creds are only required for promoting the repo, which uses diff ACL as compared to normal user. Therefore dont use the release creds for the verify jobs and the scm sections in both the job templates.
- 1. The release merge job is a one way operation. Given this, Release jobs should only exist in the format {project-name}-release-verify and {project-name}-release-merge As these jobs trigger from a change to any branch/** These jobs Must Exit 0 if release job has already tagged a repo. Inevitably a release file will be pulled in from master to branch or a remerge will be requested This will again trigger the release jobs. since in this case the repo is already tagged, the job should not report a failure. This is solved by having the verfiy and merge exit 0 when the repo is already tagged 2. Rather than use project as defined in the release file use ${PROJECT////-} This changes PROJECT="optf/osdf/foo/bar" to optf-osdf-foo-bar so that we can fetch the log files. by changing /'s in the project names to -'s

1.1.111 v0.40.3

Known Issues

- Update release job template to tigger on any branch name, and not just 'master'. ODL projects branches are version '4.0.x' which requires passing the branch name to the template.

Bug Fixes

- Fix the release job script to handle any trailing '/' set on log_dir and also handle unbound variables correctly.
- Update lftools version to **v0.25.3**.

1.1.112 v0.40.1

Upgrade Notes

- Projects using lf-release-job will need to add the project's signing public key in their Jenkins Settings Files.

Bug Fixes

- Use {GERRIT_PROJECT} when calling Gerrit in release-merge job.
- Project pattern was incorrectly set to ** must be {project}
- The self-release jobs does not handle multiple repositories listed in staging-repo.txt file. This fixes the issue by deriving the NEXUS_URL and the STAGING_REPO from each entry in the file. This approach also eliminates the need for having multiple release.yaml files for every staging-repo.
- Allow lf_release_verify and lf_release_merge to verify tag signature.

1.1.113 v0.40.0

New Features

- Add packer image $NAME to the description setting so that the image name is displayed above the job logs URL. This saves a some time from looking for the image name in the jobs logs.
- Allows projects to promote their own builds. Requires setup of accounts and permissions in Gerrit, Jenkins and Nexus. Please refer to the lf-release-jobs documentation for details.
- Remove orphaned ports from the openstack cloud environment. These orphaned ports are residue of CSIT jobs that needs to be purged, as a part of the openstack cron job. A large number of stale jobs could cause IP address allocation failures.
- Enable JAVA_HOME to point to openjdk12 install path for CentOS 7.

Upgrade Notes

- Consolidated lf-infra-jjbini macros with JJB 2.0. This requires renaming any Jenkins managed files "jjbini-sandbox" to "jjbini" to switch to the format supported in JJB > 2.0.
- Projects using lf-release-jobs need to make sure they have the global variable NEXUSPROXY added in Jenkins production and Jenkins sandbox servers. The value of this variable should be the URL to the project's Nexus server. Previous commit 118b7cbf171aca498d1a0a3a485bad990ad2e7b6 missed this variable.
- Projects using lf-release-jobs need to make sure they have the global variable NEXUSPROXY added in Jenkins production and Jenkins sandbox servers. The value of this variable should be the URL to the project's Nexus server.

- This change will require to update lf-release-job calls. Update from using "{project-name}-releases-merge-{stream}", "{project-name}-releases-verify-{stream}" to "{project-name}-release-merge-{stream}", "{project-name}-release-verify-{stream}". No upgrade need to be done if using "{project-name}-gerrit-release-jobs" group.

Bug Fixes

- There is no way on finding out the $JOB_NAME pushed to sandbox with the jjb-deploy command in the logs. The change outputs the $JOB_NAME to the logs, which is useful for debugging purposes.
- Add release-schema used to validate the releases yaml file as part of lf-release-jobs.
- Tarball the $JAVADOC_DIR as a workaround for javadoc verfiy jobs to avoid uploading a large number of small files. Uploading a large number of small files does not work well with Nexus unpack plugin which fails on 504 gateway timeout.
- Allow maven goals to be configured in sonatype-clm.sh. Set to "clean install" by default.
- Allow maven goals to be configured in maven-sonar.sh. Set to "clean install" by default.
- Add support for JAVA_OPTIONS in sonar job. Some of the maven build options in the sonar job require to set the JAVA_OPTIONS to specific value for the build to pass This option will help to pass the JAVA_OPTIONS from the template.
- Perform "lftools schema verify" command to validate the release files against schema/release-schema.yaml Obtain optional maven central URL inside the loop that scans release files.
- Allow only semantic release versions like "v${SEMVER}" or "${SEMVER}". Fail the script if the version is not valid. Do not append any additional characters to the release version during tag and push steps.
- Delete stacks with the `--force` option to ensure that any delete failures does not stop the openstack-cron jobs from continuing.
- Download raw version of release-schema.yaml to compare against release files using lftools.
- Avoid the usage of project specific variables. Do not use ODLNEXUSPROXY var, but instead use a generalized variable.
- Avoid the usage of project specific variables. Do not use ODLNEXUSPROXY var, but instead use a generalized variable.
- Using "releases" and "release" in different places is becoming confusing. Standardize to "release" to match lftools command and the majority of the exisiting wording.

  Use "releases" for the list of tech team releases and trggers since it is intuitive there. For example "releases/1.1.1.yaml"
- Move info-schema to schema/info-schema.yaml to keep schemas consistency.
- Download only needed files for lf-info-yaml-verify rather than cloning the entire repo.
- Allow lf-maven-stage jobs to be triggered using either "stage-release" or "stage-maven-release".
- Allow lf-maven-docker-stage jobs to be triggered using either "stage-release" or "stage-docker-release".
- Upgrade to the WhiteSource Unified Agent version 19.7.1.

1.1.114 v0.39.1

Bug Fixes

- Update lftools version to **v0.25.2**.

1.1.115 v0.39.0

Critical Issues

- lftools v0.24.0 introduced a major issue which caused the Jenkins cloud configuration merge job for OpenStack clouds to fail. This has been corrected in lftools v0.25.1

Bug Fixes

- Add missing git-url variable in {project-name}-releases-merge-{stream} job.
- Add the {stream} name in releases-verify and releases-merge jobs.
- Some ONAP components like DCAEGEN2 do not host a version.properties file in the root of their repos. We need to be able to provide a location and/or different name for the version.properties file for jobs using the lf-maven-versions-plugin builder step.

1.1.116 v0.38.4

New Features

- Group {project-name}-releases-verify and {project-name}-releases-merge into {project-name}-gerrit-release-jobs.

  Add test jobs for lf-release-jobs.

Bug Fixes

- Change parameter names used to specify container tag method, with a new default to use the fixed string 'latest' as a tag. Merge two shell scripts into one instead of using Jenkins conditional steps. Extend to accept a custom directory for the container-tag.yaml if the yaml-file method is used to set the docker tag information; this is an optional variable which is set to empty by default, and falls back to DOCKER_ROOT.
- Add missing scm block in gerrit-releases-merge job definition. Add missing submodule-disable variable for jobs using lf-infra-gerrit-scm. Update documentation for gerrit-releases-merge and gerrit-releases-verify to remove submodule options as optional parameters.
- Update lftools version to **v0.25.0**
- Add missing $ to variable tag_file so the yq query can pull the tag from the container-tag.yaml file.
- Add -l to /bin/bash shebang line at top of docker-get-container-tag.sh to make it a login shell, which automatically includes /home/jenkins/.local/bin on the path, because that is where pip installs the yq command.

1.1.117 v0.38.3

Bug Fixes

- Add trigger on cron to docker merge macro to support regular rebuilds. This makes the merge macro match the behavior of most other jobs.
- Add yamllint verification to INFO.yaml files.

Other Notes

- Update lftools version to **v0.24.0**.

1.1.118 v0.38.2

Bug Fixes

- Add missing config for triggering on file paths to docker macros and templates, namely gerrit_trigger_file_paths and github_included_regions, to make the verify and merge macros and templates match the behavior of other jobs.

1.1.119 v0.38.1

Bug Fixes

- Install yq to be used to read yaml files. In specific, it will be needed to read container-tag.yaml.
- When calling builder step macros "lf-docker-get-container-tag", "lf-docker-build" and "lf-docker-push", make sure the needed variables are also passed explicitly to avoid these variables appear as undefined.
- Allow DOCKER_ARGS to be empty in docker-build.sh. This is not a required parameter, it can be empty.

  Remove container reference in docker-get-git-describe.sh. The CONTAINER_PUSH_REGISTRY already gets added in the docker-push script. No need to add it again.

  Rename image_name to image_build_tag in docker-get-yaml-tag to match docker-get-git-describe. Add missing "DOCKER_NAME" in the DOCKER_IMAGE.

  docker-get-git-describe.sh and docker-get-yaml-tag.sh should only export the tag variable. Let docker-build.sh process DOCKER_NAME

Other Notes

- Add yq install as part of python-tools-install.sh Allow future scripts to use yq package.

1.1.120 v0.38.0

New Features

- **gerrit-tox-verify** now has a new parameter `gerrit-skip-vote` (bool) to control whether Jenkins should skip voting depending on the build outcome. It defaults to `false` since it is the default used by the Jenkins Gerrit Trigger Plugin.
- gerrit-docker-verify runs for new commits and runs a build of the affected Docker images.
- gerrit-docker-merge runs for merged commits, runs a build of the affected Docker images and pushes the images to a specified Docker registry.
- New `lf-release-job-merge` and `lf-release-job-verify` templates allow projects to have self-serve releases. Project will create a tagname.yaml file in the releases/ directory of their git repo. example:

```
$ cat releases/4.0.0.yaml
---
distribution_type: 'maven'
version: '4.0.0'
project: 'odlparent'
log_dir: 'odlparent-maven-release-master/11/'
#below is optional
maven_central_url: 'oss.sonatype.org'
```

- **lf-infra-gerrit-scm** and **lf-infra-github-scm** now require a `submodule-disable` parameter (bool) to control whether submodules are ignored or not during git fetch operations.
- All job-templates now provide an optional `submodule-disable` parameter for git fetch operations, defaulting to `false`.

Upgrade Notes

- Any project using the **lf-infra-gerrit-scm** and **lf-infra-github-scm** macros in global-jjb should need to add a `submodule-disable` value. It is recommended to default this value to `false` since it is the default used by the Jenkins Git Plugin.
- Update gerrit comment trigger to use a more standard regex and avoid triggering jobs, when these keywords are intended to be used as code review comments between users. Also improve the regexs to make them more succinct and readable.

Bug Fixes

- Add missing config for triggering on file paths to maven stage macros and templates, namely gerrit_trigger_file_paths nad github_included_regions, to make those macros and templates match the behavior of maven verify and maven merge.
- fix multiple jobs created using same job-template update same github check status due to hard coded status-context to Maven Verify. Now appending status-context with maven-version and java-version to make it unique. And create different status checks in the github. fix applied for maven verify and maven docker verify jobs
- Fix log shipping script to not require a LOGS_SERVER. There was a regression that caused the log shipping script to start requiring a LOGS_SERVER which fails in the case of a system that does not have that optional environment variable set.
- Handle multiple search extension or patterns passed by upstream JJB ARCHIVE_ARTIFACTS param as a single string by spliting these values before being passed to `lftools deploy archives`.

```
ARCHIVE_ARTIFACTS="**/*.prop \
                   **/*.log \
                   **/target/surefire-reports/*-output.txt \
                   **/target/failsafe-reports/failsafe-summary.xml \
                   **/hs_err_*.log **/target/feature/feature.xml"
```

For example, the above env variable passed to the script and to `lftools deploy archives` as:

```
lftools deploy archives -p **/*.prop \
                   **/*.log \
                   **/target/surefire-reports/*-output.txt \
                   **/target/failsafe-reports/failsafe-summary.xml \
                   **/hs_err_*.log **/target/feature/feature.xml \
                   "$NEXUS_URL" \
                   "$NEXUS_PATH" \
                   "$WORKSPACE"
```

The correct way of passing this as per lftools implmentation is:

```
lftools deploy archives -p '**/*.prop' \
                   -p '**/*.log' \
                   -p '**/target/surefire-reports/*-output.txt' \
                   -p '**/target/failsafe-reports/failsafe-summary.xml' \
                   -p '**/hs_err_*.log' \
                   -p '**/target/feature/feature.xml' \
                   "$NEXUS_URL" \
                   "$NEXUS_PATH" \
                   "$WORKSPACE"
```

- Fix error with handling unbound arrays for search extensions, when using *set -u*. The correct way of using this.

```
set -u
arr=()
echo "output: '${arr[@]}'"
bash: arr[@]: unbound variable
echo "output: '${arr[@]:-}'"
foo: ''
```

- lf-maven-versions-plugin builder step needs to run before maven-patch-release.sh as this second script contains a condition to confirm if the maven vesions plugin was selected as a way to remove the 'SNAPSHOT' pattern from the pom.xml files. lf-maven-docker-stage was based on lf-maven-stage and it seems that these particular builder steps were switched in place accidentally.
- The logs-deploy.sh script now allows ARCHIVE_ARTIFACTS to contain zero or more files.
- request-2.22.0 does not work with python-3.4.9, so pin requests to v2.21.0 to address the tox failures.

1.1.121 v0.37.2

Bug Fixes

- Use maven goal install (not deploy) in the maven + docker verify job. An image cannot be pushed by a verification job, and the deploy target directs the plugin to push.

1.1.122 v0.37.1

Bug Fixes

- Remove maven-versions-plugin-set-version variable in newly added macro. This is a variable that does not need to be defined by the users of the jobs. The version needed in this builder step is inherited from versions.properties as "release_version".

1.1.123 v0.37.0

New Features

- Add verify, merge and stage templates for Java projects that build and wrap a JAR (e.g., a Spring-Boot application) inside a Docker image, and do not need to deploy any JAR libraries or POM files.

Upgrade Notes

- The next release of common-packer will require a minimum version of 1.3.2 for packer. The current release of packer is 1.4.0.

Bug Fixes

- The packer-merge job for Gerrit systems was improperly configured to use the Gerrit Trigger choosing strategy and not default. This caused issues unexpected issues with retriggering merged changes when the expectation was that it would pick up the lastest change as per normal.
- This is a variable that does not need to be defined by the users of the jobs. The version needed in this builder step is inherited from versions.properties as "release_version" and it is fixed as that. This also helps teams not having to define this version in 2 places and just rely on version.properties.
- Projects using maven versions plugin let this plugin take care or updating their versions in the pom.xml. When maven-versions-plugin is set to "true", skip the stripping of SNAPSHOTS from the pom.xml files. maven-versions-plugin is set to "false" by default.

Other Notes

- Update example Jenkins Init Script in README to redirect all output to a log file.

1.1.124 v0.36.0

Prelude

WhiteSource is a security and license compliance management platform. It is used to perform scans on a great variety of coding and scripting languages.

New Features

- New `comment-to-gerrit` builder will comment back to gerrit patchset if a file called gerrit_comment.txt is created by the build.
- Allows maven to run a clean install step before the WhiteSource scan runs the Unified Agent to fetch additional dependencies. Set to false by default.
- Job {project-name}-whitesource-scan-{stream} uses the WhiteSource Unified Agent scanner CLI tool to perform the code scan and report the results into the WhiteSource dashboard.

Bug Fixes

- Tag releases will now trigger a docs build to regenerate and update the release note link.
- Update jenkins-cfg-verify job to validate new images names obtained from $GERRIT_REFSPEC instead of the master branch.
- Hardcode project version to the "GERRIT_BRANCH". Follow previous convention from CLM where reports were versioned after the branch name. Fix minor nits with bash varaibles.
- wss-unified-agent.config file should not be opened for configuration to tech teams. The config file should be part of Jenkins Settings Files and called via Managed Files. wss-unified-agent.config must be created in Jenkins config files based on wss-unified-agent.config.example.

Other Notes

- To run this job, a configuration file is needed (wss-unified-agent.config.example). A new secret text credential will need to be created. (ID=wss-apiKey Secret=WhiteSource organization API key)
- Update lftools version to **v0.23.1**.

1.1.125 v0.35.0

New Features

- The **jjb-merge** job now has a new parameter `jjb-workers` to allow configuration of the number of threads to run update with. Default is *0* which is equivalent to the number of CPU cores available on the system.
- New `info-vote-verify` macro Will count votes against an INFO.yaml change and sumbit automatically if a majority of committers vote +1 or +2 on the change. Job is triggered by +2 votes or a comment of "vote"

Other Notes

- The Maven Verify job will now call `-Dmaven.source.skip` to skip source jar generation in the verify job. This saves us some time in the verify build as the source artifacts are not useful in a verify job.

1.1.126 v0.34.0

New Features

- **jenkiins-init-scripts** The 'ciman' repo is not longer required to be located in '/opt/ciman'.
- lf-maven-set-version conditional step for lf-maven-stage to allow teams to run Maven versions plugin to update their artifact versions. Step will run if maven-versions-plugin is set to true.
- Support for the Throttle Plugin is added to JJB jobs so static build servers can restrict the number of concurrent JJB jobs ran at the same time.

  This must be explicitly enabled by setting *throttle-enabled* on the jobs.

Bug Fixes

- Adapt maven path search for files and dirs. The "-f" maven param can specify both a directory, in which case it will look for "pom.xml" in the directory, or a specific file. The original version of this search was only compatible with directories that contain a pom.xml file.
- Update the lf-maven-cental macro documentation and example templates with the missing requireed params.
- Fix JAVA_HOME for openjdk11 on CentOS 7 to use the OpenJDK version installed in /usr/lib/jvm/java-11-openjdk.
- The JJB Deploy Job is configured to trigger only if the Gerrit comment starts with the *jjb-deploy* keyword.

  Without the regex being optimized the job triggers on any occurance of the *jjb-deploy* keyword in a Gerrit comment, with is waste infra resources.

  Example of a valid command in Gerrit comment that triggers the job:

  ```
  jjb-deploy builder-jjb-*
  ```

  Example of a invalid command in Gerrit comment that would _not_ trigger the job:

  ```
  Update the job. jjb-deploy builder-jjb-*
  ```

1.1.127 v0.33.0

New Features

- **jenkins-init-scripts** If the environmental variable 'SWAP_SIZE' is set when the 'init.sh' script is called, then a 'SWAP_SIZE' GB swap space will be configured. If 'SWAP_SIZE' is '0' or is not a valid integer, then no swap space is configured. If it is unset then 1GB of swap will be configured. Previously the swap size was fixed at 1GB.
- **jenkins-init-scripts** If the work directory or volume (/w) aleady exists, the ownership will be recursivly set to 'jenkins:jenkins'. Previously only the top directory /w was owned by 'jenkins:jenkins'
- **lf-sigul-sign-dir** macros now supports a `sign-mode` parameter which allows jobs to choose to sign artifacts using either *parallel* mode or *serial* mode (default).

Upgrade Notes

- **lf-sigul-sign-dir** users need to add a new parameter `sign-mode` to their job-templates setting either *parallel* or *serial* as the value, we recommend setting serial mode for this setting.

  {project-name}-maven-stage-{stream}'s Sigul signer now defaults to *serial* mode instead of the previous *parallel* behaviour. To change this back to the previous behaviour pass the "sign-mode" parameter to the job template:

```
- project:
    name: parallel-sign
    jobs:
      - gerrit-maven-stage:
          sign-mode: parallel
```

1.1.128 v0.31.0

New Features

- New job-template **{project-name}-release-announce** for lf-releng projects to automate release announcement emails.
- Add support for pushing Sonar results to SonarCloud. Refer to *Maven Sonar docs* for details.

Upgrade Notes

- Jobs using the **lf-maven-stage** macro now need to update to the new usage. Preparation calls to **lf-provide-maven-settings**, **lf-infra-create-netrc**, and **lf-provide-maven-settings-cleanup** are no longer necessary to prepare the **lf-maven-stage** macro.

  Usage:

```
- lf-maven-stage:
   mvn-global-settings: 'global-settings'
   mvn-settings: 'settings'
   mvn-staging-id: 'staging profile id'
```

1.1.129 v0.30.0

New Features

- Packer merge jobs now include the image name in the Jenkins build description.

Bug Fixes

- Extend ${JOB_NAME} to include {java-version} parameter to support jobs to build with multiple versions of openjdk{8,11}.
- Modified lf-maven-jobs.yaml sonar cron entry to '{obj:cron}' to pass value from custom user config file.

1.1.130 v0.29.0

New Features

- Add a puppet-verify job to lf-ci-jobs. This job will perform Puppet linting on the specified repository.

```
- project:
    name: lf-infra-puppet-mymodule
    project-name: lf-infra-puppet
    project: puppet/modules/mymodule
    jobs:
      - gerrit-puppet-verify
```

Bug Fixes

- maven-fetch-metadata.sh was not respecting the "-f" (for file path) flag in MAVEN_PARAMS, causing lf-maven-merge jobs that utilize this flag to fail. It will now set a path based on this flag if it is present, or default to the current working directory.
- Check openjdk $VERSION before setting $JAVA_HOME. This enables jobs to pass "openjdk10" or "openjdk11" on CentOS 7 images to use the OpenJDK version installed in /opt.

1.1.131 v0.28.3

Bug Fixes

- Compress and upload all jjb-verify XML files to Nexus, to ease out the IOPs on cron jobs that manage the logs on Nexus and optimize job performace by ~8 mins. This is because the job generates around ~2.3K XML files (small files) which is uploaded to Nexus in every run of jjb-verify. Doing this is faster as compared to the Nexus Unpack plugin in the Nexus end unpacking the zip file we upload takes longer.

1.1.132 v0.28.1

Other Notes

- Update lftools version to **v0.19.0**.

1.1.133 v0.28.0

New Features

- New `lf-stack-create` macro allows job-templates to setup a OpenStack Heat stack, useful for spinning up CSIT labs to run integration tests against. Use with the `lf-stack-delete` macro.
- Concurrency for the gerrit-jjb-verify job can now be configured by setting the 'build-concurrent' parameter.
- New macro **lf-maven-central** is available to deploy artifacts to OSSRH staging for jobs that want to eventually deploy to Maven Central.

```
---
- job-template:
    name: lf-maven-central-macro-test

    ####################
    # Default variables #
    ####################

    mvn-central: true
    mvn-global-settings: ""
    mvn-settings: ""
    ossrh-profile-id: ""

    ####################
    # Job configuration #
    ####################

    builders:
      - lf-maven-central:
          mvn-central: "{mvn-central}"
          mvn-global-settings: "{mvn-global-settings}"
          mvn-settings: "{mvn-settings}"
          ossrh-profile-id: "{ossrh-profile-id}"
```

- The GERRIT_REFSPEC build parameter can now be used to trigger a test build from the Jenkins Sandbox system against a work in progress packer image patch from a GitHub Pull Request.

Upgrade Notes

- `lf-stack-delete` has been modified to be a companion macro to `lf-stack-create` in order to cleanup the stack at the end of a job run. It now includes a required parameter **openstack-cloud** to choose the `clouds.yaml` cloud configuration for the project. Existing users of this macro will need to update their job templates accordingly.
- Requires JJB 2.8.0 for the jenkins-sandbox-cleanup job to not fail.

> **Note:** Despite the failure if JJB 2.8.0 is not available the job will successfully delete all jobs and views, the primary purpose of this job.

Bug Fixes

- RELENG-1450 All view disappears on Jenkins Sandbox after views are deleted. The **All** view is now recreated after `delete-all` is run.

1.1.134 v0.27.0

New Features

- Add the ability to configure the location of JJB's cache directory for CI jobs.
- New view-templates `project-view`, `common-view`, and `csit-view` are available for projects to manage Jenkins views through code.

  To use the `project-view` template in a project:

```
- project:
    name: aaa-view
    views:
      - project-view

    project-name: aaa
```

  To use the `common-view` template in a project:

```
- project:
    name: daily-builds
    views:
      - common-view

    view-name: Periodic
    view-regex: '.*-periodic-.*'
```

  To use the `csit-view` template in a project:

```
- project:
    name: csit
    views:
      - csit-view

    view-name: CSIT
    view-regex: '.*csit.*'

- project:
    name: csit-1node
    views:
      - csit-view

    view-name: CSIT-1node
    view-regex: '.*-csit-1node-.*'
```

- Add support to maven-stage jobs to publish to Maven Central via OSSRH.

  This is accomplished by adding these 2 new optional parameters to the job configuration.

```
- gerrit-maven-stage:
    mvn-central: true
    ossrh-profile-id: 7edbe315063867
```

- The **openstack-cron** job now has the ability to remove images older than a specified age (default: 30).
- The **openstack-cron** job now has the ability to remove orphaned servers.
- The **openstack-cron** job now has the ability to remove orphaned stacks.
- The **openstack-cron** job now has the ability to remove orphaned volumes.
- **lf-infra-gerrit-scm** and **lf-infra-github-scm** now require a `submodule-timeout` parameter to provide a timeout value (in minutes) for git fetch operations.
- All job-templates now provide an optional `submodule-timeout` parameter for git fetch operations, defaulting to 10 minutes.

Upgrade Notes

- Some LF projects are already using a `common-view` template in their local ci-management repo. This `common-view` is called `project-view` in global-jjb so rename all instances of `common-view` to `project-view` when upgrading and remove the local `common-view` view-template definition from ci-management.
- The **openstack-cron** job now requires a new parameter configured `jenkins-urls` in order to use the job.
- Any project using the **lf-infra-gerrit-scm** and **lf-infra-github-scm** macros in global-jjb should need to add a `submodule-timeout` value. It is recommended to default this value to `10` since it is the default used by the Jenkins Git Plugin.

Bug Fixes

- The jenkins-init scripts dir is now updated to reflect changes recommended for the v0.25.0 release. We unfortunately missed this critical piece in the v0.25.0 release.
- Specify refspec to be blank for SCM config on github-maven-merge job. Setting the refspec to *+refs/pull/*:refs/remotes/origin/pr/** causes there to be no merge job triggered.
- New detox version requires tox >= 3.5 and < 4. Instead of explicitly pulling tox, we are now implicitly pulling via detox. (https://jira.opendaylight.org/browse/RELENG-136)

Other Notes

- lftools' `openstack` module will now be installed as part of pre-build.
- The **openstack-cron** job now runs every hour instead of daily. This is because stack cleanup should happen more regularly.

1.1.135 v0.26.0

New Features

- Add a new `nexus-iq-namespace` optional parameter to insert a namespace into Nexus IQ AppID. This is useful for shared Nexus IQ systems where projects might have concern about namespace collision.

  Note: We recommend when using the namespace to add a trailing - to the value. Eg. 'odl-', this is to make the namespace look nice for example "odl-aaa" is the result of namespace odl-, and project name aaa.

- Add `lf-infra-publish-windows`. A publisher for use at the end of Windows based job-templates.

Bug Fixes

- Fix packer-verify job to correctly work with `clouds.yaml` config model implemented in global-jjb v0.25.0.

1.1.136 v0.25.1

Bug Fixes

- jjb-cleanup.sh may be merged with other shell scripts that `set -u` which causes Jenkins to fail when activating virtualenvs

1.1.137 v0.25.0

New Features

- Add support to the packer-build job to use clouds.yaml for openstack builder configuration rather than through the cloud-env file. This allows us to simplify the template configuration for openstack builders moving forward.
- New macro `lf-sigul-sign-dir` available to sign artifacts in a provided directory using Sigul.

  Usage:

```
- lf-sigul-sign-dir:
    sign-dir: '$WORKSPACE/m2repo'
```

  This macro also requires a boolean variable to `SIGN_ARTIFACTS` to be set to true to activate the macro. We recommend the job-template that uses this macro to define it in the job parameters section.

  Example:

```
- bool:
    name: SIGN_ARTIFACTS
    default: '{sign-artifacts}'
    description: Use Sigul to sign artifacts.
```

- Add Sigul signing support to the maven-staging job. To activate Sigul signing make sure to set `sign-artifacts: true`. Example:

```
- project:
    name: abc
    jobs:
      - gerrit-maven-stage

    sign-artifacts: true
```

- Add lf-stack-delete macro to delete an openstack heat stack at the end of the job.

  This macro requires a parameter defined in the job named STACK_NAME containing the name of the stack to delete.

- Add lf-infra-wrappers-windows to handle Windows specific wrapper configuration.
- Refactor lf-infra-wrappers to be for Linux systems and split out the non-linux specific components into a new lf-infra-wrappers-common. This change is seamless for current users of lf-infra-wrappers.

Upgrade Notes

- Upgrade to global-jjb v0.24.6 before performing this upgrade. This ensures that jjb-verify job pulls in a regex fix that will allow it to verify the v0.25.0 upgrade.
- Global JJB now has non-JJB YAML configuration and requires action on the ci-management repo when upgrading to this version of Global JJB to prevent JJB from picking up these YAMLs as config. Follow the instructions below BEFORE upgrading globall-jjb:

```
cd <git-root>
git mv jjb/global-jjb global-jjb
mkdir jjb/global-jjb
ln -s ../../global-jjb/shell jjb/global-jjb/shell
ln -s ../../global-jjb/jjb jjb/global-jjb/jjb
git add jjb/global-jjb
git commit -sm "Prepare repo for global-jjb v0.25.0"
```

- Minimum packer version 1.2.5 is now required for the packer-build job.
- **lf-infra-packer-build** macro now requires 2 new variables to be passed.
  1. **openstack:** Set to true if template is built using the openstack builder
  2. **openstack-cloud:** The clouds.yaml cloud to use when running `packer build`

Deprecation Notes

- lftools-install.sh is deprecated and will be removed in a future release. We recommend installing lftools via *pip install –user lftools* to install instead of using this script.

Bug Fixes

- Fix *pip install pip setuptools* which seems to fail against the Nexus 3 proxy. Run them as separate calls to make things happier.
- jjb-verify will now test on all changes in the jjb directory. The previous pattern was too specific and sometimes missed verifying patches that should be verified.
- Replace jjb-verify to test on all changes in the shell/* directory.
- Fix the lftools virtualenv workaround we had to put in place in the tox-verify job by using `pip install --user` for global tool installs.
- Fix jobs failing with UNSTABLE build due to install pip==18.0 missing. This change moves all the jobs to using lf-infra-pre-build to install lftools via –user command.
- Use *python -m pip* to ensure that we are using the pip version that was installed rather than the OS wrapper version of pip.
- Fix package listing script in post-builder from causing UNSTABLE build due to difference in the two files being compared.
- Fix RTD job failing to find PBR install.

Other Notes

- Update lftools to ~ 0.17.1

CHAPTER

TWO

GUIDES

2.1 Install

global-jjb requires configuration in 2 places; `Jenkins` and the *ci-management* repository.

2.1.1 Jenkins configuration

On the Jenkins side, we need to prep `environment variables` and `plugins` required by the jobs in global-jjb before we can start our first jobs.

Install Jenkins plugins

Install the following required Jenkins plugins and any optional ones as necessary by the project.

Required

- Config File Provider
- Description Setter
- Environment Injector Plugin
- Git plugin
- Post Build Script
- SSH Agent
- Workspace Cleanup

Required for Gerrit connected systems

- Gerrit Trigger

Required for GitHub connected systems

- GitHub plugin
- GitHub Pull Request Builder

Optional

- Mask Passwords
- MsgInject
- OpenStack Cloud

- Timestamper

Environment Variables

The *Jenkins Configuration Merge* job can manage environment variables job but we must first bootstrap them in Jenkins so that the job can run and take over.

Required:

```
GIT_URL=ssh://jenkins-$SILO@git.opendaylight.org:29418
JENKINS_HOSTNAME=jenkins092
NEXUS_URL=https://nexus.opendaylight.org
SILO=production
SONAR_URL=https://sonar.opendaylight.org
```

Gerrit:

```
GERRIT_URL=https://git.opendaylight.org/gerrit
```

GitHub:

```
GIT_URL=https://github.com
GIT_CLONE_URL=git@github.com:
```

Note: Use `GIT_CLONE_URL` for GitHub projects as this will be different from the URL used in the properties configuration.

Optional:

```
LOGS_SERVER=https://logs.opendaylight.org
```

Steps

1. Navigate to https://jenkins.example.org/configure
2. Configure the environment variables as described above
3. Configure the same environment variables in the *ci-management* repo

2.1.2 ci-management

ci-management is a git repository containing *JJB* configuration files for Jenkins Jobs. Deploying Global JJB here as a submodule allows us easy management to install, upgrade, and rollback changes via git tags. Install Global JJB as follows:

1. Install Global JJB

```
GLOBAL_JJB_VERSION=v0.1.0
git submodule add https://github.com/lfit/releng-global-jjb.git global-jjb
cd global-jjb
git checkout $GLOBAL_JJB_VERSION
cd ..

# Setup symlinks
```

(continues on next page)

(continued from previous page)

```
mkdir -p jjb/global-jjb
ln -s ../../global-jjb/jenkins-init-scripts jjb/global-jjb/jenkins-init-scripts
ln -s ../../global-jjb/shell jjb/global-jjb/shell
ln -s ../../global-jjb/jjb jjb/global-jjb/jjb
git add jjb/global-jjb

git commit -sm "Install global-jjb $GLOBAL_JJB_VERSION"
```

Note: We are purposely using github for production deploys of global-jjb so that uptime of LF Gerrit does not affect projects using global-jjb. In a test environment we can use https://gerrit.linuxfoundation.org/infra/releng/global-jjb if desired.

2. Setup `jjb/defaults.yaml`

   Create and configure the following parameters in the `jjb/defaults.yaml` file as described in the *defaults.yaml configuration docs <defaults-yaml>*.

   Once configured commit the modifications:

```
git add jjb/defaults.yaml
git commit -sm "Setup defaults.yaml"
```

3. Push patches to Gerrit / GitHub using your favourite push method

At this point global-jjb installation is complete in the *ci-management* repo and is ready for use.

2.1.3 Deploy ci-jobs

The CI job group contains jobs that should deploy in all LF Jenkins infra. The minimal configuration to deploy the **{project-name}-ci-jobs** job group as defined in **lf-ci-jobs.yaml** is as follows:

jjb/ci-management/ci-management.yaml:

```
- project:
    name: ci-jobs

    jobs:
      - '{project-name}-ci-jobs'

    project: ci-management
    project-name: ci-management
    build-node: centos7-builder-2c-1g
```

Required parameters:

project The project repo as defined in source control.

project-name A custom name to call the job in Jenkins.

build-node The name of the builder to use when building (Jenkins label).

Optional parameters:

branch The git branch to build from. (default: master)

jjb-version The version of JJB to install in the build minion. (default: <defined by the global-jjb project>)

2.1.4 Deploy packer-jobs

The packer job group contains jobs to build custom minion images. The minimal configuration needed to deploy the packer jobs is as follows which deploys the **{project-name}-packer-jobs** job group as defined in **lf-ci-jobs.yaml**.

jjb/ci-management/packer.yaml:

```
- project:
    name: packer-builder-jobs

    jobs:
      - '{project-name}-packer-jobs'

    project: ci-management
    project-name: ci-management
    branch: master
    build-node: centos7-builder-2c-1g

    platforms:
      - centos
      - ubuntu-16.04

    templates: builder

- project:
    name: packer-docker-jobs

    jobs:
      - '{project-name}-packer-jobs'

    project: ci-management
    project-name: ci-management
    branch: master
    build-node: centos7-builder-2c-1g

    templates: docker

    platforms:
      - centos
      - ubuntu-16.04
```

Required parameters:

project The project repo as defined in source control.

project-name A custom name to call the job in Jenkins.

build-node The name of the builder to use when building (Jenkins label).

platforms A list of supported platforms.

templates A list of templates to build. We recommend setting one template per `project` section so that we can control which platforms to build for specific templates.

Optional parameters:

branch The git branch to build from. (default: master)

packer-version The version of packer to install in the build minion, when packer is not available. (default: <defined by global-jjb>)

2.2 Configuration

2.2.1 defaults.yaml

This file lives in the ci-management repo typically under the path `jjb/defaults.yaml`. The purpose of this file is to store default variable values used by global-jjb templates.

Required

jenkins-ssh-credential The name of the Jenkins Credential to use for ssh connections. (ex: jenkins-ssh)

lftools-version Version of lftools to install. Can be a specific version like '0.6.1' or a PEP-440 definition For example *<1.0.0* or *>=1.0.0,<2.0.0*.

mvn-site-id Maven Server ID from settings.xml containing the credentials to push to a Maven site repository.

mvn-staging-id Maven Server ID from settings.xml containing the credentials to push to a Maven staging repository.

Gerrit required parameters:

gerrit-server-name The name of the Gerrit Server as defined in Gerrit Trigger global configuration. (ex: Primary)

GitHub required parameters:

git-url Set this to the base URL of your GitHub repo. In general this should be https://github.com. If you are using GitHub Enterprise, or some other GitHub-style system, then it should be whatever your installation base URL is. This sets a job property that GitHub Pull Request Builder requires to work. Note that this is the web url to your project: (eg. https://github.com/\protect\T1\textdollarORG/\protect\T1\textdollarPROJECT)

git-clone-url This is the clone prefix used by GitHub jobs. Set this to either the same base url as **git-url**, or to 'git@github.com:' including the trailing ':'. Determined by your clone method (https or git).

github-org The name of the GitHub organization interpolated into the scm config.

github_pr_org The name of the GitHub organization. All members of this organization will be able to trigger jobs.

github_pr_allowlist List of GitHub members you wish to be able to trigger jobs.

github_pr_admin_list List of GitHub members that will have admin privileges on the jobs.

Example Gerrit Infra:

```
- defaults:
    name: global

    # lf-infra defaults
    jenkins-ssh-credential: jenkins-ssh
    gerrit-server-name: OpenDaylight
    lftools-version: '<1.0.0'
    mvn-site-id: opendaylight-site
    mvn-staging-id: opendaylight-staging
```

Example GitHub Infra:

```
- defaults:
    name: global

    # lf-infra defaults
    jenkins-ssh-credential: jenkins-ssh
    github-org: lfit
    github_pr_allowlist:
      - jpwku
      - tykeal
      - zxiiro
    github_pr_admin_list:
      - tykeal
    lftools-version: '<1.0.0'
    mvn-site-id: opendaylight-site
    mvn-staging-id: opendaylight-staging
```

2.2.2 Jenkins Files

global-jjb makes use of the Jenkins Config File Provider plugin to provide some default configurations for certain tools. This section details the files to define in Jenkins' **Managed Files** configuration (eg: https://jenkins.example.org/configfiles/index).

npmrc

This file contains default npmrc configuration and lives in $HOME/.npmrc. Documentation for npmrc is available via the npm project.

Required This file MUST exist. An empty file is acceptable if a proxy is not available for the project.

type Custom file

Create a **Custom file** with contents:

```
registry = https://nexus.opendaylight.org/content/repositories/npmjs/
```

clouds-yaml

Needed by `openstack client` and `packer` to fetch OpenStack credentials and configuration. This file is OpenStack's clouds.yaml file.

Optional Needed for jobs that use `openstack client`. `packer` if building against OpenStack infra.

type Custom file

Create a **Custom file** with contents:

```
clouds:
vex:
 auth:
   project_name: OS_PROJECT_NAME
   username: OS_USERNAME
```

(continues on next page)

(continued from previous page)

```
  password: OS_PASSWORD
  auth_url: 'https://auth.vexxhost.net/v3/'
  user_domain_name: Default
  project_domain_name: Default
region_name: ca-ymq-1
```

Warning: If using packer 1.3.0 make sure that the clouds.yaml **profile** configuration is **NOT** configured. Using **profile** causes packer to look for another file called `clouds-public.yaml` for configuration.

pipconf

This file contains default configuration for the python-pip tool and lives in $HOME/.config/pip/pip.conf. Documentation for pip.conf is available via the pip project.

Required This file MUST exist. An empty file is acceptable if a proxy is not available for the project.

type Custom file

Create a **Custom file** with contents:

```
[global]
timeout = 60
index-url = https://nexus3.opendaylight.org/repository/PyPi/simple
```

jjbini

This file contains the Jenkins Job Builder configuration for *CI Jobs*.

Required This file MUST exist.

type Custom file

Create a **Custom file** with contents:

```
[job_builder]
ignore_cache=True
keep_descriptions=False
include_path=.:scripts:~/git/
recursive=True

[jenkins]
user=jenkins-jobbuilder
password=1234567890abcdef1234567890abcdef
url=https://jenkins.example.org
query_plugins_info=False

[production]
user=jenkins-jobbuilder
password=1234567890abcdef1234567890abcdef
url=https://jenkins.example.org
query_plugins_info=False
```

(continues on next page)

(continued from previous page)

```
[sandbox]
user=jenkins-jobbuilder
password=1234567890abcdef1234567890abcdef
url=https://jenkins.example.org/sandbox
query_plugins_info=False
```

The last 2 sections are for the `jenkins-cfg` job use, they should match the `silo` names for the respective Jenkins systems, typically `production` and `sandbox`.

jenkins-log-archives-settings

See *lf-infra-ship-logs* for usage. If not archiving logs then keep this file with default settings, global-jjb needs the file to exist to function.

Requires a *credential* named 'logs' of type 'Username and Password' created in the Jenkins Credentials system.

1. Add Server Credentials
2. Set `ServerId` to `logs`
3. Set `Credentials` to the `logs` user created in the Credentials System

   Required This file MUST exist if using log archiving.

   type Maven settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.
↪org/xsd/settings-1.0.0.xsd">
</settings>
```

Note: This example is the default boilerplate generated by Jenkins with the comments stripped out. We can also use the default generated by Jenkins without modifying it.

packer-cloud-env

Cloud environment configuration variables for Packer jobs. These can contain credentials and configuration for whichever clouds packer jobs are using.

Required This file MUST exist to use packer jobs.

type Json file

```
{
  "cloud_auth_url": "https://auth.vexxhost.net/v3/",
  "cloud_tenant": "TENANT_ID",
  "cloud_user": "CLOUD_USERNAME",
  "cloud_pass": "CLOUD_PASSWORD",
  "cloud_network": "CLOUD_NETWORK",
```

(continues on next page)

(continued from previous page)

```
  "ssh_proxy_host": ""
}
```

2.2.3 Jenkins CI Jobs

jenkins-cfg-merge

This job manages Jenkins Global configuration. Refer to the *CI Documentation* for job configuration details.

2.2.4 Log Archiving

The logs account requires a Maven Settings file created called **jenkins-log-archives-settings** with a server ID of **logs** containing the credentials for the logs user in Nexus.

2.3 Best Practices

2.3.1 JJB YAML Layout

Note: While some of this applies to the Global JJB project other recommendations are generally useful to projects that might be defining JJB templates.

The Global JJB project is a useful example project to look at so we recommend referring to the Maven job definitions as an example as you read the documentation below:

https://github.com/lfit/releng-global-jjb/blob/master/jjb/lf-maven-jobs.yaml

We recommend sectioning off the template into 3 general sections in order:

1. Job Groups (optional)
2. Common Functions
3. Job Template Definitions

In section 1) not all configurations need this so is an optional section. Job groups are useful in cases where there are jobs that are generally useful together. For example the OpenDaylight uses a lot of Merge and Verify job combinations so every new project will want both job types defined in their project.

In section 2) we want to define all common functions (anchors, aliases, macros) that are generally useful to all jobs in the file. This allows job template developers to look at the top of the file to see if there are useful functions already defined that they can reuse.

In section 3) we can declare our job definitions. In the Global JJB project we create Gerrit and GitHub versions of the jobs so the format we use here might look strange at first but is well layed out for code reuse if we need to define 2 or more versions of the same job template for different systems. We will define this in more detail in the next section.

Job Template Layout

1. Comment of Job Template Name
2. Macro containing build definition of the job a. Macro named after job b. Complete documentation of the job parameters c. Default parameters defined by the job d. Job configuration
3. job-template definition containing build triggers

In section 1) we need to declare a in large comment text to identify the job section.

In section 2) we declare the actual job definition. This is so that we have a single macro that we call in all the real job-template sections that is reusable and not duplicating any code. First we declare the macro as the job name. Then in 2.b) we provide the complete documentation of the job parameters this is so that we can link users of the job to this file and they can understand fully what options they can configure for this particular job. Then we define defaults for any parameters that are optional. The last section we define the job configuration which completes the macro.

In section 3) we declare the actual job-template. Because of all the preparations above job template definitions should be small and simple. It needs to define the scm and job triggers. The Global JJB project needs to support both Gerrit and GitHub versions of the same job so the job definitions there have 2 templates for each job defined.

2.3.2 Passing parameters to shell scripts

There are 2 ways to pass parameters into scripts:

1. JJB variables in the format {var}
2. Environment variables in the format ${VAR}

We recommend avoiding using method 1 (Pass JJB variables) into shell scripts and instead always use method 2 (Environment variables). This makes troubleshooting JJB errors easier and does not require escaping curly braces.

This method requires 3 steps:

1. Declare a parameter section or inject the variable as properties-content.
2. Invoke the shell script with *include-raw-escape* instead of *include-raw*.
3. Use the shell variable in shell script.

The benefit of this method is that parameters will always be at the top of the job page and when clicking the Build with Parameters button in Jenkins we can see the parameters before running the job. We can review the parameters retro-actively by visiting the job parameters page `job/lastSuccessfulBuild/parameters/`. Injecting variables as properties-content makes the variable local to the specific macro, while declaring it as parameter makes the variable global.

Note: When a macro which invokes a shell script has no JJB parameters defined *!include-raw-escape* will insert extra curly braces, in such cases its recommended to use *!include-raw*.

2.3.3 Shell scripts

When developing shell scripts for JJB we recommend to create shell scripts as a separate file instead of inlining in YAML. This way we can ensure that the ShellCheck linter can catch potential issues with the scripts.

When writing the script itself, we recommend to redeclare all expected inputs at the top of the file using lowercase variable names before setting `set -u` after the inputs section. This ensures that all variables the script expects are at the top of the file which is useful for others to review and debug the script at a later stage. The `set -u` configuration before the start of the script code ensures that we catch any of these undeclared variables at the top of the file.

Example:

```
#!/bin/bash

# Inputs
tox_dir="${TOX_DIR:-$WORKSPACE}"
tox_envs="${TOX_ENVS:-}"

# Script start
set -eux -o pipefail

# ... script code goes here
```

2.3.4 Usage of config-file-provider

When using the config-file-provider plugin in Jenkins to provide a config file. We recommend using a macro so that we can configure the builder to remove the config file as a last step. This ensures that credentials do not exist on the system for longer than it needs to.

ship-logs example:

```
- builder:
    name: lf-ship-logs
    builders:
      - config-file-provider:
          files:
            - file-id: jenkins-log-archives-settings
              variable: SETTINGS_FILE
      - shell: !include-raw:
          - ../shell/logs-get-credentials.sh
      - shell: !include-raw:
          - ../shell/logs-deploy.sh
      - shell: !include-raw:
          - ../shell/logs-clear-credentials.sh
      - description-setter:
          regexp: '^Build logs: .*'
```

In this example the script logs-deploy requires a config file to authenticate with Nexus to push logs up. We declare a macro here so that we can ensure that we remove credentials from the system after the scripts complete running via the logs-clear-credentials.sh script. This script contains 3 basic steps:

1. Provide credentials via config-file-provider
2. Run logs-deploy.sh
3. Remove credentials provided by config-file-provider

2.3.5 Preserving Objects in Variable References

JJB has an option to preserve a data structure object when you want to pass it to a template. https://jenkins-job-builder.readthedocs.io/en/latest/definition.html#variable-references

One thing that is not explicitly covered is the format of the variable name that you pass the object to. When you use the *{obj:key}* notation to preserve the original data structure object, it will not work if the variable name has a dash - in it. The standard that we follow, and recommend, is to use an underscore _ instead of a dash.

Example:

```
.. literalinclude:: _static/github-pr-trigger.example
```

In the above example note the use of underscores in `github_pr_allowlist`, `github_pr_admin_list`, and `github_included_regions`.

2.3.6 Using single quotes around variables

Its recommended to use single quotes around JJB variables '{variable}-field' during variable substitution or when using a variable in a string field, in other cases its recommended to drop the single quotes.

Example:

```
- builder:
    name: lf-user-logs
    builders:
      - inject:
          properties-content: |
              'HOME={user-home}'
      - build-file:
          settings: '{settings-file}'
          file-version: '{file-version}'
```

2.3.7 Variable expansion and Defaults

JJB has a concept called Defaults which is what JJB will replace a variable with if unset. Variables can configure dynamic content in job-template sections and allow certain options in these sections to be configurable.

The section that expands Defaults is Job Templates no other sections will expand a default. This documentation will explain how variables and defaults expansion works and which take precedence in JJB's variable expansion logic for the following configuration sections.

- macro
- job-template
- project
- default

Macro sections

Macro sections can contain variables but do **NOT** support default values getting filled in both at the macro definition level and at the defaults configuration level. Macros and Job Templates can use Macros but any variables defined in a Macro needs to pass a value or a new variable redefined in the Job Template if you want to pass on the configuration.

So for example if you have a macro that has a '{msg}' variable:

Example:

```
- builder:
    name: echo-msg
    builders:
      - shell: "echo {msg}"
```

Any downstream job-templates or macros that use this macro **MUST** pass in a *msg: Hello* definition or redefine the msg variable *msg: {msg}*.

Job Template sections

Job Template sections can use defaults in two ways.

1. Configure the message:

```
- job-template:
    name: echo-hello-world
    builders:
      - echo-msg:
          msg: 'Hello World'
```

2) Re-define '{msg}' variable

```
- job-template:
    name: echo-message
    builders:
      - echo-msg:
          msg: '{message}'
```

In option 2, we redefine the variable msg as *{message}* which a user of the job-template can now pass into the job their own custom message which is different than option 1, where we set a static message to pass in. We purposely redefined the **{msg}** to **{message}** here to show that you do not need to redefine it with the same name but we could have used the same name *{msg}* in the template too if we wanted to keep it the same.

Job Templates can also default a default variable for the variables it defines.

Example:

```
- job-template:
  name: echo-message
  message: 'Hello World'
  builders:
    - echo-msg:
        msg: '{message}'
```

This creates a job template variable called '{message}' which will default to "Hello World" if the user of the template does not explicitly pass in a message.

We should be aware of 2 Defaults concepts:

1. Default as defined in the job-template
2. Default as defined in a defaults configuration (typically defaults.yaml)

In this case there is a default '{message}' set in the job-template. JJB will use this default if the user (project section) does not declare a {message}.

If we do not declare a default in the job-template then JJB will fallback to checking the "defaults configuration".

This means that the precedence of defaults is as follows:

1. User-provided
2. Job Template
3. Defaults.yaml

Project sections

Project sections define real jobs and pass in variables as necessary. Projects sections do NOT expand defaults.yaml. So you cannot configure a setting with {var} in here and expect defaults.yaml to fill it in for you. Define required configuration here.

Example:

```
- project
    name: foo
    jobs:
      - 'echo-message'
    message: 'I am foo'
```

Defaults sections

Defaults sections are the absolute last thing JJB checks if a variable is not configured in a job-template and user did not pass in the variable. JJB will fill in whatever is in the defaults configuration.

Variable expansion order of precedence seems to be:

1. job-group section definition
2. project section definition
3. job-template variable definition
4. defaults.yaml variable definition

Note: Defaults set variables in job-templates and are NOT used in Macros.

---

global-jjb should not provide job-group definitions and leave it up to users of global-jjb to create their own as a job-group as a variable defined in a job group the highest precedence. Global JJB should strive to be purely a job-template and macro library for downstream consumers.

Final thoughts

For any Basic Job Configuration for example "concurrent", "jdk", "node" etc… we cannot set defaults with the same name as JJB will not expand them. To use "node" we need to give the variable for that setting a different name such as "build-node" instead, if we want JJB to perform expansion for those settings. This issue affects top level job configuration, it does not appear to affect items below the top level such as calling a builder, wrapper or parameter.

2.4 Glossary

ciman Short for *ci-management*.

ci-management Refers to the SCM repository containing the *JJB* configuration files. In most LF Projects this is the repository named `ci-management`, but is `releng/builder` in the OpenDaylight project and `releng` in the OPNFV project.

JJB Short for Jenkins Job Builder (JJB) a tool used to convert YAML definitions into XML as a way to define Jenkins job configuration.

2.5 Appendix

2.5.1 ShellCheck

When using ShellCheck to lint global-jjb or any projects that include global-jjb as part of their project (common with ci-management repos) then we require version 0.4.x of ShellCheck installed on the build vms. This version introduces annotations used by shell scripts in this repo.

CHAPTER

THREE

GLOBAL JJB TEMPLATES

Job template code is in the `jjb/` directory but documentation is in the `docs/jjb/` directory of this project.

3.1 C/C++ Jobs

3.1.1 Job Templates

Autotools PackageCloud Stage

Stage job which runs configure && make, then uploads all DEB/RPM files in the build directory to PackageCloud.io. Triggered by comment.

The default configuration supplies a pre-build script that runs GNU Autotools to generate the configure shell script. Must be overridden if that script is in the version-control system.

The Jenkins system must have a configuration file provider that installs files ".packagecloud" and "packagecloud_api" to the Jenkins home directory with appropriate credentials.

The Jenkins build minion must have the Ruby gem "package_cloud" installed.

Template Names

- {project-name}-autotools-packagecloud-stage-{stream}
- gerrit-autotools-packagecloud-stage
- github-autotools-packagecloud-stage

Comment Trigger stage-release

Required parameters

build-node The node to run build on.

debian-distribution-versions list of DEB package distro/version strings separated by space; example: "ubuntu/bionic debian/stretch"

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

packagecloud-account PackageCloud account ID; example: oran

packagecloud-repo PackageCloud repository; example: master, staging

project The git repository name.

project-name Prefix used to name jobs.

rpm-distribution-versions list of RPM package distro/version strings separated by space; example: "el/4 el/5"

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-dir Directory with package files (default: $WORKSPACE)

build-timeout Timeout in minutes before aborting build. (default: 15)

configure-opts Parameters to pass to configure. (default: '')

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install Install build products to /usr/local. (default: false)

install-prefix Path to use for install. (default: $BUILD_DIR/output)

make-opts Parameters and targets for make. (default: '')

pre-build Shell script to generate the configure file and install dependencies. (default: 'autoreconf –install')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

Autotools SonarQube

The SonarQube job installs the SonarQube CXX build wrapper, runs configure, uses the build wrapper to invoke make, then runs the SonarQube Scanner Jenkins plug-in to analyze code for bugs, code smells and security vulnerabilities, and to upload the result (possibly including code-coverage statistics) to a SonarQube server or to SonarCloud.io. Optionally runs a shell script before the build to install prerequisites.

The default configuration supplies a pre-build script that runs GNU Autotools to generate the configure shell script. Must be overridden if that script is in the version-control system.

Requires `SonarQube Scanner for Jenkins`

This job runs on the master branch because the basic Sonar configuration does not support multi-branch.

Plug-in configurations

Manage Jenkins –> Configure System –> SonarQube servers

- Name: Sonar (fixed)
- Server URL: https://sonar.project.org/ or https://sonarcloud.io
- Server authentication token: none for local, API token (saved as a "secret text" credential) for Sonarcloud

Manage Jenkins –> Global Tool Configuration –> SonarQube Scanner

- Name: SonarQube Scanner (fixed)
- Install automatically

- Select latest version

Template Names

- {project-name}-autotools-sonarqube
- gerrit-autotools-sonarqube
- github-autotools-sonarqube

Comment Trigger `run-sonar`

Required parameters

build-node The node to run the build on. (Commonly in defaults.yaml)

jenkins-ssh-credential Credential to use for SSH. (Commonly in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional Parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

build-wrap-dir Build wrapper output subdirectory name. (default: $WORKSPACE/bw-output)

configure-opts Parameters to pass to configure. (default: '')

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: @weekly)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install-prefix Path to use for install. (default: $BUILD_DIR/output)

make-opts Parameters and targets for make. (default: '')

pre-build Shell script to generate the configure file and install dependencies. (default: 'autoreconf –install')

sonar-additional-args Command line arguments. (default: '')

sonar-java-opts JVM options. For example, use option -Xmx to increase the memory size limit. (default: '')

sonar-project-file The file name with Sonar configuration properties (default: sonar-project.properties)

sonar-properties Sonar configuration properties. (default: '')

sonar-task Sonar task to run. (default: '')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

Note: A job definition must provide one of the optional parameters `sonar-project-file` and `sonar-properties`; they cannot both be empty. Set Sonar properties directly in the job definition by setting the `sonar-project-file` property to "" and adding all properties under `sonar-properties`.

Required Sonar Properties

- sonar.login: The API token for authentication at SonarCloud. Commonly defined as key "sonarcloud_api_token" in defaults.yaml.
- sonar.organization: The umbrella project name; e.g., "opendaylight". Commonly defined as key "sonarcloud_project_organization" in defaults.yaml.
- sonar.projectName: The git repository name without slashes; e.g., "infrautils".
- sonar.projectKey: The globally unique key for the report in SonarCloud. Most teams use the catenation of sonar.organization, an underscore, and sonar.projectName; e.g., "opendaylight_infrautils".

Optional Sonar Properties

- sonar.cfamily.gcov.reportsPath: directory with GCOV output files
- Documentation of SonarQube properties is here: https://docs.sonarqube.org/latest/analysis/overview/

Example job definition

The following example defines a job for a project with CXX source files in the "src" directory, and unit tests that write coverage files in GCOV format to the "test" directory. This definition uses configuration parameters in the umbrella project's defaults.yaml file.

```
- project:
    name: my-project-sonar
    project: my/project
    project-name: my-project
    sonar-project-file: ""
    sonar-properties: |
        sonar.login={sonarcloud_api_token}
        sonar.projectKey={sonarcloud_project_organization}_{project-name}
        sonar.projectName={project-name}
        sonar.organization={sonarcloud_project_organization}
        sonar.sourceEncoding=UTF-8
        sonar.sources=src
        sonar.cfamily.build-wrapper-output=$WORKSPACE/bw-output
        sonar.cfamily.gcov.reportsPath=test
    jobs:
      - gerrit-autotools-sonarqube
```

Autotools Verify

Verify job which runs configure && make to test a project build, then optionally runs make install, copies the build products to /usr/local and runs ldconfig to make the shared lib(s) available. The install steps run by default, see parameter "install".

The default configuration supplies a pre-build script that runs GNU Autotools to generate the configure shell script. Must be overridden if the script is in the version-control system.

Template Names

- {project-name}-autotools-verify-{stream}
- gerrit-autotools-verify
- github-autotools-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

configure-opts Parameters to pass to configure. (default: '')

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install Install build products to /usr/local. (default: true)

install-prefix Path to use for install. (default: $BUILD_DIR/output)

make-opts Parameters and targets for make. (default: 'test')

pre-build Shell script to generate the configure file and install dependencies. (default: 'autoreconf –install')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths which to filter which file modifications will trigger a build.

CMake Sonar

The Sonar job installs the SonarQube CXX build wrapper and scanner tools, runs cmake, uses the build wrapper to invoke make, runs the scanner to analyze code files, then publishes the results to a SonarQube server like SonarCloud. Optionally runs a shell script before the build to install prerequisites. Does not support code coverage reporting.

Deprecated; new projects should use a CMake SonarQube template.

This job purposely runs on the master branch because the basic SonarCloud configuration does not support multi-branch.

Template Names

- {project-name}-cmake-sonar
- gerrit-cmake-sonar
- github-cmake-sonar

Comment Trigger run-sonar

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

sonarcloud-organization SonarCloud project organization.

sonarcloud-project-key SonarCloud project key.

Optional parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

cmake-opts Parameters to pass to cmake. (default: '')

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: '@daily')

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install-prefix CMAKE_INSTALL_PREFIX to use for install. (default: $BUILD_DIR/output)

make-opts Parameters to pass to make. (default: '')

pre-build Shell script to run before performing build. Useful for setting up dependencies. (default: '')

sonar-scanner-version Version of sonar-scanner to install. (see YAML for default value; e.g., 3.3.0.1492)

sonarcloud-api-token-cred-id Jenkins credential ID which has the SonarCloud API Token. This one SHOULDN'T be overwritten as per we are standarizing the credential ID for all projects (default: 'sonarcloud-api-token')

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_sonar_triggers Override Gerrit Triggers.

CMake SonarQube

The SonarQube job installs the SonarQube CXX build wrapper, runs cmake, uses the build wrapper to invoke make, then runs the SonarQube Scanner Jenkins plug-in to analyze code for bugs, code smells and security vulnerabilities, and to upload the result (possibly including code-coverage statistics) to a SonarQube server or to SonarCloud.io. Optionally runs a shell script before the build to install prerequisites.

Requires `SonarQube Scanner for Jenkins`

This job runs on the master branch because the basic Sonar configuration does not support multi-branch.

Plug-in configurations

Manage Jenkins –> Configure System –> SonarQube servers

- Name: Sonar (fixed)
- Server URL: https://sonar.project.org/ or https://sonarcloud.io
- Server authentication token: none for local, API token (saved as a "secret text" credential) for Sonarcloud

Manage Jenkins –> Global Tool Configuration –> SonarQube Scanner

- Name: SonarQube Scanner (fixed)
- Install automatically
- Select latest version

Template Names

- {project-name}-cmake-sonarqube
- gerrit-cmake-sonarqube
- github-cmake-sonarqube

Comment Trigger `run-sonar`

Required parameters

build-node The node to run the build on. (Commonly in defaults.yaml)

jenkins-ssh-credential Credential to use for SSH. (Commonly in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional Parameters

archive-artifacts Pattern for files to archive to the logs server (default: '**/*.log')

build-wrap-dir Build wrapper output subdirectory name. (default: $WORKSPACE/bw-output)

cmake-opts Parameters to pass to cmake. (default: '')

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: @weekly)

install-prefix CMAKE_INSTALL_PREFIX to use for install. (default: $BUILD_DIR/output)

make-opts Parameters to pass to make. (default: '')

pre-build Shell script to run before performing build. Useful for setting up dependencies. (default: '')

sonar-additional-args Command line arguments. (default: '')

sonar-java-opts JVM options. For example, use option -Xmx to increase the memory size limit. (default: '')

sonar-project-file The file name with Sonar configuration properties (default: sonar-project.properties)

sonar-properties Sonar configuration properties. (default: '')

sonar-task Sonar task to run. (default: '')

Note: A job definition must provide one of the optional parameters `sonar-project-file` and `sonar-properties`; they cannot both be empty. Set Sonar properties directly in the job definition by setting the `sonar-project-file` property to "" and adding all properties under `sonar-properties`.

Required Sonar Properties

- sonar.login: The API token for authentication at SonarCloud. Commonly defined as key "sonarcloud_api_token" in defaults.yaml.
- sonar.organization: The umbrella project name; e.g., "opendaylight". Commonly defined as key "sonarcloud_project_organization" in defaults.yaml.
- sonar.projectName: The git repository name without slashes; e.g., "infrautils".
- sonar.projectKey: The globally unique key for the report in SonarCloud. Most teams use the catenation of sonar.organization, an underscore, and sonar.projectName; e.g., "opendaylight_infrautils".

Optional Sonar Properties

- sonar.cfamily.gcov.reportsPath: directory with GCOV output files
- Documentation of SonarQube properties is here: https://docs.sonarqube.org/latest/analysis/overview/

Example job definition

The following example defines a job for a project with CXX source files in the "src" directory, and unit tests that write coverage files in GCOV format to the "test" directory. This definition uses configuration parameters in the umbrella project's defaults.yaml file.

```
- project:
    name: my-project-sonar
    project: my/project
    project-name: my-project
    sonar-project-file: ""
    sonar-properties: |
        sonar.login={sonarcloud_api_token}
        sonar.projectKey={sonarcloud_project_organization}_{project-name}
        sonar.projectName={project-name}
        sonar.organization={sonarcloud_project_organization}
        sonar.sourceEncoding=UTF-8
```

(continues on next page)

(continued from previous page)

```
        sonar.sources=src
        sonar.cfamily.build-wrapper-output=$WORKSPACE/bw-output
        sonar.cfamily.gcov.reportsPath=test
    jobs:
      - gerrit-cmake-sonarqube
```

CMake Stage

Stage job which runs cmake && make && make install and then packages the project into a tar.xz tarball to produce a release candidate.

Template Names

- {project-name}-cmake-stage-{stream}
- gerrit-cmake-stage
- github-cmake-stage

Comment Trigger stage-release

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

nexus-group-id The Maven style Group ID for the namespace of the project in Nexus.

staging-profile-id The unique Nexus Staging Profile ID for the project. Contact your infra admin if you do not know it.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-dir Directory to build the project in. (default: $WORKSPACE/target)

build-timeout Timeout in minutes before aborting build. (default: 60)

cmake-opts Parameters to pass to cmake. (default: '')

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install Install build products to /usr/local. (default: true)

install-prefix CMAKE_INSTALL_PREFIX to use for install. (default: $BUILD_DIR/output)

make-opts Parameters to pass to make. (default: '')

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

pre-build Shell script to run before performing build. Useful for setting up dependencies. (default: '')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

version (default: '') Project version to stage release as. There are 2 methods for using this value:

1) Defined explicitly here.

2) Leave this value blank and set /tmp/artifact_version

Use method 2 in conjunction with 'pre-build' configuration to generate the artifact_version automatically from files in the project's repository. An example pre-build script appears below.

```
#!/bin/bash
MAJOR_VERSION="$(grep 'set(OCIO_VERSION_MAJOR' CMakeLists.txt | awk '{{print $NF}}' |␣
↪awk -F')' '{{print $1}}')"
MINOR_VERSION="$(grep 'set(OCIO_VERSION_MINOR' CMakeLists.txt | awk '{{print $NF}}' |␣
↪awk -F')' '{{print $1}}')"
PATCH_VERSION="$(grep 'set(OCIO_VERSION_PATCH' CMakeLists.txt | awk '{{print $NF}}' |␣
↪awk -F')' '{{print $1}}')"
echo "${{MAJOR_VERSION}}.${{MINOR_VERSION}}.${{PATCH_VERSION}}" > /tmp/artifact_version
```

CMake PackageCloud Stage

Stage job which runs cmake && make, then uploads all DEB/RPM files in the build directory to PackageCloud.io. Triggered by comment.

The Jenkins system must have a configuration file provider that installs files ".packagecloud" and "packagecloud_api" to the Jenkins home directory with appropriate credentials.

The Jenkins build minion must have the Ruby gem "package_cloud" installed.

Template Names

- {project-name}-cmake-packagecloud-stage-{stream}
- gerrit-cmake-packagecloud-stage
- github-cmake-packagecloud-stage

Comment Trigger stage-release

Required parameters

build-node The node to run build on.

debian-distribution-versions list of DEB package distro/version strings separated by space; example: "ubuntu/bionic debian/stretch"

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

packagecloud-account PackageCloud account ID; example: oran

packagecloud-repo PackageCloud repository; example: master, staging

project The git repository name.

project-name Prefix used to name jobs.

rpm-distribution-versions list of RPM package distro/version strings separated by space; example: "el/4 el/5"

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-dir Directory to build the project in. (default: $WORKSPACE/build)

build-timeout Timeout in minutes before aborting build. (default: 60)

cmake-opts Parameters to pass to cmake. (default: '')

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install Install build products to /usr/local. (default: false)

install-prefix CMAKE_INSTALL_PREFIX to use for install. (default: $BUILD_DIR/output)

make-opts Parameters to pass to make. (default: '')

pre-build Shell script to run before performing build. Useful for setting up dependencies. (default: '')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

CMake Verify

Verify job which runs cmake && make to test a project build, then runs make install, copies the build products to /usr/local and runs ldconfig to make the shared lib(s) available. The install steps run by default, see optional parameter "install".

Template Names

- {project-name}-cmake-verify-{stream}
- gerrit-cmake-verify
- github-cmake-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Configure in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-dir Directory to build the project in. (default: $WORKSPACE/target)

build-timeout Timeout in minutes before aborting build. (default: 60)

cmake-opts Parameters to pass to cmake. (default: '')

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

install Install build products to /usr/local. (default: true)

install-prefix CMAKE_INSTALL_PREFIX to use for install. (default: $BUILD_DIR/output)

make-opts Parameters to pass to make. (default: '')

pre-build Shell script to run before performing build. Useful for setting up dependencies. (default: '')

stream Keyword that to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths which to filter which file modifications will trigger a build.

3.2 CI Jobs

3.2.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35
```

(continues on next page)

(continued from previous page)

```
- project:
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of CI job groups:

```
---
- job-group:
    name: "{project-name}-ci-jobs"

    jobs:
      - gerrit-jenkins-cfg-merge
      - gerrit-jenkins-cfg-verify
      - gerrit-jenkins-sandbox-cleanup
      - gerrit-jjb-deploy-job
      - gerrit-jjb-merge
      - gerrit-jjb-verify
      - gerrit-branch-lock

- job-group:
    name: "{project-name}-github-ci-jobs"

    jobs:
      - github-jenkins-cfg-merge
      - github-jenkins-cfg-verify
      - github-jenkins-sandbox-cleanup
      - github-jjb-deploy-job
      - github-jjb-merge
      - github-jjb-verify

- job-group:
    name: "{project-name}-info-yaml-jobs"

    jobs:
      - gerrit-info-yaml-verify

- job-group:
    name: "{project-name}-github-info-yaml-jobs"

    jobs:
      - github-info-yaml-verify

- job-group:
    name: "{project-name}-packer-jobs"

    jobs:
```

(continues on next page)

(continued from previous page)

```
      - gerrit-packer-merge
      - gerrit-packer-verify
      - gerrit-packer-verify-build

- job-group:
    name: "{project-name}-github-packer-jobs"

    jobs:
      - github-packer-merge
      - github-packer-verify
      - github-packer-verify-build

- job-group:
    name: "{project-name}-openstack-jobs"

    jobs:
      - gerrit-openstack-update-cloud-image
      - gerrit-openstack-cron

- job-group:
    name: "{project-name}-github-openstack-jobs"

    jobs:
      - github-openstack-update-cloud-image
      - github-openstack-cron
```

3.2.2 Macros

lf-infra-jjb-parameters

Required Parameters

jjb-cache Location of Jenkins Job Builder (JJB) cache used for jjb jobs.

jjb-version Version of Jenkins Job Builder (JJB) to install and use in the jjb jobs.

lf-jenkins-cfg-clouds

Deploys Jenkins Cloud configuration read from the `jenkins-clouds` directory in ci-management repositories.

Note: Requires the jjbini file in Jenkins CFP to contain JJB 2.0 style config definitions for "production" and "sandbox" systems.

Required Parameters

jenkins-silos Space-separated list of Jenkins silos to update configuration for as defined in ~/.config/jenkins_jobs/jenkins_jobs.ini

lf-jenkins-cfg-global-vars

Manages the Global Jenkins variables. This macro will clear all exist macros in Jenkins and replaces them with the ones defined by the ci-management/jenkins-config/global-vars-SILO.sh script.

Note: Requires the jjbini file in Jenkins CFP to contain JJB 2.0 style config definitions for "production" and "sandbox" systems.

Required parameters

jenkins-silos Space-separated list of Jenkins silos to update configuration for as defined in ~/.config/jenkins_jobs/jenkins_jobs.ini

lf-infra-jjbini

Provides jenkins_jobs.ini configuration for Jenkins.

lf-packer-common

Common packer configuration.

lf-packer-file-paths

Gerrit file-paths for packer jobs.

lf-packer-parameters

Parameters useful for packer related tasks.

Parameters

packer-version Version of packer to install / use. (shell: PACKER_VERSION)

lf-packer-verify-file-paths

Gerrit file-paths for packer verify jobs.

lf-puppet-parameters

Parameters useful for Puppet related tasks.

Parameters

puppet-lint-version Version of puppet-lint to install / use. (shell: PUPPET_LINT_VERSION)

3.2.3 Job Templates

Gerrit Branch Lock

Job submits a patch to lock or unlock a project's branch.

This job will process lock/unlock requests for all projects and all branches and does not need to have per-project configuration.

Template Names

- {project-name}-gerrit-branch-lock
- gerrit-branch-lock

Comment Trigger

- lock branch
- unlock branch

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

git-url URL to clone project from. (default: $GIT_URL/$GERRIT_PROJECT)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

gerrit_merge_triggers Override Gerrit Triggers.

Jenkins Configuration Merge

Jenkins job to manage Global Jenkins configuration.

Note: Requires the jjbini file in Jenkins CFP to contain JJB 2.0 style config definitions for "production" and "sandbox" systems.

Template names

- {project-name}-jenkins-cfg-merge
- gerrit-jenkins-cfg-merge
- github-jenkins-cfg-merge

Optional parameters

branch Git branch to build against. (default: master)

cron How often to run the job on a cron schedule. (default: @daily)

git-url URL to clone project from. (default: $GIT_URL/$GERRIT_PROJECT)

jenkins-silos Space separated list of Jenkins silos to update configuration for as defined in ~/.config/jenkins_jobs/jenkins_jobs.ini (default: production sandbox)

Typically this template is automatically pulled in by the "{project-name}-ci-jobs" job-group and does not need to be explicitly called if you are already using the job group.

Minimal Example:

```
---
- project:
    name: jenkins-cfg-merge-minimal-test
    jobs:
      - "gerrit-jenkins-cfg-merge"

    project-name: ci-management
```

Full Example:

```
---
- project:
    name: jenkins-cfg-merge-full-test
    jobs:
      - "gerrit-jenkins-cfg-merge"

    project-name: builder
    jenkins-silos: releng sandbox
```

Global Environment Variables

Manage Global Environment Variables via the `jenkins-config/global-vars-SILO.sh` file in ci-management. Replace SILO with the name of the Jenkins silo the variable configuration is for.

The format for this file is `KEY=value` for example:

```
GERRIT_URL=https://git.opendaylight.org/gerrit
GIT_BASE=git://devvexx.opendaylight.org/mirror/$PROJECT
GIT_URL=git://devvexx.opendaylight.org/mirror
JENKINS_HOSTNAME=vex-yul-odl-jenkins-2
LOGS_SERVER=https://logs.opendaylight.org
NEXUS_URL=https://nexus.opendaylight.org
ODLNEXUSPROXY=https://nexus.opendaylight.org
SILO=sandbox
SONAR_URL=https://sonar.opendaylight.org
```

Cloud Configuration

This configuration requires the **OpenStack Cloud plugin** in Jenkins.

OpenStack Cloud plugin version supported:

- 2.30 - 2.34
- 2.35 - 2.37

Cloud configuration follows a directory structure in ci-management like this:

- jenkins-config/clouds/openstack/
- jenkins-config/clouds/openstack/**cattle**/cloud.cfg

- jenkins-config/clouds/openstack/**cattle**/centos7-builder-2c-2g.cfg
- jenkins-config/clouds/openstack/**cattle**/centos7-builder-4c-4g.cfg
- jenkins-config/clouds/openstack/**cattle**/centos7-docker-4c-4g.cfg

This job uses the directory name of the directory inside of the "openstack" directory as the name of the cloud configuration in Jenkins. This is to support systems that want to use more than one cloud provider. In this example "cattle" is the cloud name.

The `cloud.cfg` file is a special file used to configure the main cloud configuration in the format `KEY=value`.

Cloud Parameters

CLOUD_URL API endpoint URL for Keystone. (default: "")

CLOUD_IGNORE_SSL Ignore unverified SSL certificates. (default: false)

CLOUD_ZONE OpenStack region to use. (default: "")

CLOUD_CREDENTIAL_ID Credential to use for authentication to OpenStack. (default: "os-cloud")

INSTANCE_CAP Total number of instances the cloud will allow spin up. (default: null)

SANDBOX_CAP Total number of instances the cloud will allow to spin up. This applies to "sandbox" systems and overrides the INSTANCE_CAP setting. (default: null)

Template Parameters

Note: Parameters below that are not defined will inherit the ones defined in the default clouds configuration.

IMAGE_NAME The image name to use for this template. (required)

HARDWARE_ID OpenStack flavor to use. (required)

LABELS Labels to assign to the vm. (default: FILE_NAME)

VOLUME_SIZE Volume size to assign to vm. (default: "")

HARDWARE_ID Hardware Id to assign to vm. (default: "")

NETWORK_ID OpenStack network to use. (default: "")

USER_DATA_ID User Data to pass into the instance. (default: jenkins-init-script)

INSTANCE_CAP Total number of instances of this type that is available for use at one time. When defined in clouds.cfg it defines the total for the entire cloud. (default: null)

SANDBOX_CAP Total number of instances of this type that is available for use at one time. When defined in clouds.cfg it defines the total for the entire cloud. This applies to "sandbox" systems and overrides the INSTANCE_CAP setting. (default: null)

FLOATING_IP_POOL Floating ip pool to use. (default: "")

SECURITY_GROUPS Security group to use. (default: "default")

AVAILABILITY_ZONE OpenStack availability zone to use. (default: "")

START_TIMEOUT Number of milliseconds to wait for agent provisioning. (default: 600000)

KEY_PAIR_NAME SSH Public Key Pair to use for authentication. (default: jenkins-ssh)

NUM_EXECUTORS Number of executors to enable for the instance. (default: 1)

JVM_OPTIONS JVM Options to pass to Java. (default: null)

FS_ROOT File system root for the workspace. (default: "/w")

NODE_PROPERTIES Node properties. (default: null)

RETENTION_TIME Number of minutes to wait for an idle minion before removing it from the system. If set to -1, the minion will stick around forever. (default: 0)

CONNECTION_TYPE The connection type for Jenkins to connect to the build minion. Valid options: JNLP, SSH. (default: "SSH")

CONFIG_TYPE Configuration drive. (default: null)

For a live example see the OpenDaylight project jenkins-config directory. https://github.com/opendaylight/releng-builder/tree/master/jenkins-config

Troubleshooting

Cloud Configuration

The directory `groovy-inserts` contains the groovy script output used by Jenkins to push the cloud configuration. In the event of a job failure use this file to debug.

Jenkins Configuration Verify

Jenkins job to verify the Global Jenkins configuration.

Requires the `clouds-yaml` file to be setup on the Jenkins host.

Template names

- {project-name}-jenkins-cfg-verify
- gerrit-jenkins-cfg-verify
- github-jenkins-cfg-verify

Optional parameters

branch Git branch to build against. (default: master)

git-url URL to clone project from. (default: $GIT_URL/$GERRIT_PROJECT)

This job is not part of the "{project-name}-ci-jobs" group and requires separate configuration.

Example:

```
---
- project:
    name: jenkins-cfg-verify
    jobs:
      - "gerrit-jenkins-cfg-verify"

    project-name: ci-management
```

Jenkins Sandbox Cleanup

Cleanup Jenkins Sandbox of jobs and views periodically.

Template names

- {project-name}-jenkins-sandbox-cleanup
- gerrit-jenkins-sandbox-cleanup
- github-jenkins-sandbox-cleanup

Comment Trigger NONE

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

cron Schedule to run job. (default: '0 8 * * 6')

JJB Deploy Job

Deploy jobs to jenkins-sandbox system via code review comment.

This job checks out the current code review patch and then runs a `jenkins-jobs update` to push a patch defined by the comment.

Template names

- {project-name}-jjb-deploy-job
- gerrit-jjb-deploy-job
- github-jjb-deploy-job

Comment Trigger jjb-deploy JOB_NAME

Note: The JJB Deploy Job is a manual job and triggers via Gerrit comment which starts with the `jjb-deploy` keyword.

Example of a valid command in Gerrit comment that triggers the job:

```
jjb-deploy builder-jjb-*
```

Example of a invalid command in Gerrit comment that would _not_ trigger the job:

```
Update the job. jjb-deploy builder-jjb-*
```

JOB_NAME can include the * wildcard character to push jobs matching the pattern. For example `jjb-deploy builder-jjb-*` will push all builder-jjb-* jobs to the sandbox system.

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

gerrit_jjb_deploy_job_triggers Override Gerrit Triggers.

JJB Merge

Runs `jenkins-jobs update` to update production job configuration

Template Names

- {project-name}-jjb-merge
- gerrit-jjb-merge
- github-jjb-merge

Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

jjb-cache JJB cache location. (default: $HOME/.cache/jenkins_jobs)

jjb-workers Number of threads to run **update** with. Set to 0 by default which will use the number of available CPU cores. (default: 0)

jjb-version JJB version to install. (default: see job-template)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build. (default defined by lf_jjb_common)

JJB Verify

Runs `jenkins-jobs test` to verify JJB syntax. Optionally verifies build-node labels used in templates and job definitions.

Template Names

- {project-name}-jjb-verify
- gerrit-jjb-verify
- github-jjb-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-concurrent Set to `true` to allow this job to run jobs simultaneously. (default: true)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-node-label-check Whether to check build-node labels in jobs against node names in cloud config files (default: false)

build-node-label-list Space-separated list of external build-node labels not present in cloud config files (default: "")

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

jjb-cache JJB cache location. (default: $HOME/.cache/jenkins_jobs)

jjb-version JJB version to install. (default: see job-template)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Set to `true` to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

throttle_categories List of categories to throttle by.

throttle-enabled Set to `true` to enable throttling on the job. (default: true)

throttle-max-per-node Max jobs to run on the same node. (default: 1)

throttle-max-total Max jobs to run across the entire project. - 0 means 'unlimited' (default: 0)

throttle-option Throttle by the project or by list of categories defined in the throttle plugin configuration. (options: 'project', 'category'; default: project)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build. (default defined by lf_jjb_common)

JJB Verify Upstream Global JJB

Runs `jenkins-jobs test` to verify JJB syntax for upstream global-jjb patches. This job is useful to notify upstream that they may be breaking project level jobs.

Template Names

- {project-name}-jjb-verify-upstream-gjjb
- gerrit-jjb-verify-upstream-gjjb
- github-jjb-verify-upstream-gjjb

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

jjb-cache JJB cache location. (default: $HOME/.cache/jenkins_jobs)

jjb-version JJB version to install. (default: see job-template)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

Info YAML Verify

This job verifies that `INFO.yaml` file changes follow the schema defined in *lfit/releng-global-jjb/schema/info-schema.yaml*.

The `INFO.yaml` file changes must be independent of any other files changes.

Template Names

- {project-name}-info-yaml-verify
- gerrit-info-yaml-verify
- github-info-yaml-verify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

LF Pipelines Verify

Verify job for the LF RelEng pipeline library.

Requires the Pipelines plugins installed. This job will look for a Gerrit system named "lf-releng" (mapped to https://gerrit.linuxfoundation.org/infra/), and pull in the Jenkinsfile in the root directory of the repo.

Template Names

- lf-pipelines-verify

Comment Trigger recheck|reverify

License Checker

Job to scan projects for files missing license headers.

Template Names

- {project-name}-license-check
- gerrit-license-check
- github-license-check

Optional parameters

build-timeout Timeout in minutes before aborting build. (default: 15)

file-patterns Space-separated list of file patterns to scan. (default: *.go *.groovy *.java *.py *.sh)

spdx-disable Disable the SPDX-Identifier checker. (default: false)

lhc-version Version of LHC to use. (default: 0.2.0)

license-exclude-paths Comma-separated list of paths to exclude from the license checker. Matches the paths defined here using a contains rule, we recommend you to configure as precisely as possible. For example a path of '/src/generated/' will search as '**/src/generated/**'. Example: org/opendaylight/yang/gen,protobuff/messages (default: '')

licenses-allowed Comma-separated list of allowed licenses. (default: Apache-2.0,EPL-1.0,MIT)

project-pattern The ANT based pattern for Gerrit Trigger to choose which projects to trigger job against. (default: '**')

OpenStack Cron

Cron job that runs on a schedule to perform periodic tasks against OpenStack.

This job requires a Config File Provider file named `clouds-yaml` available containing the credentials for the cloud.

Template Names

- {project-name}-openstack-cron
- gerrit-openstack-cron
- github-openstack-cron

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

jenkins-urls URLs to Jenkins systems to check for active builds.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 90)

cron Time when the packer image should be rebuilt (default: @hourly)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

openstack-cloud OS_CLOUD setting to pass to openstack client. (default: vex)

openstack-image-cleanup Set `true` to run the image cleanup script. (default: true)

openstack-image-cleanup-age Age in days of image before marking it for removal. (default: 30)

openstack-image-protect Set `true` to run the image protect script. (default: true)

openstack-server-cleanup Set `true` to run the server cleanup script. (default: true)

openstack-stack-cleanup Set `true` to run the stack cleanup script. (default: true)

openstack-volume-cleanup Set `true` to run the volume cleanup script. (default: true)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

Minimal Example:

```
---
- project:
    name: openstack-cron-minimal-test
    jobs:
      - "gerrit-openstack-cron"

    project-name: ci-management
```

(continues on next page)

(continued from previous page)

```
    jenkins-urls: >
      https://jenkins.example.org
      https://jenkins.example.org/sandbox
```

Full Example:

```
---
- project:
    name: openstack-cron-full-test
    jobs:
      - "gerrit-openstack-cron"

    project-name: ciman-full

    jenkins-urls: >
      https://jenkins.example.org
      https://jenkins.example.org/sandbox
    openstack-cloud: example-cloud
    openstack-image-cleanup: false
    openstack-image-cleanup-age: 42
    openstack-image-protect: false
    openstack-server-cleanup: false
    openstack-stack-cleanup: false
    openstack-volume-cleanup: false
```

OpenStack Update Cloud Image

This job finds and updates OpenStack cloud images on the ci-management source repository.

This job functions in 2 ways:

1. When triggered via packer-merge job, updates the image created by the job.
2. When triggered manually or via cron, updates all images.

When triggered through an upstream packer merge job, this generates a change request for the new image built.

When triggered manually, this job finds the latest images on OpenStack cloud and compares them with the images in use in the source ci-management source repository. If the compared images have newer time stamps are **all** updated through a change request.

This job requires a Jenkins configuration merge and verify job setup and working on Jenkins.

Template Names

- {project-name}-openstack-update-cloud-image
- gerrit-openstack-update-cloud-image
- github-openstack-update-cloud-image

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

new-image-name Name of new image name passed from packer merge job or set to 'all' to update all images. (default: all)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 90)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

openstack-cloud OS_CLOUD setting to pass to openstack client. (default: vex)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

update-cloud-image Submit a change request to update new built cloud image to Jenkins. (default: false)

Minimal Example:

```
---
- project:
    name: openstack-update-cloud-images-minimal-test
    jobs:
      - "gerrit-openstack-update-cloud-image"

    project-name: ciman-minimal
    gerrit-user: "jenkins-user"
    gerrit-host: "git.example.org"
    gerrit-topic: "update-cloud-image"
    reviewers-email: "jenkins-user@example.org"
```

Full Example:

```
---
- project:
    name: openstack-update-cloud-images-full-test
    jobs:
      - "gerrit-openstack-update-cloud-image"

    project: ciman
    project-name: ciman-full
    build-timeout: 10
    branch: master
    archive-artifacts: "**/*.log"
    jenkins-ssh-credential: "jenkins-ssh-credential"
    gerrit-user: "jenkins-user"
    gerrit-host: "git.example.org"
    gerrit-topic: "update-cloud-image"
    reviewers-email: "jenkins-user@example.org"
```

Packer Merge

Packer Merge job runs *packer build* to build system images in the cloud.

This job requires a Config File Provider file named `ansible-cfg` created on Jenkins. The file can include ansible host configuration required for the environment.

Template Names

- {project-name}-packer-merge-{platforms}-{templates}
- gerrit-packer-merge
- github-packer-merge

Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

platforms Platform or distribution to build. Typically json file found in the packer/vars directory. (Example: centos-7)

templates System template to build. Typically a yaml file or shell script found in the packer/provision directory. (Example: docker)

Optional parameters

cron Time when the packer image should be rebuilt (default: @monthly)

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 90)

gerrit_verify_triggers Override Gerrit Triggers.

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

openstack Packer template uses an OpenStack builder (default: true).

openstack-cloud Sets OS_CLOUD variable to the value of this parameter. (default: vex).

packer-cloud-settings Name of settings file containing credentials for the cloud that packer will build on. (default: packer-cloud-env)

packer-version Version of packer to install / use in build. (default: 1.0.2)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

update-cloud-image Submit a change request to update newly built cloud image to Jenkins. (default: false)

Test an in-progress patch

To test an in-progress patch from a GitHub Pull Request, upload this job to the Jenkins Sandbox. Then when manually building the job, replace the GERRIT_REFSPEC parameter with the GitHub Pull Request number of the patch you would like to test.

Example GitHub:

```
GERRIT_REFSPEC: origin/pr/49/merge
```

Packer Verify

Packer Verify job runs `packer validate` to verify packer configuration. The verify job checks superficial syntax of the template and other files. It does not attempt to build an image, and cannot detect all possible build issues.

This job requires a Config File Provider file named `ansible-cfg` created on Jenkins. The file can include ansible host configuration required for the environment.

Template Names

- {project-name}-packer-verify
- gerrit-packer-verify
- github-packer-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

gerrit_verify_triggers Override Gerrit Triggers.

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

openstack Packer template uses an OpenStack builder (default: true).

openstack-cloud Sets OS_CLOUD variable to the value of this parameter. (default: vex).

packer-cloud-settings Name of settings file containing credentials for the cloud that packer will build on. (default: packer-cloud-env)

packer-version Version of packer to install / use in build. (default: 1.0.2)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

Packer Verify Build

Packer Verify Build job is essentially the same as the *Packer Merge job*. Trigger using its keyword, and will build a useable image. If the last patch set before a merge has a successful verify build, the merge job will not build the same image.

Template Names

- {project-name}-packer-verify-build-{platforms}-{templates}
- gerrit-packer-verify-build
- github-packer-verify-build

Comment Trigger verify-build|packer-build

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

platforms Platform or distribution to build. Typically json file found in the packer/vars directory. (Example: centos-7)

templates System template to build. Typically a yaml file or shell script found in the packer/provision directory. (Example: docker)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

gerrit_verify_triggers Override Gerrit Triggers.

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

openstack Packer template uses an OpenStack builder (default: true).

openstack-cloud Sets OS_CLOUD variable to the value of this parameter. (default: vex).

packer-cloud-settings Name of settings file containing credentials for the cloud that packer will build on. (default: packer-cloud-env)

packer-version Version of packer to install / use in build. (default: 1.0.2)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

update-cloud-image Submit a change request to update new built cloud image to Jenkins. (default: false)

Puppet Verify

Runs puppet-lint in the `puppet-dir` directory. Since `puppet-lint` runs recursively, we recommend to run from the base directory.

Template Names

- {project-name}-puppet-verify
- gerrit-puppet-verify
- github-puppet-verify

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

gerrit_trigger_file_paths Override file paths which used to filter which file modifications will trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

git-url URL clone project from. (default: $GIT_URL/$GERRIT_PROJECT)

puppet-dir Directory containing the project's puppet module(s) relative to the workspace. (default: '')

puppet-lint-version Version of puppet-lint to use for testing. (default: 2.3.6)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

Sonar

Runs the Jenkins SonarQube Scanner plug-in to analyze code for bugs, code smells and security vulnerabilities, and to upload the result (possibly including code-coverage statistics) to a SonarQube server or to SonarCloud.io.

Requires `SonarQube Scanner for Jenkins`

Configuration must set one of the parameters `sonar-project-file` or `sonar-properties`; they cannot both be empty.

Plug-in configurations

Manage Jenkins –> Configure System –> SonarQube servers

- Name: Sonar (fixed)
- Server URL: https://sonar.server.org/ or https://sonarcloud.io
- Server authentication token: none for local, API token (saved as a "secret text" credential) for Sonarcloud

Manage Jenkins –> Global Tool Configuration –> SonarQube Scanner

- Name: SonarQube Scanner (fixed)
- Install automatically
- Select latest version

Note: Optionally, set Sonar properties directly in the job definition by setting the sonar-project-file to "" and adding all properties under `sonar-properties`.

Template Names

- {project-name}-sonar
- gerrit-sonar
- github-sonar

Optional Parameters

sonar-task Sonar task to run. (default: "")

sonar-project-file The filename for the project's properties (default: "sonar-project.properties")

sonar-properties Sonar configuration properties. (default: "")

sonar-java-opts JVM options. (default: "")

sonar-additional-args Additional command line arguments. (default: "")

sonarcloud-java-version Version of Java to run the Sonar scan. (default: "openjdk11")

Sonar with Prescan

The same as the Sonar job above, except the caller also defines a builder called `lf-sonar-prescan`, in which they can put any builders that they want to run before the Sonar scan.

```
- builder:
    name: lf-sonar-prescan
    builders:
      - shell: "# Pre-scan shell script"
```

Template Names

- {project-name}-sonar-prescan
- gerrit-sonar-prescan
- github-sonar-prescan

Required Parameters

lf-sonar-prescan A builder that will run before the Sonar scan.

Optional Parameters

sonar-task Sonar task to run. (default: "")

sonar-project-file The filename for the project's properties (default: "sonar-project.properties")

sonar-properties Sonar configuration properties. (default: "")

sonar-java-opts JVM options. (default: "")

sonar-additional-args Additional command line arguments. (default: "")

sonarcloud-java-version Version of Java to run the Sonar scan. (default: "openjdk11")

Sonar with Prescan Script

The same as the Sonar job above, except the caller must supply a shell script to run before the Sonar scan. This is commonly used to install prerequisites, build the project, execute unit tests and generate a code-coverage report.

Template Names

- {project-name}-sonar-prescan-script
- gerrit-sonar-prescan-script
- github-sonar-prescan-script

Required Parameters

sonar-prescan-script A shell script that will run before the Sonar scan.

Optional Parameters

sonar-task Sonar task to run. (default: "")

sonar-project-file The filename for the project's properties. (default: "sonar-project.properties")

sonar-properties Sonar configuration properties. (default: "")

sonar-java-opts JVM options. (default: "")

sonar-additional-args Additional command line arguments. (default: "")

sonarcloud-java-version Version of Java to run the Sonar scan. (default: "openjdk11")

3.3 Docker Jobs

3.3.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
```

(continues on next page)

(continued from previous page)

```
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35

- project:
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of Docker job groups:

```
---
- job-group:
    name: "{project-name}-gerrit-docker-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for any docker project ci.

    jobs:
      - gerrit-docker-verify
      - gerrit-docker-merge

- job-group:
    name: "{project-name}-github-docker-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for any docker project ci.

    jobs:
      - github-docker-verify
      - github-docker-merge
```

3.3.2 Macros

lf-docker-get-container-tag

Chooses a tag to label the container image based on the 'container-tag-method' parameter using the global-jjb script docker-get-container-tag.sh. Use one of the following methods:

If `container-tag-method: latest`, uses the literal string `latest`.

If `container-tag-method: stream`, uses the value of the variable `stream`.

If `container-tag-method: git-describe`, reads the tag from the `git describe` command on the repository, which requires that the repository has a git tag. For example, if the most recent tag is 'v0.48.1', this method yields a string like 'v0.48.1' or 'v0.48.1-25-gaee2dcb'.

If `container-tag-method: yaml-file`, reads the tag from the YAML file `container-tag.yaml` in the docker-root directory using the top-level entry 'tag'. Alternately specify the directory with the YAML file in parameter 'container-tag-yaml-dir'. An example file appears next.

Example container-tag.yaml file:

```
---
tag: 1.0.0
```

Optionally, teams can supply their own script to choose the docker tag. Pass the shell script path in optional configuration parameter 'docker-get-container-tag-script' which by default is the path to file docker-get-container-tag.sh. The script must create the file 'env_docker_inject.txt' in the workspace with a line that assigns a value to shell variable DOCKER_IMAGE_TAG, as shown next.

Example env_docker_inject.txt file:

```
DOCKER_IMAGE_TAG=1.0.0
```

lf-docker-build

Calls docker build to build the container.

lf-docker-push

Calls docker-push.sh script to push docker images.

3.3.3 Job Templates

Docker Verify

Executes a docker build task to verify an test image build and discards the test image upon completion.

Template Names

- {project-name}-docker-verify-{stream}
- gerrit-docker-verify
- github-docker-verify

Comment Trigger recheck|reverify post a comment with one of the triggers to launch this job manually. Do not include any other text or vote in the same comment.

Required parameters

build-node The node to run build on.

container-public-registry Docker registry source with base images.

docker-name Name of the Docker image.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings Maven settings.xml file containing Docker credentials.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

container-tag-method Specifies the docker tag-choosing method. Options are "latest", "git-describe" or "yaml-file". Option latest uses the "latest" tag. Option git-describe uses the string returned by git-describe, which requires a tag to exist in the repository. Option yaml-file uses the string from file "container-tag.yaml" in the repository. (default: latest)

container-tag-yaml-dir Directory with container-tag.yaml. (default: $DOCKER_ROOT)

docker-build-args Arguments for the docker build command.

docker-get-container-tag-script Path to script that chooses docker tag. (default: ../shell/docker-get-container-tag.sh in global-jjb)

docker-root Build directory within the repo. (default: $WORKSPACE, the repo root)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre_docker_build_script Build script to execute before the main verify builder steps. (default: "")

post_docker_build_script Build script to execute after the main verify builder steps. (default: "")

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override Gerrit file paths to filter which file modifications will trigger a build.

github_included_regions Override Github file paths to filter which file modifications will trigger a build; must match parameter gerrit_trigger_file_paths

container-tag.yaml example:

```
---
tag: 1.0.0
```

Docker Merge

Executes a docker build task and pushes the resulting image to the specified Docker registry. If every image is a release candidate, this should use a staging repository and occassionally run to check dependencies.

Template Names

- {project-name}-docker-merge-{stream}
- gerrit-docker-merge
- github-docker-merge

Comment Trigger remerge post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required parameters

build-node The node to run build on.

container-public-registry Docker registry source with base images.

container-push-registry Docker registry target for the push action.

docker-name Name of the Docker image.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings Maven settings.xml file containing Docker credentials.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

container-tag-method Specifies the docker tag-choosing method. Options are "latest", "git-describe" or "yaml-file". Option latest uses the "latest" tag. Option git-describe uses the string returned by git-describe, which requires a tag to exist in the repository. Option yaml-file uses the string from file "container-tag.yaml" in the repository. (default: latest)

container-tag-yaml-dir Directory with container-tag.yaml. (default: $DOCKER_ROOT)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. Use '@daily' to run daily or '@weekly' to run weekly. (default: @weekly)

docker-build-args Arguments for the docker build command.

docker-get-container-tag-script Path to script that chooses docker tag. (default: ../shell/docker-get-container-tag.sh in global-jjb)

docker-root Build directory within the repo. (default: $WORKSPACE, the repo root)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre_docker_build_script Build script to execute before the main merge builder steps. (default: "")

post_docker_build_script Build script to execute after the main merge builder steps. (default: "")

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override Gerrit file paths to filter which file modifications will trigger a build.

github_included_regions Override GitHub file paths to filter which file modifications will trigger a build; must match parameter gerrit_trigger_file_paths

Sample container-tag.yaml File

```
---
tag: 1.0.0
```

Docker Snyk CLI

Builds the code, downloads and runs a Snyk CLI scan of the code into the Snyk dashboard.

Template Names

- {project-name}-docker-snyk-cli-{stream}
- gerrit-docker-snyk-cli
- github-docker-snyk-cli

Comment Trigger run-snyk

Required parameters

build-node The node to run build on.

container-public-registry Docker registry source with base images.

docker-name Name of the Docker image.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings Maven settings.xml file containing Docker credentials.

snyk-token-credential-id Snyk API token to communicate with Jenkins.

snyk-org-credential-id Snyk organization ID.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

container-tag-method Specifies the docker tag-choosing method. Options are “latest”, “git-describe” or “yaml-file”. Option latest uses the “latest” tag. Option git-describe uses the string returned by git-describe, which requires a tag to exist in the repository. Option yaml-file uses the string from file “container-tag.yaml” in the repository. (default: latest)

container-tag-yaml-dir Directory with container-tag.yaml. (default: $DOCKER_ROOT)

docker-build-args Arguments for the docker build command.

docker-get-container-tag-script Path to script that chooses docker tag. (default: ../shell/docker-get-container-tag.sh in global-jjb)

docker-root Build directory within the repo. (default: $WORKSPACE, the repo root)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre_docker_build_script Build script to execute before the main verify builder steps. (default: "")

post_docker_build_script Build script to execute after the main verify builder steps. (default: "")

snyk-cli-options Snyk CLI options. (default: '')

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

gerrit_snyk_triggers Override Gerrit Triggers.

3.4 Go Jobs

3.4.1 Macros

lf-go-test

Calls go-test.sh script against a Go project.

Required Parameters

go-root Path to the Go project root directory.

lf-go-common

Common Jenkins configuration for Go jobs.

3.4.2 Job Templates

Go SNYK CLI

Builds the code, downloads and runs a Snyk CLI scan of the code into the Snyk dashboard.

Template Names

- {project-name}-go-snyk-cli-{stream}
- gerrit-go-snyk-cli
- github-go-snyk-cli

Comment Trigger run-snyk

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

snyk-token-credential-id Snyk API token to communicate with Jenkins.

snyk-org-credential-id Snyk organization ID.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

snyk-cli-options Snyk CLI options. (default: '')

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_snyk_triggers Override Gerrit Triggers.

Go Verify

Job which runs go test ./... to verify a Go project. 'go test ./...' runs unit tests on current folder and all subfolders.

Template Names

- {project-name}-go-verify-{stream}"
- gerrit-go-verify
- github-go-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

3.5 Gradle Jobs

3.5.1 Job Templates

Gradle Build

Runs a gradle build command to perform the verification.

Template Names

- {project-name}-gradle-build-{stream}

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

java-version Version of Java to execute Maven build. (default: openjdk17)

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings Maven settings.xml file containing credentials to use.

wrapper Use the gradle wrapper (default: false)

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

deploy-path The path in Nexus to deploy javadoc to. (default: $PROJECT/$STREAM)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

3.6 Info Vote Job

Job counts the votes from the committers against a change to the INFO.yaml file

Auto-merges the change on a majority vote.

3.6.1 info-vote

Comment Trigger recheck|reverify|Vote

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

3.7 Global Macros

3.7.1 Builders

comment-to-gerrit

This macro will post a comment to the gerrit patchset if the build creates a file named gerrit_comment.txt To use this macro add it to the list of builders.

lf-ansible-config-file-provider

Provisions files required by the ansible, namely "~/.ansible/ansible.cfg" in the Jenkins home directory.

lf-fetch-dependent-patches

Fetch all patches provided via comment trigger

This macro will fetch all patches provided via comment trigger and will create a list of projects from those patches via environment variable called `DEPENDENCY_BUILD_ORDER` to build projects in the specified order. Order calculated by the first patch instance for a project in the patch list.

lf-license-check

Checks files for

Required parameters

file-patterns Space-separated list of file patterns to scan. For example: *.go *.groovy *.java *.py *.sh

spdx-disable Disable the SPDX-Identifier checker.

lhc-version Version of LHC to use.

license-exclude-paths Comma-separated list of paths to exclude from the license checker. Matches the paths defined here using a contains rule, we recommend you to configure as precisely as possible. For example a path of '/src/generated/' will search as '**/src/generated/**'. Example: org/opendaylight/yang/gen,protobuff/messages

licenses-allowed Comma-separated list of allowed licenses. For example: Apache-2.0,EPL-1.0,MIT

lf-infra-capture-instance-metadata

Capture instance metadata.

lf-infra-create-netrc

Create a ~/.netrc file from a Maven settings.xml

Required parameters

server-id The id of a server as defined in settings.xml.

Optional parameters

ALT_NEXUS_SERVER URL of custom nexus server. If set this will take precedence. Use this to point at nexus3.$PROJECTDOMAIN for example.

lf-infra-deploy-maven-file

Deploy files to a repository.

Required parameters

global-settings-file Global settings file to use.

group-id Group ID of the repository.

maven-repo-url URL of a Maven repository to upload to.

mvn-version Version of Maven to use.

repo-id Repository ID

settings-file Maven settings file to use.

upload-files-dir Path to directory containing one or more files

lf-infra-docker-login

Login into a custom hosted docker registry and / or docker.io

The Jenkins system should have the following global variables defined

Environment variables

DOCKER_REGISTRY The DNS address of the registry (IP or FQDN) ex: nexus3.example.com (GLOBAL variable)

REGISTRY_PORTS Required when setting `DOCKER_REGISTRY`. Space-separated list of the registry ports to login to. ex: 10001 10002 10003 10004 (GLOBAL variable)

DOCKERHUB_EMAIL If set, then the job will attempt to login to DockerHub (docker.io). Set to the email address for the credentials that will get looked up. Returns the _first_ credential from the maven settings file for DockerHub. (GLOBAL variable)

lf-infra-gpg-verify-git-signature

Verify gpg signature of the latest commit message in $WORKSPACE. This command assumes that $WORKSPACE is a git repo.

lf-infra-pre-build

Macro that runs before all builders to prepare the system for job use.

lf-infra-package-listing

Lists distro level packages.

lf-infra-packer-build

Run *packer build* to build system images.

Required parameters

openstack Packer template uses an OpenStack builder (true|false).

openstack-cloud Sets OS_CLOUD variable to the value of this parameter.

packer-version Version of packer to use.

platform Build platform as found in the vars directory.

template Packer template to build as found in the templates directory.

Optional parameters

update-cloud-image Submit a change request to update new built cloud image to Jenkins.

lf-infra-packer-validate

Run `packer validate` to verify packer configuration.

Required parameters

openstack Packer template uses an OpenStack builder (true|false).

openstack-cloud Sets OS_CLOUD variable to the value of this parameter.

packer-cloud-settings Cloud configuration file. Loaded on the build server as CLOUDENV environment variable.

packer-version Version of packer to use.

lf-infra-push-gerrit-patch

Push a change through a Jenkins job to a Gerrit repository in an automated way using git-review.

Required parameters

gerrit-commit-message Commit message to assign.

gerrit-host Gerrit hostname.

gerrit-topic Gerrit topic.

gerrit-user Gerrit user-id used for submitting the change.

reviewers-email Reviewers email. Space-separated list of email addresses to CC on the patch.

project Gerrit project name.

lf-infra-ship-logs

Gather and deploy logs to a log server.

lf-infra-sysstat

Retrieves system stats.

lf-infra-update-packer-images

Find and update the new built cloud image{s} in the ci-management source repository.

lf-jacoco-nojava-workaround

Workaround for Jenkins not able to find Java in JaCoCo runs.

lf-maven-central

Publish artifacts to OSSRH (Maven Central) staging.

Requires that the project's settings.xml contains a ServerId 'ossrh' with the credentials for the project's OSSRH account.

This macro assumes the directory `$WORKSPACE/m2repo` contains a Maven 2 repository which is to upload to OSSRH.

Required parameters

mvn-central Set to `true` to upload to mvn-central. (true|false)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-settings The name of settings file containing credentials for the project.

ossrh-profile-id Nexus staging profile ID as provided by OSSRH.

```
---
- job-template:
    name: lf-maven-central-macro-test

    ######################
    # Default variables #
    ######################

    mvn-central: true
    mvn-global-settings: ""
    mvn-settings: ""
    ossrh-profile-id: ""

    ######################
    # Job configuration #
    ######################

    builders:
      - lf-maven-central:
          mvn-central: "{mvn-central}"
          mvn-global-settings: "{mvn-global-settings}"
          mvn-settings: "{mvn-settings}"
          ossrh-profile-id: "{ossrh-profile-id}"
```

lf-maven-install

Call maven-target builder with a goal of `--version` to force Jenkins to install the declared version of Maven. Use this as a preparation step for any shell scripts that want to use Maven.

Required parameters

mvn-version Version of Maven to install.

lf-packagecloud-file-provider

Provisions files required by the Ruby gem package_cloud, namely ".packagecloud" and "packagecloud_api" in the Jenkins home directory.

lf-packagecloud-push

Pushes DEB/RPM package files to PackageCloud using the Ruby gem package_cloud.

Required parameters

build-dir Directory with deb/rpm files to push

debian-distribution-versions list of DEB package distro/version strings separated by space; example: ubuntu/bionic debian/stretch

packagecloud-account PackageCloud account ID; example: oran

packagecloud-repo PackageCloud repository; example: master, staging

rpm-distribution-versions list of RPM package distro/version strings separated by space; example: el/4 el/5

lf-pip-install

Call pip install to install packages into a virtualenv located in /tmp/v/VENV

Note: Uses the first package listed in PIP_PACKAGES as the VENV name.

lf-provide-maven-settings

Push a global settings and user settings maven files to the build node.

lf-provide-maven-settings-cleanup

Cleanup maven `settings.xml` configuration. Set at the end of any macros that calls the *lf-provide-maven-settings* macro.

lf-rtd-trigger-build

Script to trigger a build on http://readthedocs.org

lf-rtd-verify

ReadTheDocs verify script. Installs and runs tox.

Required parameters

doc-dir Document directory.

python-version Python version.

lf-rtdv3-build

Read the docs scripts that leverage the new Read the Docs v3 api RTD v3 API Runs tox to verify that the docs are good and then runs the RTDv3 shell script. This script handles creating projects as needed, assiging subprojects to the main read the docs project and triggering builds to update the documentation. Jobs will run but skip verify bits until a `.readthedocs.yaml` exists in the root of their repository.

check-info-votes

Validates votes on a changes to `INFO.yaml`.

lf-release

releases lftools.ini (required) needed to push to nexus.

[nexus] username= password=

Then runs ../shell/release-job.sh

lf-sigul-sign-dir

Use Sigul to sign a directory via {sign-dir}.

Requires `SIGUL_BRIDGE_IP` configured as a global envvar.

Required Parameters

sign-artifacts Set `true` to sign artifacts with Sigul.

sign-dir Directory to sign.

sign-mode serial|parallel

lf-infra-provide-docker-cleanup

Forcefully removes all docker images.

lf-infra-sonar

Runs Jenkins SonarQube plug-in.

Requires `SonarQube Scanner for Jenkins`

Note: Optionally, set Sonar properties directly in the job definition by setting the sonar-project-file to `""` and adding all properties under `sonar-properties`.

Optional Parameters

sonar-task Sonar task to run. (default: "")

sonar-project-file The filename for the project's properties (default: "sonar-project.properties")

sonar-properties Sonar configuration properties. (default: "")

sonar-java-opts JVM options. (default: "")

sonar-additional-args Additional command line arguments. (default: "")

lf-infra-sonar-with-prescan

Runs Jenkins SonarQube plug-in after a pre-scan builder.

Requires `SonarQube Scanner for Jenkins`

Note: Optionally, set Sonar properties directly in the job definition by setting the sonar-project-file to `""` and adding all properties under `sonar-properties`.

Required Parameters

lf-sonar-prescan A builder that will run before the Sonar scan.

Optional Parameters

sonar-task Sonar task to run. (default: "")

sonar-project-file The filename for the project's properties (default: "sonar-project.properties")

sonar-properties Sonar configuration properties. (default: "")

sonar-java-opts JVM options. (default: "")

sonar-additional-args Additional command line arguments. (default: "")

lf-infra-pipeline-verify

Verify a Jenkins pipeline by linting it and ensuring that it cannot run on the Jenkins master.

3.7.2 Parameters

lf-autotools-parameters

Provides parameters needed by configure and make. Use in any jobs that need to call the `configure && make` pattern.

lf-clm-parameters

Provides the policy evaluation stage to run against Nexus IQ Server. Valid values include: 'build', 'stage-release', 'operate'.

lf-cmake-parameters

Provides parameters required by CMake. Use in any jobs that need to call the `cmake && make && make install` pattern.

lf-infra-maven-parameters

Provides parameters required by Maven. Use in any jobs that need to call the `mvn` CLI.

lf-infra-openstack-parameters

Provides parameters needed by OpenStack client CLI. Use in jobs that need to call the openstack cli.

Required Parameters

os-cloud Configures `OS_CLOUD` envvar as used by openstack cli.

lf-infra-parameters

Standard parameters used in the LF CI environments. GitHub projects will ignore the Gerrit variables and vice-versa, so defining them is not harmful. Use in every job template.

lf-infra-node-parameters

Provides parameters needed by NodeJS and NPM. Use in any jobs that need to run NodeJS or NPM.

lf-infra-sonar-cli-parameters

Provides parameters needed by Python jobs to run the SonarCloud CLI.

lf-infra-tox-parameters

Provides parameters required by python-tox. Use in any jobs that need to run *tox <https://tox.readthedocs.io>*.

lf-build-with-parameters-maven-release

Provides parameters needed for maven release jobs 'build with parameters'.

3.7.3 Properties

lf-infra-properties

Configures the build-discarder plugin for Jenkins with the recommended lf-infra settings. We recommend to include in all job-templates.

3.7.4 Publishers

lf-jacoco-report

Provides basic configuration for the JaCoCo plugin.

lf-infra-publish

Provides basic lf-infra recommended publisher configurations for use in all job templates. The purpose of this trigger is to gather package listing, instance metadata, sar reports, build logs and copy them to a log server.

lf-infra-publish-windows

Windows publisher for use at the end of Windows job templates. Takes care of cleaning out the workspace at the end of a job.

3.7.5 SCM

lf-infra-gerrit-scm

Basic SCM configuration for Gerrit based projects.

Required parameters

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

lf-infra-github-scm

Basic SCM configuration for GitHub based projects.

On the *branch* variable you can assign `$sha1` or `$ghprbActualCommit` as the value. This will enable the jobs to trigger via the GHPRB plugin.

Required parameters

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

3.7.6 Wrappers

lf-infra-wrappers-common

Provides lf-infra recommended wrappers for use in every job-template. Include this wrapper when creating more specific platform wrappers to ensure they pick up the common settings.

lf-infra-wrappers

Provides lf-infra recommended wrappers for use in every job-template targetting Linux systems.

This wrapper requires a managed file named `npmrc` to exist in Jenkins. The main use case here is to point to a npm proxy, on Nexus for example. Set the file type to "Custom file". You can set any npmrc settings in it, documentation on npm configuration is available at <https://docs.npmjs.com/files/npmrc>. If you are not using npm then create an empty file.

Example npmrc:

```
registry=https://nexus3.onap.org/repository/npm.public/
```

lf-infra-wrappers-windows

Provides lf-infra recommended wrappers for use in every job-template targetting Windows systems.

global-jjb-email-notification

Provides a publisher macro that ties into the lf-openstack-cron job to alert admins if this job fails.

3.8 Maven Jobs

3.8.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35

- project:
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of Maven job groups:

```
---
- job-group:
    name: "{project-name}-maven-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Gerrit that builds with maven.

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-stage
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies

- job-group:
    name: "{project-name}-github-maven-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Github that builds with maven.

    jobs:
      - github-maven-clm
      - github-maven-merge
      - github-maven-stage
      - github-maven-verify

- job-group:
    name: "{project-name}-gerrit-maven-docker-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Gerrit that builds with maven and docker.

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-docker-merge
      - gerrit-maven-docker-stage
      - gerrit-maven-docker-verify

- job-group:
    name: "{project-name}-github-maven-docker-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Github that builds with maven and docker.

    jobs:
      - github-maven-clm
      - github-maven-docker-merge
      - github-maven-docker-stage
      - github-maven-docker-verify
```

(continues on next page)

(continued from previous page)

```
- job-group:
    name: "{project-name}-maven-javadoc-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Gerrit that publishes javadoc with maven.

    jobs:
      - gerrit-maven-javadoc-publish
      - gerrit-maven-javadoc-verify

- job-group:
    name: "{project-name}-github-maven-javadoc-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Java project in Github that publishes javadoc with maven.

    jobs:
      - github-maven-javadoc-publish
      - github-maven-javadoc-verify
```

3.8.2 Macros

lf-infra-maven-sonar

Runs Sonar against a Maven project.

Required Parameters

java-version Version of Java to execute Sonar with. (default: openjdk11)

mvn-version Version of Maven to execute Sonar with.

mvn-settings Maven settings.xml file containing credentials to use.

lf-infra-maven-sonarcloud

Runs Sonar against a Maven project and pushes results to SonarCloud.

Required Parameters

java-version Version of Java to execute Maven build. (default: openjdk11)

mvn-version Version of Maven to execute Sonar with.

mvn-settings Maven settings.xml file containing credentials to use.

sonarcloud-project-key SonarCloud project key.

sonarcloud-project-organization SonarCloud project organization.

sonarcloud-java-version Version of Java to run the Sonar scan. (default: openjdk11)

sonarcloud-qualitygate-wait SonarCloud flag that forces the analysis step to wait for the quality gate result. (default: false)

lf-maven-build

Calls the maven build script to perform a maven build.

Required parameters

mvn-goals The maven goals to perform for the build. (default: clean deploy)

lf-maven-common

Common Jenkins configuration for Maven jobs.

lf-maven-deploy

Calls the maven deploy script to push artifacts to Nexus.

lf-maven-versions-plugin

Conditionally calls Maven versions plugin to set, update and commit the maven *versions:set*.

Required Parameters

maven-versions-plugin Whether to call Maven versions plugin or not. (default: false)

version-properties-file Name and path of the version properties file. (default: version.properties)

mvn-version Version of Maven to execute Sonar with.

mvn-pom Location of pom.xml.

mvn-settings Maven settings.xml file containing credentials to use.

lf-maven-stage

Calls the maven stage script to push artifacts to a Nexus staging repository.

Required Parameters

mvn-global-settings The name of the Maven global settings to use for Maven configuration.

mvn-settings The name of settings file containing credentials for the project.

lf-update-java-alternatives

Setup Java alternatives for the Distro.

Required Parameters

java-version Version of Java to set as the default Java. Eg. openjdk11

lf-infra-sonatype-clm

Runs a Sonatype CLM scan against a Maven project and pushes results to Nexus IQ server.

Optional parameters

mvn-goals The maven goals to perform for the build. (default: clean install)

lf-infra-snyk-cli-scanner

Downloads the latest Snyk CLI and triggers a code scan. It publishes a report into the Snyk dashboard.

Optional parameters

mvn-goals The maven goals to perform for the build. (default: clean install)

lf-infra-maven-sbom-generator

Runs a specific version of SPDX SBOM Generator tool to generate a report. The calling job template sets the version to run in the SBOM_GENERATOR_VERSION parameter.

Optional parameters

sbom-flags SBOM generator options. See https://github.com/opensbom-generator/spdx-sbom-generator

3.8.3 Job Templates

Maven CLM

Produces a CLM scan of the code into Nexus IQ Server.

Template Names

- {project-name}-maven-clm-{stream}
- gerrit-maven-clm
- github-maven-clm

Comment Trigger run-clm

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-goals The maven goals to perform for the build. (default: clean install)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

nexus-iq-namespace Insert a namespace to project AppID for projects that share a Nexus IQ system to avoid project name collision. We recommend inserting a trailing - dash if using this parameter. For example 'odl-'. (default: '')

nexus-iq-stage Sets the **stage** which the policy evaluation will run against on the Nexus IQ Server. (default: 'build')

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_merge_triggers Override Gerrit Triggers.

Maven SNYK CLI

Builds the code, downloads and runs a Snyk CLI scan of the code into the Snyk dashboard.

Template Names

- {project-name}-maven-snyk-cli-{stream}
- gerrit-maven-snyk-cli
- github-maven-snyk-cli

Comment Trigger run-snyk

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

snyk-token-credential-id Snyk API token to communicate with Jenkins.

snyk-org-credential-id Snyk organization ID.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-goals The maven goals to perform for the build. (default: clean install)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

snyk-cli-options Snyk CLI options. (default: '')

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_snyk_triggers Override Gerrit Triggers.

Maven JavaDoc Publish

Produces and publishes javadocs for a Maven project.

Expects javadocs to be available in `$WORKSPACE/target/site/apidocs`, but overrideable with the `mvn-dir` parameter. If set, will search for javadocs in `$WORKSPACE/{mvn-dir}/target/site/apidocs`.

Template Names

- {project-name}-maven-javadoc-publish-{stream}-{java-version}
- gerrit-maven-javadoc-publish
- github-maven-javadoc-publish

Comment Trigger remerge

Required parameters

build-node The node to run build on.

javadoc-path The path in Nexus to deploy javadoc to.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

mvn-site-id Maven Server ID from settings.xml to pull credentials from. (Note: This setting is generally configured in `defaults.yaml`.)

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-dir Directory supplied as argument to -f option (default: '.')

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '') Must not include a "-f" option; see parameter mvn-dir.

mvn-version Version of maven to use. (default: mvn35)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_merge_triggers Override Gerrit Triggers.

Maven JavaDoc Verify

Produces javadocs for a Maven project.

Expects javadocs to be available in `$WORKSPACE/target/site/apidocs`, but overrideable with the `mvn-dir` parameter. If set, will search for javadocs in `$WORKSPACE/{mvn-dir}/target/site/apidocs`.

Template Names

- {project-name}-maven-javadoc-verify-{stream}-{java-version}
- gerrit-maven-javadoc-verify
- github-maven-javadoc-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

deploy-path The path in Nexus to deploy javadoc to. (default: $PROJECT/$STREAM)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-dir Directory supplied as argument to -f option (default: '.')

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '') Must not include a "-f" option; see parameter mvn-dir.

mvn-version Version of maven to use. (default: mvn35)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

Maven Merge

Merge job which runs *mvn clean deploy* to build a project.

This job pushes files to Nexus using cURL instead of allowing the Maven deploy goal to push the upload. This is to get around the issue that Maven deploy does not properly support uploading files at the end of the build and instead pushes as it goes. There exists a `-Ddeploy-at-end` feature but it does not work with extensions.

This job uses the following strategy to deploy jobs to Nexus:

1. `wget -r` to fetch maven-metadata.xml from Nexus
2. `mvn deploy -DaltDeploymentRepository` to prepare files for upload
3. Removes untouched maven-metadata.xml files before upload
4. Use lftools (cURL) upload script to push artifacts to Nexus

Template Names

- {project-name}-maven-merge-{stream}
- gerrit-maven-merge
- github-maven-merge

Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

mvn-snapshot-id Maven Server ID from settings.xml to pull credentials from. (Note: This setting is generally configured in `defaults.yaml`.)

nexus-snapshot-repo The repository id of the Nexus snapshot repo to deploy to.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: 'H H * * 0' to run weekly)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

nexus-cut-dirs Number of directories to cut from file path for *wget -r*.

pre-build-script Shell script to run before maven build. (default: a string with a shell comment)

post-build-script Shell script to run after maven build. (default: a string with a shell comment)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

Maven Merge for Docker

Produces a snapshot docker image in a Nexus registry. Appropriate for Java projects that do not need to deploy any POM or JAR files.

Like the Maven Merge job as described above but logs in to Docker registries first and skips the lf-maven-deploy builder. The project POM file should invoke a plugin to build and push a Docker image. This pulls the base image from the registry in the environment variable `CONTAINER_PULL_REGISTRY` and pushes new image into the registry in the environment variable `CONTAINER_PUSH_REGISTRY`.

Template Names

- {project-name}-maven-docker-merge-{stream}
- gerrit-maven-docker-merge
- github-maven-docker-merge

Required parameters

container-public-registry Docker registry source with base images.

container-snapshot-registry Docker registry target for the deploy action.

All other required and optional parameters are identical to the Maven Merge job described above.

Maven Stage

Produces a release candidate by creating a staging repo in Nexus.

The staging repo name is in the format PROJECT-NUMBER for example "aaa-1234", "autorelease-2000", "odlparent-1201", etc…

This job runs a Maven build and deploys to $WORKSPACE/m2repo directory. This directory is then used later to deploy to Nexus.

Template Names

- {project-name}-maven-stage-{stream}
- gerrit-maven-stage
- github-maven-stage

Comment Trigger "stage-release" or "stage-maven-release"

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

mvn-staging-id Maven Server ID from settings.xml to pull credentials from. (Note: This setting is generally configured in `defaults.yaml`.)

staging-profile-id Profile ID of the project's Nexus staging profile.

Optional parameters

archive-artifacts Artifacts to archive to the logs server (default: '').

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: '')

deploy-path The path in Nexus to deploy javadoc to. (default: $PROJECT/$STREAM)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-central Set to `true` to also stage to **OSSRH**. This is for projects that want to release to Maven Central. If set, then also set the parameter `ossrh-profile-id`. (default: false)

maven-versions-plugin Whether to call Maven versions plugin or not. (default: false)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

ossrh-profile-id Profile ID for project as provided by OSSRH. (default: '')

sbom-flags SBOM generator options if using sbom-generator. See https://github.com/opensbom-generator/spdx-sbom-generator

sbom-generator Calls lf-infra-maven-sbom-generator to run the SPDX SBOM generator tool. (default: false)

sbom-generator-version SBOM generator version to download and run if using sbom-generator. (default: v0.0.10)

sbom-path SBOM execution path. (default: $WORKSPACE)

sign-artifacts Sign artifacts with Sigul. (default: false)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

version-properties-file Name and path of the version properties file. (default: version.properties)

gerrit_release_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

Maven Stage for Docker

Produces a release candidate docker image in a Nexus registry. Appropriate for Java projects that do not need to deploy any POM or JAR files.

Like the Maven Stage job as described above but logs in to Docker registries first and skips the lf-maven-deploy builder. The project POM file should invoke a plugin to build and push a Docker image. This pulls the base image from the registry in the environment variable `CONTAINER_PULL_REGISTRY` and pushes new image into the registry in the environment variable `CONTAINER_PUSH_REGISTRY`.

Template Names

- {project-name}-maven-docker-stage-{stream}
- gerrit-maven-docker-stage
- github-maven-docker-stage

Comment Trigger "stage-release" or "stage-docker-release"

Required parameters

container-public-registry Docker registry source with base images.

container-staging-registry Docker registry target for the deploy action.

Optional parameters

gerrit_release_docker_triggers Override Gerrit Triggers.

All other required and optional parameters are identical to the Maven Stage job described above.

Maven Sonar

Sonar job which runs mvn clean install then publishes to Sonar.

This job purposely runs on the `master` branch and does not support multi-branch configuration.

Template Names

- {project-name}-sonar
- gerrit-maven-sonar
- github-maven-sonar
- {project-name}-sonar-prescan-script
- gerrit-maven-sonar-prescan-script
- github-maven-sonar-prescan-script

Comment Trigger run-sonar

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

sonar-prescan-script (maven-sonar-prescan-script jobs) A shell script to run before sonar scans.

Optional parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: 'H H * * 6' to run weekly)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the Maven build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-goals The maven goals to perform for the build. (default: clean install)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

sonar-mvn-goals Maven goals to run for sonar analysis. (default: sonar:sonar)

sonarcloud Set to `true` to use SonarCloud `true|false`. (default: false)

sonarcloud-project-key SonarCloud project key. (default: '')

sonarcloud-project-organization SonarCloud project organization. (default: '')

sonarcloud-api-token-cred-id Jenkins credential ID which has the SonarCloud API Token. This one SHOULDN'T be overwritten as we are standarizing the credential ID for all projects (default: 'sonarcloud-api-token')

sonarcloud-java-version Version of Java to use for the Sonar scan. (default: openjdk11)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

scan-dev-branch Run the scan on a developer branch. (default: false)

gerrit_sonar_triggers Override Gerrit Triggers.

SonarCloud Example:

```
---
- project:
    name: example-sonarcloud
    jobs:
      - gerrit-maven-sonar

    project: "sonarcloud"
    project-name: "sonarcloud"
    branch: "master"
    mvn-settings: "sonarcloud-settings"
    mvn-opts: "-Xmx1024m"
    sonarcloud: true
    sonarcloud-project-key: KEY
    sonarcloud-project-organization: ORGANIZATION
    sonarcloud-api-token-cred-id: TOKEN
    scan-dev-branch: false
    sonarcloud-qualitygate-wait: false

- project:
    name: example-sonarcloud-with-prescan-script
    jobs:
      - gerrit-maven-sonar-prescan-script

    project: "sonarcloud"
    project-name: "sonarcloud"
    branch: "master"
    mvn-settings: "sonarcloud-settings"
    mvn-opts: "-Xmx1024m"
    sonarcloud: true
    sonarcloud-project-key: KEY
    sonarcloud-project-organization: ORGANIZATION
    sonarcloud-api-token-cred-id: TOKEN
    sonar-prescan-script: |
      echo "Run script at start of job."
    scan-dev-branch: false
    sonarcloud-qualitygate-wait: false

- project:
    name: example-sonarcloud-verify
    jobs:
```

(continues on next page)

(continued from previous page)

```
    - gerrit-maven-sonar-verify

  project: "sonarcloud"
  project-name: "sonarcloud"
  branch: "master"
  mvn-settings: "sonarcloud-settings"
  mvn-opts: "-Xmx1024m"
  sonarcloud: true
  sonarcloud-project-key: KEY
  sonarcloud-project-organization: ORGANIZATION
  sonarcloud-api-token-cred-id: TOKEN
  scan-dev-branch: true
  sonarcloud-qualitygate-wait: true
```

Maven Sonar Verify

Sonar job which runs mvn clean install then publishes to Sonar.

This job runs on dev branches and its triggered on new patchsets.

Template Names

- {project-name}-sonar-verify
- gerrit-maven-sonar-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the Maven build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-goals The maven goals to perform for the build. (default: clean install)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

pre-build-script Shell script to run before maven build. (default: a string with a shell comment)

post-build-script Shell script to run after maven build. (default: a string with a shell comment)

sonar-mvn-goal Maven goals to run for sonar analysis. (default: sonar:sonar)

sonarcloud Set to `true` to use SonarCloud `true|false`. (default: true)

sonarcloud-project-key SonarCloud project key. (default: '')

sonarcloud-project-organization SonarCloud project organization. (default: '')

sonarcloud-api-token-cred-id Jenkins credential ID which has the SonarCloud API Token. This one SHOULDN'T be overwritten as we are standarizing the credential ID for all projects (default: 'sonarcloud-api-token')

sonarcloud-java-version Version of Java to use for the Sonar scan. (default: openjdk11)

sonarcloud-qualitygate-wait SonarCloud flag that forces the analysis step to wait for the quality gate result. (default: false)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

scan-dev-branch Run the scan on a developer branch. (default: true)

gerrit_sonar_triggers Override Gerrit Triggers.

SonarCloud Example:

```
---
- project:
    name: example-sonarcloud
    jobs:
      - gerrit-maven-sonar

    project: "sonarcloud"
    project-name: "sonarcloud"
    branch: "master"
    mvn-settings: "sonarcloud-settings"
    mvn-opts: "-Xmx1024m"
    sonarcloud: true
    sonarcloud-project-key: KEY
    sonarcloud-project-organization: ORGANIZATION
    sonarcloud-api-token-cred-id: TOKEN
    scan-dev-branch: false
    sonarcloud-qualitygate-wait: false

- project:
    name: example-sonarcloud-with-prescan-script
    jobs:
      - gerrit-maven-sonar-prescan-script

    project: "sonarcloud"
    project-name: "sonarcloud"
    branch: "master"
    mvn-settings: "sonarcloud-settings"
    mvn-opts: "-Xmx1024m"
```

(continues on next page)

(continued from previous page)

```
    sonarcloud: true
    sonarcloud-project-key: KEY
    sonarcloud-project-organization: ORGANIZATION
    sonarcloud-api-token-cred-id: TOKEN
    sonar-prescan-script: |
      echo "Run script at start of job."
    scan-dev-branch: false
    sonarcloud-qualitygate-wait: false

- project:
    name: example-sonarcloud-verify
    jobs:
      - gerrit-maven-sonar-verify

    project: "sonarcloud"
    project-name: "sonarcloud"
    branch: "master"
    mvn-settings: "sonarcloud-settings"
    mvn-opts: "-Xmx1024m"
    sonarcloud: true
    sonarcloud-project-key: KEY
    sonarcloud-project-organization: ORGANIZATION
    sonarcloud-api-token-cred-id: TOKEN
    scan-dev-branch: true
    sonarcloud-qualitygate-wait: true
```

Maven Verify

Verify job which runs mvn clean install to test a project build..

Template Names

- {project-name}-maven-verify-{stream}-{mvn-version}-{java-version}
- gerrit-maven-verify
- github-maven-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

Maven Verify for Docker

Like the Maven Verify job as described above but logs in to Docker registries first. The project POM file should invoke a plugin to build a Docker image. This pulls the base image from the registry in the environment variable `CONTAINER_PULL_REGISTRY`.

Template Names

- {project-name}-maven-docker-verify-{stream}-{mvn-version}-{java-version}
- gerrit-maven-docker-verify
- github-maven-docker-verify

Required parameters

container-public-registry Docker registry source with base images.

All other required and optional parameters are identical to the Maven Verify job described above.

Maven Verify w/ Dependencies

Verify job which runs mvn clean install to test a project build /w deps

This job's purpose is to verify a patch in conjunction to a list of upstream patches it depends on. The user of this job can provide a list of patches via comment trigger.

Template Names

- {project-name}-maven-verify-deps-{stream}-{mvn-version}-{java-version}
- gerrit-maven-verify-dependencies

Comment Trigger recheck: SPACE_SEPARATED_LIST_OF_PATCHES

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

mvn-settings The name of settings file containing credentials for the project.

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use for Maven configuration. (default: global-settings)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths to filter which file modifications will trigger a build.

3.9 NodeJS Jobs

3.9.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35

- project:
```

(continues on next page)

(continued from previous page)

```
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of Node job groups:

```
---
- job-group:
    name: "{project-name}-github-node-jobs"

    # Job group containing recommended jobs to deploy for a Node Project.

    node-version: 6.11.4

    jobs:
      - github-node-verify

- job-group:
    name: "{project-name}-node-jobs"

    # Job group containing recommended jobs to deploy for a Node Project.

    node-version: 6.11.4

    jobs:
      - gerrit-node-verify
```

3.9.2 Job Templates

Node Verify

Verify job for NodeJS projects

Template Names

- {project-name}-node-verify-{stream}
- gerrit-node-verify
- github-node-verify

Comment Trigger recheck|reverify

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

node-version Version of NodeJS to install. Default defined in job-group.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

node-dir Path to a NodeJS project to run node test against (default: '')

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build.

3.10 OpenStack Heat

This section contains a series of macros for projects that need to spin up full test labs using HEAT scripts.

3.10.1 Job Setup

The 2 macros *lf-stack-create* & *lf-stack-delete* are companion macros and used together when constructing a job template that needs to spin up a full integration lab using Heat Orchestration Templates (HOT).

Example Usage:

```
- job-template:
    name: csit-test

    ######################
    # Default variables #
    ######################

    openstack-cloud: vex
    openstack-heat-template: csit-2-instance-type.yaml
    openstack-heat-template-dir: 'openstack-hot'

    odl_system_count: 1
    odl_system_flavor: odl-highcpu-4
    odl_system_image: ZZCI - CentOS 7 - builder - x86_64 - 20181010-215635.956
    tools_system_count: 1
    tools_system_flavor: odl-highcpu-2
    tools_system_image: ZZCI - Ubuntu 16.04 - mininet-ovs-25 - 20181029-223449.514

    ######################
    # Job configuration #
```

(continues on next page)

(continued from previous page)

```
    ####################

    builders:
      - lf-infra-pre-build
      - lf-stack-create:
          openstack-cloud: '{openstack-cloud}'
          openstack-heat-template: '{openstack-heat-template}'
          openstack-heat-template-dir: '{openstack-heat-template-dir}'
          openstack-heat-parameters: |
              vm_0_count: '{odl_system_count}'
              vm_0_flavor: '{odl_system_flavor}'
              vm_0_image: '{odl_system_image}'
              vm_1_count: '{tools_system_count}'
              vm_1_flavor: '{tools_system_flavor}'
              vm_1_image: '{tools_system_image}'

    publishers:
      - lf-stack-delete:
          openstack-cloud: '{openstack-cloud}'
```

3.10.2 Macros

lf-stack-create

Creates an OpenStack stack as configured by the job. Name pattern of stack is `$SILO-$JOB_NAME-$BUILD_NUMBER`.

Requires `lf-infra-pre-build` macro to run first to install the `openstack` and `lftools` packages.

Requires a Config File Provider configuration for clouds.yaml named `clouds-yaml`.

Required Parameters

openstack-cloud The `OS_CLOUD` variable to pass to OpenStack client. (Docs: https://docs.openstack.org/python-openstackclient)

openstack-heat-template Name of template file to use when running stack create.

openstack-heat-template-dir Directory in the ci-management repo containing the OpenStack heat templates.

Example:

```
---
- job-template:
    name: stack-create-test

    ####################
    # Default variables #
    ####################

    openstack-cloud: lf-cloud
    openstack-heat-template: csit-2-instance-type.yaml
    openstack-heat-template-dir: "openstack-hot"
```

(continues on next page)

(continued from previous page)

```
    odl_system_count: 1
    odl_system_flavor: odl-highcpu-4
    odl_system_image: ZZCI - CentOS 7 - builder - x86_64 - 20181010-215635.956
    tools_system_count: 2
    tools_system_flavor: odl-highcpu-2
    tools_system_image: ZZCI - Ubuntu 16.04 - mininet-ovs-25 - 20181029-223449.514

    ######################
    # Job configuration #
    ######################

    builders:
      - lf-infra-pre-build
      - lf-stack-create:
          openstack-cloud: "{openstack-cloud}"
          openstack-heat-template: "{openstack-heat-template}"
          openstack-heat-template-dir: "{openstack-heat-template-dir}"
          openstack-heat-parameters: |
            vm_0_count: '{odl_system_count}'
            vm_0_flavor: '{odl_system_flavor}'
            vm_0_image: '{odl_system_image}'
            vm_1_count: '{tools_system_count}'
            vm_1_flavor: '{tools_system_flavor}'
            vm_1_image: '{tools_system_image}'
```

lf-stack-delete

Deletes the stack associated with this job. Name pattern of stack is `$SILO-$JOB_NAME-$BUILD_NUMBER`.

Requires `lf-infra-pre-build` macro to run first to install the `openstack` and `lftools` packages.

Requires a Config File Provider configuration for clouds.yaml named `clouds-yaml`.

Required Parameters

openstack-cloud The `OS_CLOUD` variable to pass to OpenStack client. (Docs: https://docs.openstack.org/python-openstackclient)

Example:

```
---
- job-template:
    name: stack-delete-test

    ######################
    # Default variables #
    ######################

    openstack-cloud: lf-cloud

    ######################
    # Job configuration #
    ######################
```

(continues on next page)

(continued from previous page)

```
    publishers:
      - lf-stack-delete:
          openstack-cloud: "{openstack-cloud}"
```

3.11 OpenStack Magnum (Kubernetes)

This section contains a series of macros for projects that need to spin up kubernetes clusters using JJB.

3.11.1 Job Setup

The two macros *lf-kubernetes-create* & *lf-kubernetes-delete* are companion macros and used together when constructing a job template that needs a kubernetes cluster.

Example Usage:

```
- job-template:
    name: k8s-test

    ######################
    # Default variables #
    ######################

    base-image: Fedora Atomic 29 [2019-08-20]
    boot-volume-size: 10
    cluster-settle-time: 1m
    docker-volume-size: 10
    fixed-network: ecompci
    fixed-subnet: ecompci-subnet1
    keypair: jenkins
    kubernetes-version: v1.16.0
    master-count: 1
    master-flavor: v2-standard-1
    node-count: 2
    node-flavor: v2-highcpu-8
    openstack-cloud: vex

    ######################
    # Job configuration #
    ######################

    builders:
      - lf-infra-pre-build
      - lf-kubernetes-create:
          openstack-cloud: "{openstack-cloud}"
          base-image: "{base-image}"
          boot-volume-size: "{boot-volume-size}"
          cluster-settle-time: "{cluster-settle-time}"
          docker-volume-size: "{docker-volume-size}"
          fixed-network: "{fixed-network}"
```

(continues on next page)

(continued from previous page)

```
        fixed-subnet: "{fixed-subnet}"
        keypair: "{keypair}"
        kubernetes-version: "{kubernetes-version}"
        master-count: "{master-count}"
        master-flavor: "{master-flavor}"
        node-count: "{node-count}"
        node-flavor: "{node-flavor}"
    publishers:
      - lf-kubernetes-delete
```

3.11.2 Macros

lf-kubernetes-create

Creates an OpenStack Kubernetes cluster as configured by the job. Name pattern of stack is `$SILO-$JOB_NAME-$BUILD_NUMBER`.

Requires `lf-infra-pre-build` macro to run first to install the `openstack` and `lftools` packages.

Requires a Config File Provider configuration for clouds.yaml named `clouds-yaml`.

Required Parameters

openstack-cloud The `OS_CLOUD` variable to pass to OpenStack client. (Docs: https://docs.openstack.org/python-openstackclient)

base-image The base image to use for building the cluster. LF is using the Fedora Atomic images.

boot-volume-size The size of the operating system disk for each node.

cluster-settle-time A parameter that controls the buffer time after cluster creation before we start querying the API for status.

docker-volume-size The size of the Docker volume.

fixed-network The private network to build the cluster on.

fixed-subnet The subnet to use from the above private network

keypair The ssh keypair to inject into the nodes for access.

kubernetes-version The version of kubernetes to use for the cluster. Available versions are v1.14, v1.15, and v1.16

master-count The number of masters for the cluster (configuring more than one master automatically triggers the creation of a load-balancer).

master-flavor The flavor (size) of the master node.

node-count The number of kubernetes nodes for the cluster.

node-flavor The flavor (size) of the worker nodes.

lf-kubernetes-delete

Deletes the stack associated with this job. Name pattern of stack is `$SILO-$JOB_NAME-$BUILD_NUMBER`.

Requires `lf-infra-pre-build` macro to run first to install the `openstack` and `lftools` packages.

Requires a Config File Provider configuration for clouds.yaml named `clouds-yaml`.

3.12 Pipeline Jobs

3.12.1 Macros

lf-pipeline-common

Common definitions for use within all pipeline jobs.

3.12.2 Job Templates

Pipeline Verify

Verify job that checks a Jenkins pipeline by linting it and ensuring that it cannot run on the master.

Template Names

- {project-name}-pipeline-verify-{stream}
- gerrit-pipeline-verify
- github-pipeline-verify

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-node The node to run build on.

build-timeout Timeout in minutes before aborting build. (default: 15)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

project-pattern Project to trigger build against. (default: **)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. **default**:

```
- compare-type: REG_EXP
  pattern: "Jenkinsfile.*"
```

3.13 Python Jobs

3.13.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35

- project:
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of Python job groups:

```
---
- job-group:
    name: "{project-name}-python-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Gerrit-based Python project to verify commits using tox.

    jobs:
      - gerrit-tox-verify
      - gerrit-tox-merge
```

(continues on next page)

(continued from previous page)

```
- job-group:
    name: "{project-name}-github-python-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for a Github-based Python project to verify commits using tox.

    jobs:
      - github-tox-verify
      - github-tox-merge

- job-group:
    name: "{project-name}-gerrit-pypi-jobs"

    # This job group contains the recommended jobs that should be deployed for
    # a Gerrit-based Python project to test, build and deploy a package.

    jobs:
      - gerrit-pypi-merge
      - gerrit-pypi-verify

- job-group:
    name: "{project-name}-github-pypi-jobs"

    # This job group contains the recommended jobs that should be deployed for
    # a Github-based Python project to test, build and deploy a package.

    jobs:
      - github-pypi-merge
      - github-pypi-verify

- job-group:
    name: "{project-name}-gerrit-pypi-release-jobs"

    # This job group contains the recommended jobs that should be deployed for
    # a Gerrit-based Python project to promote a package from staging to pypi.

    jobs:
      - gerrit-pypi-release-merge
      - gerrit-pypi-release-verify

- job-group:
    name: "{project-name}-github-pypi-release-jobs"

    # This job group contains the recommended jobs that should be deployed for
    # a Github-based Python project to promote a package from staging to pypi.

    jobs:
      - github-pypi-release-merge
      - github-pypi-release-verify
```

3.13.2 Macros

lf-infra-nexus-iq-python-cli

Runs Nexus IQ command-line interface CLM scan on Python package requirements.

Required Parameters

nexus-iq-project-name Project name in Nexus IQ to send results to.

requirements-file File name with output of pip freeze.

lf-infra-tox-install

Installs Tox into a virtualenv.

Required Parameters

python-version Version of Python to invoke the pip install of the tox-pyenv package that creates a virtual environment, either "python2" or "python3".

lf-infra-tox-run

Creates a Tox virtual environment and invokes tox.

Required Parameters

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false".

3.13.3 Job Templates

Tox Nexus IQ CLM

The Nexus IQ job invokes tox and the Nexus IQ scanner to analyze packages for component lifecycle management (CLM). Runs tox to discover the required packages, downloads the command-line interface (CLI) scanner, runs the scanner on the package list, then uploads the results to a Nexus IQ server. The project's tox.ini file must define a test environment that runs 'pip freeze' and captures the output; that environment does not need to execute any tests. For example:

```
[testenv:clm]
# use pip to report dependencies with versions
whitelist_externals = sh
commands = sh -c 'pip freeze > requirements.txt'
```

This job runs on the master branch because the basic Nexus IQ configuration does not support multi-branch.

Template Names

- {project-name}-tox-nexus-iq-clm
- gerrit-tox-nexus-iq-clm
- github-tox-nexus-iq-clm

Comment Trigger `run-clm`

Required parameters

build-node The node to run the build on. (Commonly in defaults.yaml)

jenkins-ssh-credential Credential to use for SSH. (Commonly in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional Parameters

archive-artifacts Pattern for files to archive to the logs server (default: '**/*.log')

branch Git branch, should be master (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via the YAML pipe | character to allow more than 1 cron timer. (default: @weekly)

disable-job Whether to disable the job (default: false)

gerrit_nexusiq_triggers Override Gerrit Triggers.

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

github-url URL for Github. (default: https://github.com)

java-version Version of Java to use for the scan. (default: openjdk11)

nexus-iq-cli-version Nexus IQ CLI package version to download and use. (default is a string like 1.89.0-02, see file lf-python-jobs.yaml)

nexus-iq-namespace Insert a namespace to project AppID for projects that share a Nexus IQ system to avoid project name collision. We recommend inserting a trailing - dash if using this parameter. For example 'odl-'. (default: '')

nexus-target-build Target directory or file for scanning by Nexus IQ CLI (default: "**/*")

pre-build-script Shell script to run before tox. Useful for setting up dependencies. (default: a string with a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

requirements-file Name of file with output of pip freeze. (default: requirements.txt)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environment with the appropriate pip freeze invocation. (default: 'clm')

Python Snyk CLI

Builds the code, downloads and runs a Snyk CLI scan of the code into the Snyk dashboard.

Template Names

- {project-name}-python-snyk-cli-{stream}
- gerrit-python-snyk-cli
- github-python-snyk-cli

Comment Trigger run-snyk

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally configured in defaults.yaml)

snyk-token-credential-id Snyk API token to communicate with Jenkins.

snyk-org-credential-id Snyk organization ID.

Optional parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre-build-script Shell script to execute before the Tox builder. For example, install prerequisites or move files to the repo root. (default: a string with a shell comment)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

snyk-cli-options Additional Snyk CLI options. (default: '')

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

Python Sonar with CLI

Sonar scans for non Maven based repos. This job downloads the CLI and runs a scan to publish the report to SonarCloud.

As suggested in SonarCoud's UI instructions, the job downloads and unzips the Sonar SLI and executes a sonar-scanner command to process the report.

For more details refer to sonar documentation:

https://docs.sonarcloud.io/advanced-setup/ci-based-analysis/sonarscanner-cli/

Template Names

- {project-name}-cli-sonar
- gerrit-cli-sonar
- github-cli-sonar

Comment Trigger **run-sonar** post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally should get configured in defaults.yaml)

mvn-settings The name of the settings file with credentials for the project.

Optional parameters

branch Git branch, should be master (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: H 11 * * * to run once a day)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

github-url URL for Github. (default: https://github.com)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use

mvn-goals The Maven goal to run first. (default: validate)

mvn-version Version of maven to use. (default: mvn35)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to execute before the Sonar builder. For example, install prerequisites or move files to the repo root. (default: a string with a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python2)

sonarcloud-project-key SonarCloud project key. (default: '')

sonarcloud-project-organization SonarCloud project organization. (default: '')

sonarcloud-api-token-cred-id Jenkins credential ID which has the SonarCloud API Token. This one SHOULDN'T be overwritten as per we are standarizing the credential ID for all projects (default: 'sonarcloud-api-token')

sonar-scanner-home Sonar scanner home directory. (default: $WORKSPACE/.sonar/sonar-scanner-$SONAR_SCANNER_VERSION-linux)

sonar-scanner-opts Sonar scanner Java options. (default: '-server')

sonar-scanner-version Version of sonar scanner to use. (default: 4.7.0.2747)

stream Keyword used to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_sonar_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

Python Sonar with Tox

Sonar scans for Python based repos. This job invokes tox to run tests and gather coverage statistics from the test results, then invokes Maven to publish the results to either a Sonar server or SonarCloud.

Deprecated, new projects should use Tox Sonarqube.

To get the Sonar coverage results, file tox.ini must exist and contain coverage commands to run.

The coverage commands define the code that gets executed by the test suites. Checking coverage does not guarantee that the tests execute properly, but it identifies code that is not executed by any test.

This job reuses the Sonar builders used for Java/Maven projects which run maven twice. The first invocation does nothing for Python projects, so the job uses the goal `validate` by default. The second invocation publishes results using the goal `sonar:sonar` by default.

For example:

```
[testenv:py3]
commands =
        coverage run --module pytest --junitxml xunit-results.xml
        coverage xml --omit=".tox/py3/*","tests/*"
        coverage report --omit=".tox/py3/*","tests/*"
```

For more details refer to coverage and sonar documentation:

https://coverage.readthedocs.io/

https://docs.sonarqube.org/display/PLUG/Python+Coverage+Results+Import

Template Names

- {project-name}-tox-sonar
- gerrit-tox-sonar
- github-tox-sonar

Comment Trigger **run-sonar** post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally should get configured in defaults.yaml)

mvn-settings The name of the settings file with credentials for the project.

Optional parameters

branch Git branch, should be master (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 60)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: H 11 * * * to run once a day)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

github-url URL for Github. (default: https://github.com)

java-version Version of Java to use for the build. (default: openjdk11)

mvn-global-settings The name of the Maven global settings to use

mvn-goals The Maven goal to run first. (default: validate)

mvn-version Version of maven to use. (default: mvn35)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to execute before the Sonar builder. For example, install prerequisites or move files to the repo root. (default: a string with a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python2)

sonarcloud Boolean indicator to use SonarCloud `true|false`. (default: false)

sonarcloud-project-key SonarCloud project key. (default: '')

sonarcloud-project-organization SonarCloud project organization. (default: '')

sonarcloud-api-token-cred-id Jenkins credential ID which has the SonarCloud API Token. This one SHOULDN'T be overwritten as per we are standarizing the credential ID for all projects (default: 'sonarcloud-api-token')

sonar-mvn-goal The Maven goal to run the Sonar plugin. (default: sonar:sonar)

stream Keyword used to represent a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_sonar_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

Tox SonarQube

The SonarQube job invokes tox to run tests and generate code-coverage statistics, then runs the SonarQube Scanner Jenkins plug-in to analyze code, gather coverage data, and upload the results to a SonarQube server such as SonarCloud.io. Optionally runs a shell script before tox.

Requires `SonarQube Scanner for Jenkins`

This job runs on the master branch because the basic Sonar configuration does not support multi-branch.

Plug-in configurations

Manage Jenkins –> Configure System –> SonarQube servers

- Name: Sonar (fixed)
- Server URL: https://sonar.project.org/ or https://sonarcloud.io
- Server authentication token: none for local, API token (saved as a "secret text" credential) for Sonarcloud

Manage Jenkins –> Global Tool Configuration –> SonarQube Scanner

- Name: SonarQube Scanner (fixed)
- Install automatically
- Select latest version

Template Names

- {project-name}-tox-sonarqube
- gerrit-tox-sonarqube
- github-tox-sonarqube

Comment Trigger `run-sonar`

Required parameters

build-node The node to run the build on. (Commonly in defaults.yaml)

jenkins-ssh-credential Credential to use for SSH. (Commonly in defaults.yaml)

project The git repository name.

project-name Prefix used to name jobs.

Optional Parameters

archive-artifacts Pattern for files to archive to the logs server (default: '**/*.log')

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

cron Cron schedule when to trigger the job. This parameter also supports multiline input via YAML pipe | character in cases where one may want to provide more than 1 cron timer. (default: @weekly)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

github-url URL for Github. (default: https://github.com)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to run before tox. Useful for setting up dependencies. (default: a string with a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

sonar-additional-args Command line arguments. (default: '')

sonar-java-opts JVM options. For example, use option -Xmx to increase the memory size limit. (default: '')

sonar-project-file The file name with Sonar configuration properties (default: sonar-project.properties)

sonar-properties Sonar configuration properties. (default: '')

sonar-task Sonar task to run. (default: '')

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

Note: A job definition must provide one of the optional parameters `sonar-project-file` and `sonar-properties`; they cannot both be empty. Set Sonar properties directly in the job definition by setting the `sonar-project-file` property to "" and adding all properties under `sonar-properties`.

Required Sonar Properties

- sonar.login: The API token for authentication at SonarCloud. Commonly defined as key "sonarcloud_api_token" in defaults.yaml.
- sonar.organization: The umbrella project name; e.g., "opendaylight". Commonly defined as key "sonarcloud_project_organization" in defaults.yaml.
- sonar.projectName: The git repository name without slashes; e.g., "infrautils".

- sonar.projectKey: The globally unique key for the report in SonarCloud. Most teams use the catenation of sonar.organization, an underscore, and sonar.projectName; e.g., "openday-light_infrautils".

Optional Sonar Properties

- sonar.cfamily.gcov.reportsPath: directory with GCOV output files
- Documentation of SonarQube properties is here: https://docs.sonarqube.org/latest/analysis/overview/

Example job definition

The following example defines a job for a basic Python project. This definition uses configuration parameters in the umbrella project's defaults.yaml file.

```
- project:
    name: my-package-sonar
    project: my/package
    project-name: my-package
    sonar-project-file: ""
    sonar-properties: |
        sonar.login={sonarcloud_api_token}
        sonar.projectKey={sonarcloud_project_organization}_{project-name}
        sonar.projectName={project-name}
        sonar.organization={sonarcloud_project_organization}
        sonar.sourceEncoding=UTF-8
        sonar.sources=mypackage
        sonar.exclusions=tests/*,setup.py
        sonar.python.coverage.reportPaths=coverage.xml
    jobs:
      - gerrit-tox-sonarqube
```

Tox Verify

Tox runner to verify a project on creation of a patch set. This job is pyenv aware so if the image contains an installation of pyenv at /opt/pyenv it will pick it up and run Python tests with the appropriate Python versions. This job will set the following pyenv variables before running.

```
export PYENV_ROOT="/opt/pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

Template Names

- {project-name}-tox-verify-{stream}
- gerrit-tox-verify
- github-tox-verify

Comment Trigger **recheck|reverify** post a comment with one of the triggers to launch this job manually. Do not include any other text or vote in the same comment.

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre-build-script Shell script to execute before the Tox builder. For example, install pre-requisites or move files to the repo root. (default: a string with a shell comment)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

python-version Python version to invoke pip install of tox-pyenv (default: python2)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

Tox Merge

Tox runner to verify a project after merge of a patch set. This job is pyenv aware so if the image contains an installation of pyenv at /opt/pyenv it will pick it up and run Python tests with the appropriate Python versions. This job will set the following pyenv variables before running.

```
export PYENV_ROOT="/opt/pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

Template Names

- {project-name}-tox-merge-{stream}
- gerrit-tox-merge
- github-tox-merge

Comment Trigger **remerge** post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 10)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pre-build-script Shell script to execute before the CLM builder. For example, install pre-requisites or move files to the repo root. (default: a string with a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python2)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

PyPI Merge

Creates and uploads package distribution files on merge of a patch set. Runs tox, builds a source distribution and (optionally) a binary distribution, and uploads the distribution(s) to a PyPI repository. The project git repository must have a setup.py file with configuration for packaging the component.

Projects can choose **either** this template to publish on merge, **or** the Stage template to publish on command.

This job should use a staging repository like testpypi.python.org, which sets up use of release jobs to promote the distributions later. This job can also use a public release area like the global PyPI repository if the release process is not needed. These PyPI repositories allow upload of a package at a specific version once, they do not allow overwrite of a package. This means that a merge job will fail in the upload step if the package version already exists in the target repository.

The tox runner is pyenv aware so if the image contains an installation of pyenv at /opt/pyenv it will pick it up and run Python tests with the appropriate Python versions. The tox runner sets the following pyenv variables before running.

```
export PYENV_ROOT="/opt/pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

See the recommended directory layout documented in the PyPI Verify job.

Jobs using this PyPI template depend on a .pypirc configuration file in the Jenkins builder home directory. An example appears next that uses API tokens. Note that in the [pypi] entry the repository key-value pair is optional, it defaults to pypi.org.

```
[distutils] # this tells distutils what package indexes you can push to
index-servers = pypi-test pypi

[pypi-test]
repository: https://test.pypi.org/legacy/
username: __token__
password: pypi-test-api-token-goes-here

[pypi]
username: __token__
password: pypi-api-token-goes-here
```

Template Names

- {project-name}-pypi-merge-{stream}
- gerrit-pypi-merge
- github-pypi-merge

Comment Trigger **remerge** post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required Parameters

build-node The node to run the build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

mvn-settings The settings file with credentials for the project

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

cron Cron schedule when to trigger the job. Supports regular builds. Not useful when publishing to pypi.org because that rejects a package if the version exists. (default: empty)

disable-job Whether to disable the job (default: false)

dist-binary Whether to build a binary wheel distribution. (default: true)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to execute before the tox builder. For example, install system prerequisites. (default: a shell comment)

pypi-repo Key for the PyPI target repository in the .pypirc file, ideally a server like test.pypi.org. (default: pypi-test)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

PyPI Stage

Creates and uploads package distribution files on receipt of a comment. Runs tox, builds a source distribution and (optionally) a binary distribution, and uploads the distribution(s) to a PyPI repository. The project git repository must have a setup.py file with configuration for packaging the component.

Projects can choose **either** this template to publish on command, **or** the Merge template to publish on merge.

This job should use a staging repository like testpypi.python.org, which sets up use of release jobs to promote the distributions later. This job can also use a public release area like the global PyPI repository if the release process is not needed. These PyPI repositories allow upload of a package at a specific version once, they do not allow overwrite of a package. This means that a job will fail in the upload step if the package version already exists in the target repository.

The tox runner is pyenv aware so if the image contains an installation of pyenv at /opt/pyenv it will pick it up and run Python tests with the appropriate Python versions. The tox runner sets the following pyenv variables before running.

```
export PYENV_ROOT="/opt/pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

See the recommended directory layout documented in the PyPI Verify job.

Jobs using this PyPI template depend on a .pypirc configuration file in the Jenkins builder home directory. An example appears next that uses API tokens. Note that in the [pypi] entry the repository key-value pair is optional, it defaults to pypi.org.

```
[distutils] # this tells distutils what package indexes you can push to
index-servers = pypi-test pypi

[pypi-test]
repository: https://test.pypi.org/legacy/
username: __token__
password: pypi-test-api-token-goes-here
```

(continues on next page)

(continued from previous page)

```
[pypi]
username: __token__
password: pypi-api-token-goes-here
```

Template Names

- {project-name}-pypi-stage-{stream}
- gerrit-pypi-stage
- github-pypi-stage

Comment Trigger **stage-release** post a comment with the trigger to launch this job manually. Do not include any other text or vote in the same comment.

Required Parameters

build-node The node to run the build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

mvn-settings The settings file with credentials for the project

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

cron Cron schedule when to trigger the job. Supports regular builds. Not useful when publishing to pypi.org because that rejects a package if the version exists. (default: empty)

disable-job Whether to disable the job (default: false)

dist-binary Whether to build a binary wheel distribution. (default: true)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to execute before the tox builder. For example, install system prerequisites. (default: a shell comment)

pypi-repo Key for the PyPI target repository in the .pypirc file, ideally a server like test.pypi.org. (default: pypi-test)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

PyPI Verify

Verifies a Python library project on creation of a patch set. Runs tox then builds a source distribution and (optionally) a binary distribution. The project repository must have a setup.py file with configuration for packaging the component.

Installable package projects should use the directory layout shown below. All Python files are in a repo subdirectory separate from non-Python files like documentation. This layout allows highly specific build-job triggers in Jenkins using the subdirectory paths. For example, a PyPI publisher job should not run on a non-Python file change such as documentation, because the job cannot upload the same package twice.

To make the document files available for building a Python package long description in setup.py, add a symbolic link "docs" in the package subdirectory pointing to the top-level docs directory.

```
git-repo-name/
|
├── docs/
|   ├── index.rst
|   └── release-notes.rst
|
├── helloworld-package/
|   |
|   └── helloworld/
|   |   ├── __init__.py
|   |   ├── helloworld.py
|   |   └── helpers.py
|   |
|   ├── tests/
|   |   ├── helloworld_tests.py
|   |   └── helloworld_mocks.py
|   |
|   ├── requirements.txt
|   └── setup.py
|   └── tox.ini
|
├── releases/
|   └── pypi-helloworld.yaml
|
├── .gitignore
```

(continues on next page)

(continued from previous page)

```
├── LICENSE
└── README.md
```

The tox runner is pyenv aware so if the image contains an installation of pyenv at /opt/pyenv it will pick it up and run Python tests with the appropriate Python versions. The tox runner sets the following pyenv variables before running.

```
export PYENV_ROOT="/opt/pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

Template Names

- {project-name}-pypi-verify-{stream}
- gerrit-pypi-verify
- github-pypi-verify

Comment Trigger **recheck|reverify** post a comment with one of the triggers to launch this job manually. Do not include any other text or vote in the same comment.

Required Parameters

build-node The node to run the build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

mvn-settings The settings file with credentials for the project

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

branch The branch to build against. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

disable-job Whether to disable the job (default: false)

dist-binary Whether to build a binary wheel distribution. (default: true)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

mvn-opts Sets MAVEN_OPTS to start up the JVM running Maven. (default: '')

mvn-params Parameters to pass to the mvn CLI. (default: '')

mvn-version Version of maven to use. (default: mvn35)

parallel If different from false, try pass this parameter to tox option "–parallel" to parallelize jobs in the envlist (and then activate the option "–parallel-live" to display output in logs). Possible values are "auto" (equivalent to "true" for legacy), "all" or any integer. Any other value is equivalent to "false". (default: false, in series)

pre-build-script Shell script to execute before the tox builder. For example, install system prerequisites. (default: a shell comment)

python-version Python version to invoke pip install of tox-pyenv (default: python3)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's tox.ini relative to the workspace. The default uses tox.ini at the project root. (default: '.')

tox-envs Tox environments to run. If blank run everything described in tox.ini. (default: '')

gerrit_trigger_file_paths Override file paths used to filter which file modifications trigger a build. Refer to JJB documentation for "file-path" details. https://jenkins-job-builder.readthedocs.io/en/latest/triggers.html#triggers.gerrit

3.14 Self-Serve Release Jobs

Self-serve release jobs allow project committers to promote a jar file, container image, Python package or PackageCloud artifact from a staging area to a release area. A release yaml file controls the process, and Jenkins promotes the artifact when a project committer merges the release yaml file in Gerrit.

To use the self-release process, create a releases/ or .releases/ directory at the root of the project repository, add one release yaml file to it, and submit a change set with that release yaml file. The required contents of the release yaml file are different for each release, see the schemas and examples shown below. The version string in the release yaml file should be a valid Semantic Versioning (SemVer) string, matching the pattern "#.#.#" where "#" is one or more digits. A version string matching the pattern "v#.#.#" is also accepted. Upon merge of the change, a Jenkins job promotes the artifact and pushes a gpg-signed tag to the repository.

Note: The release file regex is: (releases/.*.yaml|.releases/.*.yaml). In words, the directory name can be ".releases" or "releases"; the file name can be anything with suffix ".yaml". Some release jobs require a specific prefix on the file, as described below.

The build node for all release jobs must be CentOS, which supports the sigul client for accessing a signing server to sign a tag. The build node for container release jobs must have Docker installed.

A Jenkins admin user can also trigger a release job via the "Build with parameters" action, removing the need to create and merge a release yaml file. The user must enter parameters in the same way as a release yaml file, except for the special USE_RELEASE_FILE and DRY_RUN check boxes. The user must uncheck the USE_RELEASE_FILE check box if the job should run without a release file, instead passing the required information as build parameters. The user can check the DRY_RUN check box to test the job while skipping upload of files to the release repository.

For example, the parameters for a Maven release are as follows:

```
GERRIT_BRANCH = master
VERSION = 1.0.0
LOG_DIR = example-project-maven-stage-master/17/
DISTRIBUTION_TYPE = maven
USE_RELEASE_FILE = false
DRY_RUN = false
```

It's recommended to use Semantic Versions (SemVer) for releases. Refer to https://semver.org for more details on SemVer. For projects that do not follow SemVer can use a build parameter (OVERRIDE_SEMVER_REGEX) with the release job. This build param overrides the default SemVer regex.

3.14.1 Maven Release Files

An example of a maven release file appears below.

```
$ cat releases/maven-release.yaml
---
distribution_type: maven
log_dir: example-project-maven-stage-master/17/
project: example-project
version: 1.0.0
```

The following parameters must appear in a maven release yaml file.

Required Parameters

distribution_type Must be "maven".

log_dir The suffix of the logs URL reported on successful completion by the Jenkins stage job that created and pushed the artifact to the staging repository. For example, use value "example-project-maven-stage-master/17" for the logs URL https://logs.lf-project.org/production/vex-sjc-lfp-jenkins-prod-1/example-project-maven-stage-master/17

project The name of the project.

version The semantic version string used for the artifact.

Optional Parameters

git_tag The tag string to sign and push to the Git repository. (default: the semantic version string)

tag_release Tag Gerrit Repo during the release process. (default: true)

The JSON schema for a maven release file appears below.

```
# SPDX-License-Identifier: EPL-1.0
##############################################################################
# Copyright (c) 2019 The Linux Foundation and others.
#
# All rights reserved. This program and the accompanying materials
# are made available under the terms of the Eclipse Public License v1.0
# which accompanies this distribution, and is available at
# http://www.eclipse.org/legal/epl-v10.html
##############################################################################
---
$schema: "http://json-schema.org/schema#"
$id: "https://github.com/lfit/releng-global-jjb/blob/master/release-schema.yaml"

required:
  - "distribution_type"
  - "log_dir"
  - "project"
  - "version"

properties:
  distribution_type:
    type: "string"
  git_tag:
```

(continues on next page)

(continued from previous page)

```
    type: "string"
  log_dir:
    type: "string"
  project:
    type: "string"
  tag_release:
    type: "boolean"
  version:
    type: "string"
```

3.14.2 Container Release Files

An example of a container release file appears below.

```
$ cat releases/container-release.yaml
---
distribution_type: container
container_release_tag: 1.0.0
container_pull_registry: nexus.onap.org:10003
container_push_registry: nexus.onap.org:10002
project: test
ref: d1b9cd2dd345fbeec0d3e2162e008358b8b663b2
containers:
    - name: test-backend
      version: 1.0.0-20190806T184921Z
    - name: test-frontend
      version: 1.0.0-20190806T184921Z
```

The following parameters must appear in a container release yaml file.

Required Parameters

distribution_type Must be "container".

container_release_tag The string to use as a Docker tag on all released containers.

container_pull_registry The Nexus registry that supplies the staged image(s).

container_push_registry The Nexus registry that receives the released image(s).

project The name of the project.

ref The git commit reference (SHA-1 code) to tag with the version string.

containers A list of name and version (tag) pairs that specify the Docker images in the container-pull registry to promote to the container-push registry.

Optional Parameters

git_tag The tag string to sign and push to the Git repository. (default: the semantic version string)

tag_release Tag Gerrit Repo during the release process. (default: true)

The JSON schema for a container release file appears below.

```
# SPDX-License-Identifier: EPL-1.0
##############################################################################
# Copyright (c) 2019 The Linux Foundation and others.
#
# All rights reserved. This program and the accompanying materials
# are made available under the terms of the Eclipse Public License v1.0
# which accompanies this distribution, and is available at
# http://www.eclipse.org/legal/epl-v10.html
##############################################################################
---
$schema: "http://json-schema.org/schema#"
$id: "https://github.com/lfit/releng-global-jjb/blob/master/release-container-schema.yaml
↪"

required:
  - "containers"
  - "distribution_type"
  - "project"
  - "container_release_tag"
  - "ref"

properties:
  containers:
    type: "array"
    properties:
      name:
        type: "string"
      version:
        type: "string"
    additionalProperties: false
  container_release_tag:
    type: "string"
  container_pull_registry:
    type: "string"
  container_push_registry:
    type: "string"
  distribution_type:
    type: "string"
  git_tag:
    type: "string"
  project:
    type: "string"
  ref:
    type: "string"
  tag_release:
    type: "boolean"
```

3.14.3 PyPI Release Files

An example of a PyPI release file appears below. Name of the release file must start with "pypi". For example releases/pypi-1.0.0-mypackage.yaml

```
$ cat releases/pypi-1.0.0-mypackage.yaml
---
pypi_project: mypackage
python_version: '3.4'
version: 1.0.0
log_dir: example-project-pypi-merge-master/17
```

The following parameters must appear in the PyPI release yaml file. These are not part of the Jenkins job definition to allow independent self-release of a package maintained in a git repository with other packages.

Required Parameters

log_dir The suffix of the logs URL reported on successful completion by the Jenkins merge job that created and pushed the distribution files to the staging repository. For example, use value "example-project-pypi-merge-master/17" for the logs URL https://logs.lf-project.org/production/vex-sjc-lfp-jenkins-prod-1/example-project-pypi-merge-master/17

pypi_project The PyPI project name at the staging and release repositories, for example "mypackage".

python_version The Python interpreter version to use for pip "Requires-Python" compatibility checks, for example '3', '3.7' or 3.7.4. Put valid decimal values such as 3 or 3.7 in quotes to pass schema validation.

version The semantic version string used for the package in the setup.py file.

Optional Parameters

git_tag The tag string to sign and push to the Git repository. (default: the semantic version string)

tag_release Tag Gerrit Repo during the release process. (default: true)

The JSON schema for a PyPI release file appears below.

```
# SPDX-License-Identifier: EPL-1.0
##############################################################################
# Copyright (c) 2019 The Linux Foundation and others.
#
# All rights reserved. This program and the accompanying materials
# are made available under the terms of the Eclipse Public License v1.0
# which accompanies this distribution, and is available at
# http://www.eclipse.org/legal/epl-v10.html
##############################################################################
---
$schema: "http://json-schema.org/schema#"
$id: "https://github.com/lfit/releng-global-jjb/blob/master/release-pypi-schema.yaml"

required:
  - "log_dir"
  - "pypi_project"
  - "python_version"
```

(continues on next page)

(continued from previous page)

```
  - "version"

properties:
  git_tag:
    type: "string"
  log_dir:
    type: "string"
  pypi_project:
    type: "string"
  python_version:
    type: "string"
  tag_release:
    type: "boolean"
  version:
    type: "string"
```

3.14.4 PackageCloud Release Files

An example of a PackageCloud release file appears below. Name of release file must start with "packagecloud". For example releases/packagecloud-1.6-tree.yaml

```
$ cat releases/packagecloud-1.6-tree.yaml
---
package_name: tree
packages:
    - name: tree_1.6.0_amd64.deb
    - name: tree-dev_1.6.0_amd64.deb
    - name: tree-devel-1.6.0-1.x86_64.rpm
    - name: tree-1.6.0-1.x86_64.rpm
ref: 5555cd2dd345fbeec0d3e2162e00835852342cda
log_dir: example-project-packagecloud-merge/21
version: 1.6.0
```

The following parameters must appear in the PackageCloud release yaml file. These are not part of the Jenkins job definition to allow independent self-release of a package maintained in a git repository with other packages.

Required Parameters

package_name Name of the release package.

packages A list of names that specify the packages to promote. Found in jenkins console log when using gem to push package eg. "Pushing /path/of/package/name-of-package.rpm... success!" OR using rest api call to query packagecloud.io repo "curl https://packagecloud.io/api/v1/repos/test_user/test_repo/search?q= | yq -r .[].filename"

ref The git commit reference (SHA-1 code) to tag with the version string.

log_dir The suffix of the logs URL reported on successful completion by the Jenkins merge job that created and pushed the distribution files to the staging repository. For example, use value "example-project-packagecloud-merge-/21" for the logs URL https://logs.lf-project.org/production/vex-sjc-lfp-jenkins-prod-1/example-project-packagecloud-merge/21

version The semantic version string used for the package.

Optional Parameters

git_tag The tag string to sign and push to the Git repository. (default: the semantic version string)

tag_release Tag Gerrit Repo during the release process. (default: true)

The JSON schema for a PackageCloud release file appears below.

```
# SPDX-License-Identifier: EPL-1.0
##############################################################################
# Copyright (c) 2019 The Linux Foundation and others.
#
# All rights reserved. This program and the accompanying materials
# are made available under the terms of the Eclipse Public License v1.0
# which accompanies this distribution, and is available at
# http://www.eclipse.org/legal/epl-v10.html
##############################################################################
---
$schema: "http://json-schema.org/schema#"
$id: "https://github.com/lfit/releng-global-jjb/blob/master/release-packagecloud-schema"

required:
  - "package_name"
  - "packages"
  - "ref"
  - "log_dir"
  - "version"

properties:
  package_name:
    type: "string"
  packages:
    type: "array"
    properties:
      name:
        type: "string"
  git_tag:
    type: "string"
  ref:
    type: "string"
  log_dir:
    type: "string"
  tag_release:
    type: "boolean"
  version:
    type: "string"
```

3.15 Job Groups

Below is a list of Release job groups:

```
---
- job-group:
    name: "{project-name}-gerrit-release-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for any project using self releases.

    jobs:
      - gerrit-release-verify
      - gerrit-release-merge

- job-group:
    name: "{project-name}-github-release-jobs"

    # This job group contains all the recommended jobs that should be deployed
    # for any project using self releases.

    jobs:
      - github-release-verify
      - github-release-merge
```

3.15.1 Jenkins Jobs

An example of a Jenkins job configuration that uses the global-jjb templates for maven and container release jobs appears next.

```
- project:
    name: my-project-release
    project: my-project
    project-name: my-project
    build-node: centos7-docker-4c-4g
    mvn-settings: my-project-settings
    jobs:
        - '{project-name}-gerrit-release-jobs'
```

Note: Release Engineers: please follow the setup guide below before adding the job definition.

3.15.2 JJB Macros

lf-release

Release verify and merge jobs are the same except for their scm, trigger, and builders definition. This anchor is the common template.

3.15.3 JJB Templates

Release Merge

This template supports Maven and Container release jobs.

This template uses a git commit choosing strategy that builds the merged commit with the release yaml file, not the tip of the target branch, so projects can repeat the release action in case of merge job failure.

Template Name {project-name}-release-merge-{stream}

Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-release-credential Credential to use for SSH. (Generally set in defaults.yaml)

project Git repository name

project-name Jenkins job name prefix

Optional parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. The default pattern is the regular expression `(releases\/.*\.yaml|\.releases\/.*\.yaml)`

Release Verify

This template supports Maven and Container release jobs.

Template Name {project-name}-release-verify-{stream}

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-node The node to run build on.

build-timeout Timeout in minutes before aborting build. (default: 15)

gerrit-skip-vote Skip voting for this job. (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

stream Keyword that represents a release code-name. Often the same as the branch. (default: master)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. The default pattern is the regular expression `(releases\/.*\.yaml|\.releases\/.*\.yaml)`

PyPI Release Merge

Publishes a Python package on merge of a patch set with a release yaml file. Checks the format of the version string, downloads the package artifacts from the PyPI staging repository, uploads the package artifacts to the PyPI release repository, tags the git repository, signs the tag and pushes the tag to the git server. The release merge template accepts neither a branch nor a stream parameter.

These templates use a git commit choosing strategy that builds the merged commit with the release yaml file, not the tip of the target branch, so projects can repeat the release action in case of merge job failure.

Template Names

- {project-name}-pypi-release-merge
- gerrit-pypi-release-merge
- github-pypi-release-merge

Comment Trigger remerge

Required Parameters

build-node The node to run build on, which must be Centos.

jenkins-ssh-release-credential Credential to use for SSH. (Generally set in defaults.yaml)

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pypi-stage-index Base URL of the PyPI staging repository. (default https://test.pypi.org/simple)

pypi-repo Key for the PyPI release repository in the .pypirc file, should be the repository pypy.org. (default: pypi)

use-release-file Whether to use the release file. (default: true)

gerrit_release_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. The default pattern is the regular expression `(releases\/pypi.*\.yaml|\.releases\/pypi.*\.yaml)`

PyPI Release Verify

Verifies a Python package project on creation of a patch set with a release yaml file. Checks the contents of the release yaml file, checks the format of the version string, and downloads the release artifacts from the specified PyPI staging repository. The release verify template accepts neither a branch nor a stream parameter.

Template Names

- {project-name}-pypi-release-verify
- gerrit-pypi-release-verify
- github-pypi-release-verify

Comment Trigger recheck

Required Parameters

build-node The node to run build on, which must be Centos.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

pypi-stage-index Base URL of the PyPI staging repository. (default https://test.pypi.org/simple)

pypi-repo Key for the PyPI release repository in the .pypirc file, should be the repository pypy.org (default: pypi)

use-release-file Whether to use the release file. (default: true)

gerrit_release_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. The default pattern is the regular expression `(releases\/pypi.*\.yaml|\.releases\/pypi.*\.yaml)`

PackageCloud Release Verify

This template supports PackageCloud release jobs. Checks that the specified packages are present in the staging repository and absent from the release repository. The file path trigger uses the regular expression `(releases\/packagecloud.*\.yaml|\.releases\/packagecloud.*\.yaml)`

Template Name {project-name}-packagecloud-release-verify

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

project Git repository name

project-name Jenkins job name prefix

Optional Parameters

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-node The node to run build on.

build-timeout Timeout in minutes before aborting build. (default: 15)

gerrit-skip-vote Skip voting for this job. (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

PackageCloud Release Merge

This template supports PackageCloud release jobs. Promotes the specified packages from the staging repository to the release repository. The file path trigger uses the regular expression `(releases\/packagecloud.*\.yaml|\.releases\/packagecloud.*\.yaml)`

This template uses a git commit choosing strategy that builds the merged commit with the release yaml file, not the tip of the target branch, so projects can repeat the release action in case of merge job failure.

Template Name {project-name}-packagecloud-release-merge

Comment Trigger remerge

Required Parameters

build-node the node to run build on.

jenkins-ssh-release-credential credential to use for ssh. (generally set in defaults.yaml)

project git repository name

project-name jenkins job name prefix

Optional Parameters

build-days-to-keep days to keep build logs in jenkins. (default: 7)

build-timeout timeout in minutes before aborting build. (default: 15)

3.15.4 Setup for LFID, Nexus, Jenkins and Gerrit

This section is for the Linux Foundation release engineering team.

LFID

Create an `lfid` and an `ssh-key`

`YOUR_RELEASE_USERNAME` for example: onap-release

`YOUR_RELEASE_EMAIL` for example: collab-it+onap-release@linuxfoundation.org

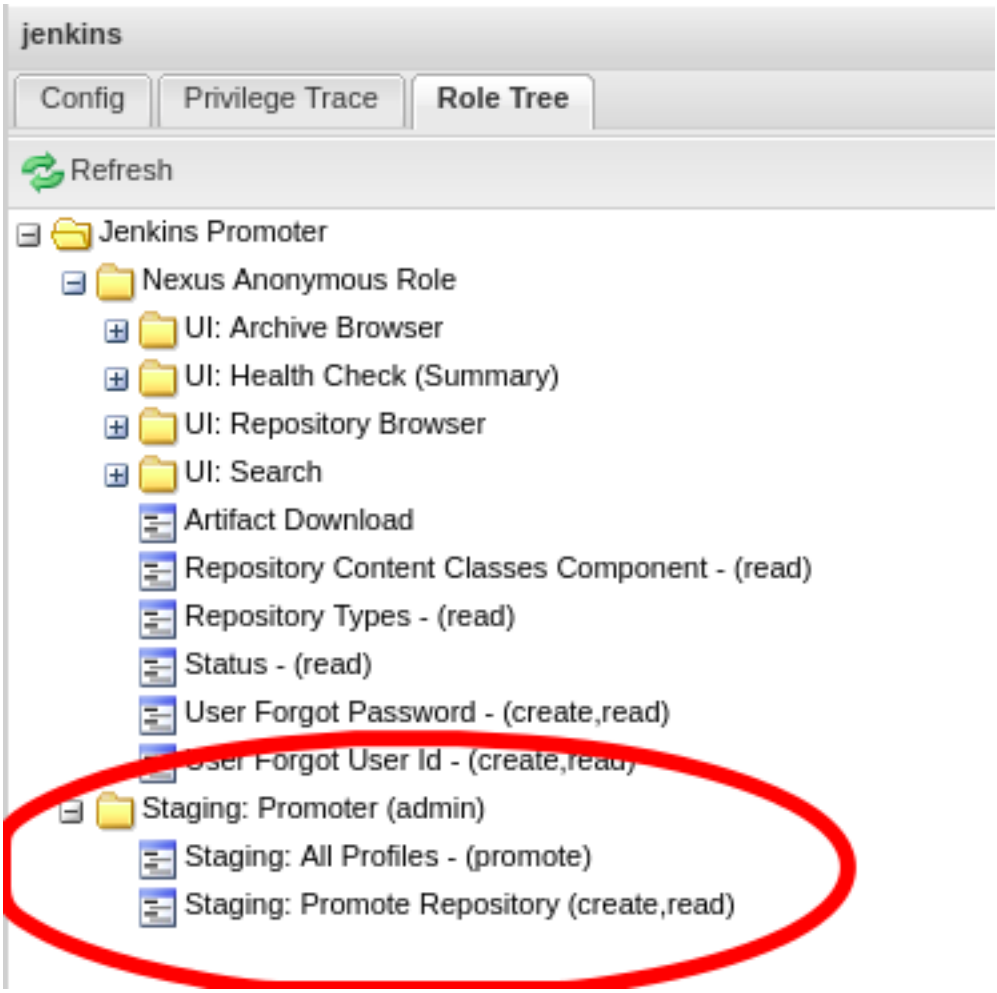
ssh-key example:

```
ssh-keygen -t rsa -C "collab-it+odl-release@linuxfoundation.org"  -f /tmp/odl-release
```

Create an LFID with the above values

Nexus

Create a Nexus account called `'jenkins-release'` with promote privileges.



Gerrit

Log into your Gerrit with `YOUR_RELEASE_USERNAME`, upload the public part of the `ssh-key` you created earlier. Log out of Gerrit and log in again with your normal account for the next steps.

In Gerrit create a new group called `self-serve-release` and give it direct push rights via `All-Projects` Add `YOUR_RELEASE_USERNAME` to group `self-serve-release` and group `Non-Interactive Users`

In All project, grant group self-serve-release the following:

```
[access "refs/heads/*"]
  push = group self-serve-release
[access "refs/tags/*"]
  createTag = group self-serve-release
  createSignedTag = group self-serve-release
  forgeCommitter = group self-serve-release
  push = group self-serve-release
```

Jenkins

Add a global credential to Jenkins called `jenkins-release` and set the ID: `'jenkins-release'` as its value insert the private half of the `ssh-key` that you created for your Gerrit user.

Add Global variables in Jenkins: Jenkins configure -> Global properties -> Environment variables:

```
RELEASE_USERNAME = YOUR_RELEASE_USERNAME
RELEASE_EMAIL = YOUR_RELEASE_EMAIL
```

Note: Add these variables to your global-vars-$SILO.sh file or they will be overwritten.

Jenkins configure -> Managed Files -> Add a New Config -> Custom File

```
id: signing-pubkey
Name: SIGNING_PUBKEY (optional)
Comment: SIGNING_PUBKEY (optional)

Content: (Ask Andy for the public signing key)
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

Add or edit the managed file in Jenkins called `lftoolsini`, appending a nexus section: Jenkins Settings -> Managed files -> Add (or edit) -> Custom file

```
[nexus.example.com]
username=jenkins-release
password=<plaintext password>
```

Ci-management

Upgrade your project's global-jjb if needed, then add the following to your global defaults file (e.g., jjb/defaults.yaml).

```
jenkins-ssh-release-credential: jenkins-release
```

— Job Templates ==========

3.16 Release Announce

Job for lf-releng projects to automate release announcement emails.

Template Names

- {project-name}-release-announce
- gerrit-release-announce

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Configured in defaults.yaml)

3.17 ReadTheDocs Jobs

3.17.1 Job Groups

Job groups are a great tool to configure categories of jobs together at the same time. Below the example are some starting point job-groups but we recommend creating your own to ensure that the jobs configured reflect the project's needs.

An example project:

```
- job-group:
    name: odl-maven-jobs

    jobs:
      - gerrit-maven-clm
      - gerrit-maven-merge
      - gerrit-maven-release
      - gerrit-maven-verify
      - gerrit-maven-verify-dependencies:
          build-timeout: 180

    mvn-version: mvn35

- project:
    name: aaa
    jobs:
      - odl-maven-jobs
```

In this example we are using the job-group to assign a list of common jobs to the aaa project. The job-group also hardcodes `mvn-version` to *mvn35* and `build-timeout` to *180* for all projects using this job-group.

A benefit of this method is for example disabling entire category of jobs by modifying the job-group, insert `disable-job: true` parameter against the jobs to disable.

Below is a list of Maven job groups:

```
---
- job-group:
    name: "{project-name}-rtd-jobs"

    jobs:
      - gerrit-rtd-merge
      - gerrit-rtd-verify

- job-group:
    name: "{project-name}-github-rtd-jobs"

    jobs:
      - github-rtd-merge
      - github-rtd-verify
```

3.17.2 Macros

lf-rtd-common

RTD verify and merge jobs are the same except for their scm, trigger, and builders definition. This anchor is the common template.

3.17.3 Job Templates

ReadTheDocs Merge

Merge job which triggers a POST of the docs project to readthedocs.

To use this job first configure the `Generic API incoming webhook` in ReadTheDocs. To do that follow these steps:

1. Browse to https://readthedocs.org/dashboard/PROJECT/integrations/
2. Click on `Generic API incoming webhook`

   Note: If not available click on `Add integration` and add the `Generic API incoming webhook`.

3. Copy the custom webhook URL, this is your `rtd-build-url`

   For example: https://readthedocs.org/api/v2/webhook/opendaylight/32321/

4. Copy the token, this is your `rtd-token`

   Template Names

   - {project-name}-rtd-merge-{stream}
   - gerrit-rtd-merge
   - github-rtd-merge

   Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

rtd-build-url This is the generic webhook url from readthedocs.org. Refer to the above instructions to generate one. (Check Admin > Integrations > Generic API incoming webhook)

rtd-token The unique token for the project Generic webhook. Refer to the above instructions to generate one. (Check Admin > Integrations > Generic API incoming webhook)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

git-url base URL of git project. (default: https://github.com)

project-pattern Project to trigger build against. (default: **)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. **default**:

```
- compare-type: ANT
  pattern: '**/*.rst'
- compare-type: ANT
  pattern: '**/conf.py'
```

ReadTheDocs Verify

Verify job which runs tox to test the docs project

Template Names

- {project-name}-rtd-verify-{stream}
- gerrit-rtd-verify
- github-rtd-verify

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-node The node to run build on.

build-timeout Timeout in minutes before aborting build. (default: 15)

doc-dir Directory where tox will place built docs. as defined in the tox.ini (default: docs/_build/html)

gerrit-skip-vote Skip voting for this job. (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

project-pattern Project to trigger build against. (default: **)

python-version Python version (default: python2)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. **default**:

```
- compare-type: ANT
  pattern: '**/*.rst'
- compare-type: ANT
  pattern: '**/conf.py'
```

3.18 ReadTheDocs Version:3 Jobs

ReadTheDocs supports the nesting of projects, by configuring a project as a subproject of another project. This allows for documentation projects to share a search index and a namespace or custom domain, while still maintained independently of each other.

The master Read The Docs project files, maintained in a "docs" Git repository should contain an index with links to all the sub-projects. Each sub-project must maintain its documentation files in a "docs" subdirectory within that software component's Git repository.

The RTDv3 Jenkins jobs publish documentation by triggering builds at ReadTheDocs.io. That build process clones the appropriate repository and transforms reStructuredText (RST) and other files into HTML. All project's Read the Docs builds separately from sub-project builds.

The Read The Docs site supports versioned documentation for the master project and every sub-project. Every project should have a development branch that's published at ReadTheDocs under the title "latest"; in Git this is the "master" branch although can be different in some projects. Most projects also declare releases periodically. ReadTheDocs automatically detects the creation of git branches and git tags, and publishes the most recent one under the title "stable." For more details please see ReadTheDocs Versions. Teams can control this process using Jenkins job configuration parameters as discussed below.

3.18.1 User setup

To transform your rst documentation into a Read The Docs page, configure as described in Admin setup below. Once this is complete, add the following files to your repository:

```
.readthedocs.yaml
tox.ini
docs
docs/_static
docs/_static/logo.png
docs/conf.yaml
docs/favicon.ico
docs/index.rst
docs/requirements-docs.txt
docs/conf.py
```

Rather than copying and pasting these files from a set of docs here, the following repo contains a script that will do this for you. Please refer to the explanation presented in: <https://github.com/lfit-sandbox/test>. This is a beta feature, so please send feedback on your experiences. Once complete, the script `docs_script.sh` is not needed. You can copy the files by hand if you prefer.

The default location of the tox.ini file is in the git repository root directory. Optionally your documentation lead may decide to store all tox files within the required "docs" subdirectory by setting configuration option "tox-dir" to value "docs/" as discussed below.

If your project's tox dir is `docs/` and not `.`, update the tox.ini configuration with the correct relative paths.

You must also set the doc-dir. For example, from the default of `doc-dir: "docs/_build/html"` to `doc-dir: "_build/html"`, as the relative path in the tox run has changed.

Once configured, in your repository you can build the rst files locally to test:

```
tox -e docs,docs-linkcheck
```

3.18.2 Stable Branch Instructions

If your project does not create branches, you can skip this step.

For Read The Docs to see your new branch, trigger a build to force RTD to run an update. Use a trivial change to any file in your project's `/docs/` directory on your newly minted branch to trigger a build and activate your project's new branch on Read The Docs. This will create a new selectable version in the bottom right corner of your project's Read The Docs page. Once all projects have branched the process to release the documentation (that is to change the default landing point of your docs from /latest/ to /branchname/) is to change the default-version in the jenkins job config as follows:

From:

```
default-version: latest
```

To:

```
default-version: ReleaseBranchName
```

3.18.3 Admin setup:

This part of the documentation explains how to enable this job so that It will trigger on docs/* changes for all projects in a Gerrit instance. It leverages the Read The Docs v3 api to create projects on the fly, as well as setting up sub-project associations with the master doc.

A `.readthedocs.yaml` must exist in the root of the repository otherwise the jobs will run but skip actual verification.

Define the master doc in jenkins-config/global-vars-{production|sandbox}.sh

This project named "doc" or "docs" or "documentation" will set all other docs builds as a subproject of this job.

examples:

```
global-vars-sandbox.sh:
MASTER_RTD_PROJECT=doc-test
global-vars-production.sh:
MASTER_RTD_PROJECT=doc
```

In this way sandbox jobs will create docs with a test suffix and will not stomp on production documentation.

Example job config:

example file: ci-management/jjb/rtd/rtd.yaml

```
---
- project:
    name: rtdv3-global-verify
    build-node: centos7-builder-1c-1g
    default-version: latest
    tox-dir: "."
    doc-dir: "docs/_build/html"
    jobs:
      - rtdv3-global-verify
    stream:
      - master:
          branch: master
      - foo:
          branch: stable/{stream}

- project:
    name: rtdv3-global-merge
    default-version: latest
    tox-dir: "."
    doc-dir: "docs/_build/html"
    build-node: centos7-builder-1c-1g
    jobs:
      - rtdv3-global-merge
    stream:
      - master:
          branch: master
      - foo:
          branch: stable/{stream}
```

Or add both jobs via a job group: This real-world example also shows how to configure your builds to use a tox.ini that lived inside your docs/ dir

```
# Global read the docs version 3 jobs
#
# jobs trigger for all projects, all branches
# on any changes to files in a docs/ directory
# and publish subprojects to readthedocs.io
# using credentials from Jenkins settings
---
- project:
    name: rtdv3-view
    project-name: rtdv3-global
    views:
      - project-view

- project:
    name: rtdv3-global
    default-version: latest
    tox-dir: "docs/"
    doc-dir: "_build/html"
    build-node: centos7-builder-2c-1g
    # override the default to ignore ref-updated-event (tag)
    gerrit_merge_triggers:
      - change-merged-event
      - comment-added-contains-event:
          comment-contains-value: remerge$
    jobs:
      - rtdv3-global-verify
      - rtdv3-global-merge
    stream:
      - master:
          branch: '*'
```

GitHub jobs must be per-project. Once proven, a different set of jobs will be available.

Job requires an lftools config section, this is to provide api access to read the docs.

```
[rtd]
endpoint = https://readthedocs.org/api/v3/
token = [hidden]
```

Merge Job will create a project on Read The Docs if none exist. Merge Job will assign a project as a subproject of the master project. Merge job will trigger a build to update docs. Merge job will change the default version if needed.

Macros

3.18.4 lf-rtdv3-common

RTD verify and merge jobs are the same except for their scm, trigger, and builders definition. This anchor is the common template.

Job Templates

3.18.5 ReadTheDocs v3 Merge

Merge job which triggers a build of the docs to Read The Docs.

Template Names

- rtdv3-global-merge-{stream}

Comment Trigger remerge

Required parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

default-version default page to redirect to for documentation (default /latest/)

disable-job Whether to disable the job (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

project-pattern Project to trigger build against. (default: **)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's Read The Docs tox.ini

doc-dir Relative directory project's docs generated by tox

gerrit_merge_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. **default**:

```
- compare-type: REG_EXP
  pattern: '^docs\/.*'
```

3.18.6 ReadTheDocs v3 Verify

Verify job which runs a tox build of the docs project. Also outputs some info on the build.

Template Names

- rtdv3-global-verify-{stream}

Comment Trigger recheck|reverify

Required Parameters

build-node The node to run build on.

jenkins-ssh-credential Credential to use for SSH. (Generally set in defaults.yaml)

Optional Parameters

branch Git branch to fetch for the build. (default: master)

build-days-to-keep Days to keep build logs in Jenkins. (default: 7)

build-timeout Timeout in minutes before aborting build. (default: 15)

gerrit-skip-vote Skip voting for this job. (default: false)

git-url URL clone project from. (default: $GIT_URL/$PROJECT)

disable-job Whether to disable the job (default: false)

project-pattern Project to trigger build against. (default: **)

stream Keyword representing a release code-name. Often the same as the branch. (default: master)

submodule-recursive Whether to checkout submodules recursively. (default: true)

submodule-timeout Timeout (in minutes) for checkout operation. (default: 10)

submodule-disable Disable submodule checkout operation. (default: false)

tox-dir Directory containing the project's Read The Docs tox.ini

doc-dir Relative directory project's docs generated by tox

gerrit_verify_triggers Override Gerrit Triggers.

gerrit_trigger_file_paths Override file paths filter which checks which file modifications will trigger a build. **default**:

```
- compare-type: REG_EXP
  pattern: '^docs\/.*'
```

3.19 Jenkins Views

3.19.1 View Templates

JJB view-templates provides a way to manage Jenkins views through code. Using view-templates we can define common views configuration that are interesting to a project.

We recommend creating separate project sections for views apart from job configuration such that job configuration does not overlap with the view configuration.

Example Usage:

```
---
- project:
    name: project-view
    views:
      - common-view

    project-name: project

- project:
    name: project-stream1
    jobs:
      - '{project-name}-{seq}'

    project: project
    project-name: project
    seq:
      - a
      - b

- project:
    name: project-stream2
    jobs:
      - '{project-name}-{seq}'

    project: project
    project-name: project
    seq:
      - x
      - y

- job-template:
    name: '{project-name}-{seq}'
```

Project view

Groups all jobs owned by a project under one view by capturing jobs with the prefix of `project-name`.

This view uses the following columns:

Columns

- status
- weather
- job
- last-success
- last-failure
- last-duration
- build-button
- jacoco
- find-bugs

Template Names

- {project-name}
- project-view

Required parameters

project-name The name of the project utilizing the view.

Optional parameters

view-filter-executors View filter executor. (default: false)

view-filter-queue View filter queue. (default: false)

view-recurse View recurse. (default: false)

Example:

```
---
- project:
    name: project-view-test
    views:
      - project-view

    project-name: project-view-test
```

Common view

Groups all jobs owned by a project under one view by capturing jobs with the prefix of `project-name`.

This view uses the following columns:

Columns

- status
- weather
- job
- last-success
- last-failure
- last-duration
- build-button
- jacoco
- find-bugs

Template Names

- {view-name}
- common-view

Required parameters

view-name The name of the view.

view-regex Regex to match the jobs.

Optional parameters

view-filter-executors View filter executor. (default: false)

view-filter-queue View filter queue. (default: false)

view-recurse View recurse. (default: false)

Example:

```
---
- project:
    name: common-view-test
    views:
      - common-view

    view-name: Daily
    view-regex: ".*-daily-.*"
```

CSIT view template

View template that loads columns useful for CSIT jobs.

This view uses the following columns:

Columns

- status
- weather
- job
- last-success
- last-failure
- last-duration
- build-button
- robot-list

Template Names

- {view-name}
- csit-view

Required parameters

view-name The name of the view.

view-regex Regex to match the jobs.

Optional parameters

view-description View description. (default: 'CSIT Jobs.')

view-filter-executors View filter executor. (default: false)

view-filter-queue View filter queue. (default: false)

view-recurse View recurse. (default: false)

Example:

```
---
- project:
    name: csit-view-test
    views:
      - csit-view

    view-name: CSIT-1node
    view-regex: ".*csit-1node.*"
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

C

J