## ARC 6 the source in GitLab

ARC 6 Retreat Umeå 07.11-09.11 2018

## ARC source code and packages

ARC 6 on GitLab hosted by Coderefinery:

• <a href="https://source.coderefinery.org/nordugrid/arc">https://source.coderefinery.org/nordugrid/arc</a>

Some documentation available

<u>http://www.nordugrid.org/documents/arc6/index.html</u>

How to contribute to ARC

• <a href="https://source.coderefinery.org/nordugrid/arc/wikis/Contributing/Cheat-Sheet">https://source.coderefinery.org/nordugrid/arc/wikis/Contributing/Cheat-Sheet</a>

Versions

- For a surely stable versions, get a release tag first will be v6.0.0a1 an alpha release
- master branch also expected to be (more or less) stable, but not guaranteed at all times

## ARC from nightly rpm's or GitLab's Cl

- For latest automatic build (Centos 6/7 or Debian 9)
  - wget 'https://source.coderefinery.org/nordugrid/arc/-/jobs/artifacts/master/download?job=<job-os-type>' -O artifacts.zip
  - <job-os-type>: make\_rpms\_el6 or make\_rpms\_el7 or make\_debs\_deb9
  - Unzip artifacts.zip
  - Install the rpm's some localinstall (e.g. yum localinstall)
- For nightlies w/all supported platforms:
  - <u>http://download.nordugrid.org/builds/index.php?pkgname=nordugrid-arc&type=master</u>
  - Suggested procedure: <u>http://www.nordugrid.org/documents/arc6/developer/nightlies-repo</u> <u>repo.html#nightlies-repo</u>

#### ARC 6 from repo

## ARC build from source

Install some necessary/useful packages for building from source (you probably have these installed already)

```
yum install -y epel-release gettext-devel cppunit
```

```
yum install -y yum-utils --enablerepo=extras
```

```
yum groupinstall -y 'Development tools'
```

```
Might not be necessary:
yum-builddep nordugrid-arc -y
```

Since ARC 6 is not yet available, the spec-file is not in epel, therefore to install build-dependencies for ARC 6 you will need to configure first, then run yum-builddep on the produced nordugrid-arc.spec file

```
./autogen.sh
```

```
./configure
```

```
yum-builddep nordugrid-arc.spec -y
```

## Our Git Branch model

- master branch for releases of current release series (currently 6.x.y)
  - A release is a tag on the master branch
  - Everything currently on master will be released
- next branch for releases of next major release series (currently 7.x.y)
  - All commits to master also to next-major
  - Holds also non-backward compatible commits with current release series
  - «next» branch will be merged into master at the time of new major release
- When needed, more semi-stable development branches will be created
  - When there is a longer term ongoing specific development effort involving several people

#### Releases

- Earlier an ARC release would be called e.g. 15.03u18 and contain
  - ARC version e.g. 5.4.2
  - Gangliarc version 1.0.2
  - Nagios probe version 1.9.0
  - Metapackages (client tools, computing element and information index,) 1.0.7
  - Docs 2.0.15
- Now ARC is not released as a bundle with nagios probes and gangliarc
  - ARC version 6.0.0
  - Metapackages 2.0.0
  - gangliarc and nagios probes have their own repo and own release cycle
  - Docs for ARC 6 will be available on the web in its latest version
- No distinction between bugfix and minor release anymore last digit in ARC version will always be 0 until we have migrated to a two-digit release number
  - Everything on master at time of released will be release.
  - Master will not contain backward incompatible changes which would trigger a new major release

## Contributing

https://source.coderefinery.org/nordugrid/arc/wikis/contributing/cheat-sheet

- 1. Update your main branches on your fork
- 2. Decide parent branch
- 3. Create dev-branch
- 4. Develop
- 5. Update dev-branch to parent branch
- 6. Repeat
- When done submit merge request(s)

	Case	branch from (parent)	merge into	mr branch labels opt 1	mr branch labels opt 2
1	Changes for current release series and for next releases series	master	master next-major	copy_mr:next-major	-
2	Changes for current release series but not for next release series	master	master	copy:none	-
3	Changes for next release series, but <b>not for</b> current release series	next-major	next-major	copy:none	-

### Merge request

#### D7. Create merge request

Note! Once a merge request has been made, all following commits are recorded into the merge request. You should therefore create a new branch to work on a different task.

#### **Option 1 (Simplest)**

Web interface and arcbot

- Make sure you have added the arcbot as member to your fork. See instructions in point G2.
- Go to https://source.coderefinery.org/<your-user-name>/arc/merge\_requests/new
  - Or navigate from your fork Merge Requests -> New Merge Request
- Source branch: <your-user-name>/arc <your-dev-branch>
- Target branch: nordugrid/arc <parent-branch> (See table of rules for choosing correct target.)
- Select template from drop-down menu and follow instructions
- If you want your changes also to be merged into another branch by arcbot, select label copy\_mr:<otherbranch> (See table of rules for choosing correct label.)

Find more details for nordugrid-arc-bot here.

#### Example of tree with this workflow

Merge branch 'revert-1ded0585' into 'master' master 🚬 Revert "Merge branch 'issue-21-controldir-permissions' into 'master'" Merge branch 'localplugin-fixes' into 'master' Merge branch 'issue-21-controldir-permissions' into 'master' 🏁 Adding proper failure message if arexjob fails. And removing some left-over debug statements. 🔀 Changing controldir permission requirement from 755 to 700 🐹 Merge branch 'gm-jobs-stderr' into 'master' Somon log format for arcctl and gm-jobs 🥵 gm-jobs now handle debug level, passing it from arcctl Merge branch 'cherrypick-80-for-master' into 'master' Merge branch 'ENV-PROXY' into 'master' Merge branch 'dev-totalcpus' into 'master' 📜 Add description of new arcsync option into man page. Making sqlite default option for jobs storage. Adding jobs storage conversion option to arcsync. ENV/RTE description 🔞 RTEs packaging 🚱 ENV/PROXY updates + autoconf 🕞 Fix for bug 3626 - totalcpus override LRMS modules propagated to gueues, gueues can be overridden indivi 🔂 Fix for bug 3626, override ComputingManager totalcpus in GLUE2. Not complete, should check how queues ar

Merge branch 'dev-bugfix-6-certs' into 'master'

#### Extras

# Comparison old (svn) and new (git) branch model and workflow I

#### Goal with new branching model and workflow

- improve release procedure quality and efficiency
- make it as simple as possible for developers

#### Old branching model in svn

- All developments go to trunk (also backward incompatible)
- Commits in trunk are ported to release branch one branch per major cycle
  - Release manager ports all commits in preparation of release
  - Tedious sorting out of merge conflicts since they happen when release manager ports commits from trunk to release branch (and not as an action initiated by developer)
- Tag branch created from release branch for each release

#### Problems with old model

- No clear way to mark backward incompabitle changes
- For each release list of all commits to trunk not already in release developers asked to say yay or nay for adding into upcoming release
  - Tedious, easy to forget that one commit should wait for next release or next major etc etc
  - Same commit comes back in the discussion since we forget from last time
  - Workload rather large and manual just before release

## Comparison old (svn) and new (git) branch model and workflow II

Improvements from old model

- Simplified branch model all commits on master go into release minor or bugfix release depending on content of commits since last release
- Release tags directly on master commits for release continously added instead of right before release
- Merge conflicts and other problems sorted out by developer at time of creating/handling merge request (not just right before release)
- Issues and merge request can be labelled for sorting of types e.g. «backward incompatible» label
- Gitlabs CI gives developer automatic check of whether code builds

# Merges to several branches – evolution of nordugrid-arc-bot

- Each merge request to master  $\rightarrow$  also to «next» branch
- Each merge request to «next» only to «next» (otherwise use master as parent)
- First attempt of working model: developer creates two merge requests
  - Complaints
- Second attempt introduced arcbot
  - Webhook set up triggered by merges.
  - Webservice that duplicates the merge request merges same development branch into other target branch
  - Unwanted result contains more commits than original merge request since the diff is different between master and nextmajor
- Third attempts cherrypicking instead
  - Cherrypick commits of the original merge request
  - Create branch with these cherrypicked commits
  - Create merge request of this cherrypick branch
  - Not perfect solution:
    - Commits get different commit has more of a hassle to compare branches
- Fourth attempt revert back to Second attempt

## Basic use – merge conflict

- Developer should solve the merge conflicts in most cases (unless it is related to the git-files gitlabci.yml or template files)
- If a merge request triggers a merge conflict, update your fork with master and solve the conflict
- git status to see what files are the problem
- Edit file to fix merge conflict
- Add, commit, push
- Commits to the source branch (your fork branch) as a result of solving the merge conflicts are recorded automatically to the merge request
- If you want their version of the file:
  - https://git-scm.com/docs/git-checkout/2.5.1

#### git accept theirs version of conflict

```
git checkout --theirs <file-path>
git add <file-path>
git commit
```

### To-do list

- Make small scripts for everyday contribution
  - Create-new-dev-branch
  - Update
  - Push
  - ...
- Extend testing
  - Find testing strategy and implement
- Do more real release-test with creation of release notes based on our labelling and merge requests and issues