

Tailor CI: How Locus ships robots at scale

Paul Bovbel Gary Servin



What does scale mean anyway?

- Some number of sites A
- Some number of robots B
- Some variety of software releases C





madness $\sim = O(A * log(B) * C)$

The Old Way

Some combination of

- packaging via ros_buildfarm
- incremental deploys via ansible
- source overlays

Abi	n_uX64	Adev Adoc All Apr Asrc_uX Bbin_uX64 Bdev Bdoc Bpr Bsrc_uX
S	w	Name ↓
	*	Abin_uX64_actionlib_ubuntu_xenial_amd64_binary
	*	Abin_uX64_actionlib_msgs_ubuntu_xenial_amd64_binary
	*	Abin_uX64amclubuntu_xenial_amd64binary
0	*	Abin_uX64anglesubuntu_xenial_amd64binary
0	*	Abin_uX64_apriltags_ubuntu_xenial_amd64_binary
	*	Abin_uX64apriltags_cppubuntu_xenial_amd64binary
	*	Abin_uX64base_local_plannerubuntu_xenial_amd64binary
	*	Abin_uX64bflubuntu_xenial_amd64binary
0	*	Abin_uX64bondubuntu_xenial_amd64binary
0	*	Abin_uX64bond_coreubuntu_xenial_amd64binary
0	*	Abin_uX64bondcppubuntu_xenial_amd64binary
0	*	Abin_uX64bondpyubuntu_xenial_amd64binary
0	*	Abin_uX64camera_calibrationubuntu_xenial_amd64binary
	*	Abin_uX64camera_calibration_parsers_ubuntu_xenial_amd64binary



The Old Way

Downsides:

- no rolling release or nightlies
- slow top-to-bottom builds
- release state is distributed amongst many repositories
- incremental deploys are flaky

Abi	n_uX64	Adev Adoc All Apr Asrc_uX Bbin_uX64 Bdev Bdoc Bpr Bsrc_uX
S	w	Name ↓
	*	Abin_uX64_actionlib_ubuntu_xenial_amd64_binary
	*	Abin_uX64actionlib_msgsubuntu_xenial_amd64binary
0	*	Abin_uX64amclubuntu_xenial_amd64binary
0	*	Abin_uX64anglesubuntu_xenial_amd64binary
0	*	Abin_uX64_apriltags_ubuntu_xenial_amd64_binary
0	*	Abin_uX64apriltags_cppubuntu_xenial_amd64binary
	*	Abin_uX64base_local_plannerubuntu_xenial_amd64binary
0	*	Abin_uX64bflubuntu_xenial_amd64binary
0	*	Abin_uX64bondubuntu_xenial_amd64binary
	*	Abin_uX64bond_coreubuntu_xenial_amd64binary
	*	Abin_uX64bondcppubuntu_xenial_amd64binary
0	*	Abin_uX64bondpyubuntu_xenial_amd64binary
0	*	Abin_uX64camera_calibrationubuntu_xenial_amd64binary
0	*	Abin_uX64camera_calibration_parsersubuntu_xenial_amd64binary



The New-ish Way!

Let's...

- build and package all our sources into one artifact.
- generate images for our targets containing this artifact
- build a nightly from the tip of every package's master branch.
- store all release state in one source of truth rather than multiple repositories.
- branch a release off any point in the source of truth.



Top-Level Configuration

Iocusrobotics / toydistromaster

- 🖌 📄 config
 - images.yaml
 - recipes.yaml
 - upstream.yaml
- 🕆 📄 rosdep
 - rosdep.yaml
- 🕶 📄 rosdistro
 - index.yaml
 - ros1.yaml
 - ros2.yaml
 - Optimizer States Jenkinsfile
 - README.md

```
#!/usr/bin/env groovy
versions = [
  tailor_meta: "0.1.12",
  tailor_distro: "0.1.15",
  tailor_image: "0.1.26",
]
```

```
library('tailor-meta@' + versions['tailor_meta'])_
tailorDistroPipeline(
   versions: versions
```



Tailor CI Pipeline





Mirror Configuration

La locusrobotics / toydistro & master
Y 🖿 config
🛢 images.yaml
recipes.yaml
🛢 upstream.yaml
✓ im rosdep
rosdep.yaml
✓ i rosdistro
index.yaml
🛢 ros1.yaml
eros2.yaml
🚱 Jenkinsfile

README.md

mirrors:

ros:

url: http://repositories.ros.org/ubuntu/main/

distributions:

- {{ distribution }}

components:

- main

keys:

- C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

filters:

- Name (% python*-catkin-pkg)
- Name (% python*-catkin-tools)
- Name (% python*-rosdep)
- Name (% python*-vcstool)
- Name (% python-pyside*) | Name (% libpyside*) | Name (% pyside*)
- Name (% libshiboken*) | Name (% shiboken*)



Sources Configuration

📮 locusrobotics / toydistro	
🖗 master	
Y 💼 config	
🛢 images.yaml	
recipes.yaml	
🛢 upstream.yaml	
✓ in rosdep	
rosdep.yaml	
✓ iii rosdistro	
🛢 index.yaml	
🛢 ros1.yaml	
🛢 ros2.yaml	
🚱 Jenkinsfile	
E README.md	

_	_	_
_	_	_

repositories:

secret_sauce:

source:

type: git

url: https://github.com/locusrobotics/secret_sauce.git

version: devel

test_commits: true



Sources Configuration

Iocusrobotics / toydistro	
₿∕ master	repositories:
	secret_sauce:
✓ in config	release:
images.yaml	tags:
🛢 recipes.yaml	<pre>release: '{{ version }}'</pre>
gupstream.vaml	url: https://github.com/locusrobotics/secret_sauce.git
 ✓ in rosdep 	version: 19.10.0
rosdep.yaml	type: git
✓ iii rosdistro	url: https://github.com/locusrobotics/secret_sauce.git
🛢 index.yaml	version: release/19.10
🛢 ros1.yaml	test_commits: true
🛢 ros2.yaml	
🚱 Jenkinsfile	
E README.md	



Sources Configuration

Iocusrobotics / toydistro	
∲ master	repositories:
	fastrtps:
✓ iii config	release:
images.yaml	tags:
🛢 recipes.yaml	<pre>release: release/{{ upstream }}/{{ package }}/{{ version }}</pre>
gupstream.yaml	url: https://github.com/ros2-gbp/fastrtps-release.git
rosdep	version: 1.7.2-0
	source:
	type: git
✓ I rosdistro	url: https://github.com/eProsima/Fast-RTPS.git
🛢 index.yaml	version: master
🧧 ros1.yaml	
🛢 ros2.yaml	
🚱 Jenkinsfile	
E README.md	



Packaging Configuration

Iocusrobotics / toydistro master	flavo
✓ ■ config	wa.
images.yaml	C
🛢 recipes.yaml	
🛢 upstream.yaml	
✓ in rosdep	
🛢 rosdep.yaml	
✓ im rosdistro	
🛢 index.yaml	os:
🛢 ros1.yaml	ubı
🛢 ros2.yaml	
🚱 Jenkinsfile	commo
E README.md	or
	CXX

ours:

lle:

description: mobile trash compactor

distributions:

ros1:

root packages:

- diff drive controller

ros2:

root packages:

- heart of gold

untu:

- xenial

- bionic

ion:

ganization: locusrobotics

x standard: 14

cxx flags:

- DNDEBUG



Packaging Pipeline





tailor-image



🇬 setup.py

- Python code to generate image artifacts
- Reads configuration from a yaml file
- Currently supports:
 - docker images for test pipelines and developer images
 - bare metal images for robots
- Uses packer to build images
 - Bare metal image build uses packer with the qemu-chroot builder to speed up the image creation process



Image Configuration

Incusropotics / toydistro		
₽ master	images:	
	dev:	
Y 🖿 config	build_type: docker	
🛢 images.yaml	distro: rosl	
recipes.yaml	package: robot_deploy	
Supstream.vaml	provision_file: developer.yaml	
rosden	description: Image for off-board testing and development.	
	bot:	
rosdep.yaml	build_type: bare_metal	
✓ iii rosdistro	distro: rosl	
🛢 index.yaml	package: robot_deploy	
🧧 ros1.yaml	provision_file: bare_metal.yaml	
s ros2.vaml	<pre>base_image: \$distribution-server-cloudimg-amd64-disk1.img</pre>	
Jenkinsfile	description: Image for production robots.	
E README.md		



Image Pipeline





How to consume images

- flash image directly to a hard drive
 - Useful for production
- OTA (Over-The-Air) updates
 - Used by Android, embedded systems, etc
 - A/B partition scheme for the root filesystem
 - Persistent data partition overlayed on top of the current active partition
 - Allows to rollback to a previous working version
 - Allows to run the update as a background process





tailor-meta

- In charge of creating/setting pipelines
- Setups build PRs tests for managed repos (optional)
- Handles updating github settings (optional)
 - Disables wiki
 - Disables projects
 - Only allow squash merge
 - Protect branches and set minimum number of reviewers



PR Builds

Iccusrobotics / test_meta % master

> 📄 test

.gitignore
 CHANGELOG.rst

🙏 CMakeLists.txt

Jenkinsfile

README.md

<>> package.xml

```
#!/usr/bin/env groovy
@Library('tailor-meta@0.1.12')
tailorTestPipeline(
 // Name of job that generated this test definition.
 rosdistro job: '/ci/toydistro/master',
 // Distribution name
 rosdistro name : 'ros1',
 // Release track to test branch against.
 release track: '19.10',
 // Release label to pull test images from.
 release label: '19.10-rc',
 // OS distributions to test.
 distributions: ['xenial', 'bionic'],
 // Branch of tailor meta to build against
 tailor meta branch : '0.1.12',
 // Master or release branch associated with this track
 source branch : 'master',
 // Docker registry where test image is stored
 docker registry: 'https://us-east-1.amazonaws.com/locus-toydistro'
```



In Closing

- The entire CI pipeline is available for anyone to use:
 - <u>https://github.com/locusrobotics/toydistro</u> (sample distribution)
 - <u>https://github.com/locusrobotics/tailor-distro</u>
 - https://github.com/locusrobotics/tailor-image
 - https://github.com/locusrobotics/tailor-meta
- Built around GitHub API, AWS Services (S3, ECS, EC2)
 - Happy to take patches to support other VCS systems or cloud providers.
- Send us feedback!
 - Better documentation in progress will publish on Discourse.







Management tools

Checkout a release branch for the toydistro repository

cd ~/toydistro

git checkout -b release/19.10

Gather all unpinned packages
packages=\$(tailor manage query --distro ros1 --unpinned)

Run catkin_generate_changelog and catkin_prepare_release on all unpinned repos tailor manage release --distro ros1 --release 19.10 \$packages

```
# Push release, this builds toydistro-walle-19.10-rc
git push -u origin release/19.10
```

```
# Tag release, this builds toydistro-walle-19.10.0
git tag -a 19.10.0 -m 19.10.0
git push --tags
```

