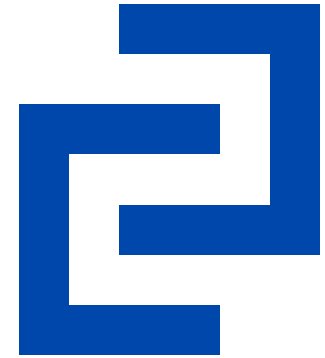


Building System Packages with Colcon

...in your own compact
buildfarm!



Scott Logan

**Software Engineer,
Intrinsic**

logans@cottsai.net



@cottsay

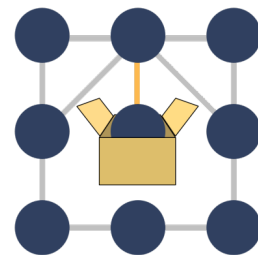


ROS™

Background Concepts



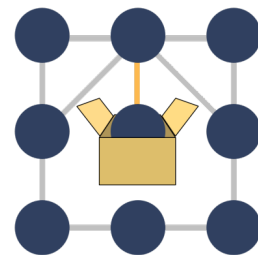
- System packages are software bundles in a format which is native to a particular platform or OS
- The ROS buildfarm produces system packages for packages released into ROS distributions
- ROS packages are typically built from source in a development context using a tool called colcon



ROS Buildfarm Review

- Produces ***.deb** and ***.rpm** binary packages for Ubuntu/Debian/RHEL
- Build logic is centralized in the **ros-infrastructure/ros_buildfarm** repository
- Frontend and orchestration are supplied by **Jenkins**
- Individual jobs can be invoked by a **script** outside of Jenkins

Motivation



Invoking `ros_buildfarm` jobs as a script:



- No dependency awareness
- No repository management

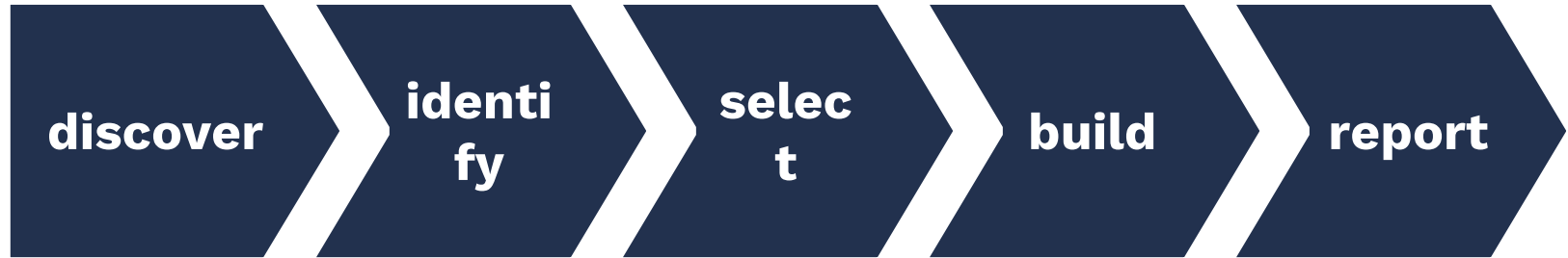
Standing up a full buildfarm:



- Complex and requires deployment to multiple hosts
- Heavy maintenance burden

Why colcon?

1. It fits the model



Why colcon?

2. Familiar to developers

```
$ colcon build --packages-select rcutils  
Starting >>> rcutils  
Finished <<< rcutils [1.90s]
```

Summary: 1 package finished [3.52s]



Why colcon?

3. Minimal boilerplate

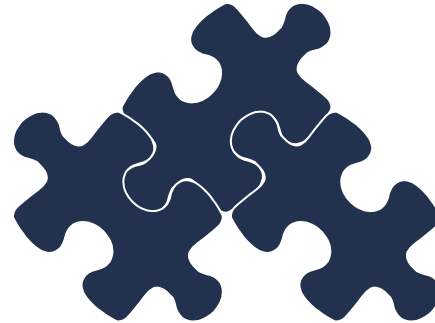
```
$ find * -type f | xargs wc -l
 4 colcon_ros_buildfarm/__init__.py
33 colcon_ros_buildfarm/command.py
54 setup.cfg
 6 setup.py
97 total
```



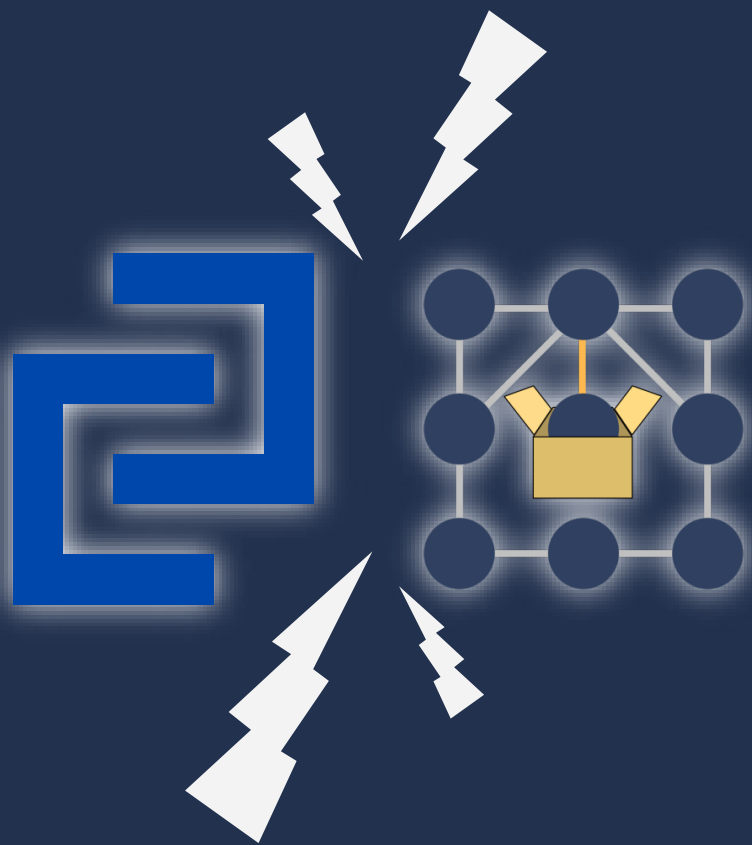
Why colcon?

4. Robust extensibility

- Event handlers
- Package selection
- Executors
- Package formats
- ...etc









colcon-ros-buildfarm



Features At A Glance

A New Tool

Not a colcon verb, but part of the colcon extension ecosystem

Identical Packages

Using `ros_buildfarm` under the hood, packages are built in containers the same way `build.ros2.org` does

Same Execution Model

Combines colcon's execution model with the script invocation mechanism in

Easy To Install

Installed the same way as any other colcon extension

Produces Repositories

Curates repository metadata locally for both `.deb` and `.rpm` packages

Easy To Run

A single command can reproduce any release job from `build.ros2.org`



Key Differences



No Source Directory

Package discovery is based on presence in a rosdistro index, not a local directory

Cross-Platform

Using containers makes it possible to build packages for an OS different from the host

Requires Network

The build process is network-intensive and downloads package sources, dependencies, and metadata for each build

Demonstration



Let's build some packages!

1.

Scenario:

Re-building official packages

Requires NO configuration - just select the packages you want to build


```
rosuser@localhost:~/bf_ws$
```



2.

Scenario:

Building a custom rosdistro

Follow the same setup as `ros_buildfarm` and configure `colcon-ros-buildfarm` to use a custom URL

```
rosuser@localhost:~/bf_ws$
```



```
rosuser@localhost:~/bf_ws$
```





Additional Extensions



colcon-run-when

Listen for "external" events and trigger appropriate colcon commands



colcon-notify-email

Send email messages when builds fail or succeed after failing previously

Replacing Jenkins? ...or not



- No web frontend/dashboard
- No workload distribution
- Missing job types (ci/dev/doc/pr)

Replacing Jenkins? ...or ~~not~~ *maybe!*



- No web front end dashboard
- No web lead in
- Missing (e.g. doc/pr)



Challenges

- The colcon API was not ready for this
- SELinux and other security measures
- Disk usage is far too high
- Memory demands





Summary

colcon-ros-buildfarm is a new tool which:

- Produces repositories of ready-to-use system package
- May replace Jenkins deployments for some use cases
- Is built from the relevant parts of colcon and ros_buildfarm



Try it out!

```
$ vcs import src --input https://github.com/colcon/colcon-ros-  
buildfarm/raw/devel/crb.repos
```

```
$ colcon build
```

```
$ . install/setup.sh
```

```
$ ros_buildfarm release --packages-select rcutils
```

THANKS!

