

**datto**

# **Flexible & Fast Software Delivery with OBS**

**Neal Gompa**

**DevOps Engineer**



# Who am I?

- Professional technologist
- Linux user for nearly fifteen years
- Contributor and developer in Fedora, Mageia, openSUSE, and OpenMandriva Linux distributions
- Contributor to RPM, DNF, and various related projects
- DevOps Engineer at Datto, Inc.
- Likes: orchestral/symphonic music, anime and manga, global history, linguistics
- Dislikes: overly spicy food

## Contact Points:

- Twitter: [@Det\\_Conan\\_Kudo](https://twitter.com/Det_Conan_Kudo)
- Email: [ngompa@datto.com](mailto:ngompa@datto.com)



# In the beginning times...

# Building and distributing software sucked...

Datto, like most modern software development companies, uses a Linux-based system for its servers and customer equipment.

However, the general focus is often just about *making* the software, rather than *shipping* it.

When it came time to actually ship the software, Datto used to wrap everything in ZIP archives, push them, and then unpack them remotely onto the filesystem.

This made things difficult as more complex “setup” tasks were required, which led to very complex mechanisms that we collectively call “checkin”.

# The transition to Linux packaging...

This really stopped scaling after the second time we had to upgrade all the servers and devices to newer base Linux distribution software.

As a company, we made the transition to using proper-ish Linux packaging for our software, to handle this more smoothly.

Our system for handling it was a custom monstrosity! It was also very limited in what it could do.

But it worked, somewhat, until...



# Introducing the Datto Linux Agent!

Austin McChord, DattoCon 15 (2015)

# Our custom system hit the limits...

There were a number of issues with the custom system that we mostly hand-waved away until now:

- Dependent packages can't build against each other
- Restricted to only one distribution family
- Setup was very fragile

At launch, the Datto Linux Agent (DLA) supported Fedora, Red Hat Enterprise Linux/CentOS, Debian, and Ubuntu. There were plans to add SUSE Linux Enterprise and openSUSE later.

Because of the wider range of distributions and dependency requirements, I was forced to build every DLA release by hand, spending a week to actually build and validate them. At the time, that was 18 builds (9 for kernel part, 9 for userspace part)!



# This was killing me...



# Fixing the situation...

The situation was not scalable. So, I started looking into what to do to make this less burdensome.

After looking into the problem on the Internet and examining what others had done, I came up with three major alternatives:

- Build a brand new custom system, possibly Buildbot-based with some extensions for handling dependency resolution using the Hawkey module
- Adapt Fedora's Mock and Koji to extend it to support Debian/Ubuntu builds
- Use SUSE's Open Build Service

After deeper examination, I decided to deploy an OBS instance and go from there.



# So why the Open Build Service?

# Open Build Service

The Open Build Service (OBS) is a software solution created by SUSE to build and manage the openSUSE and SUSE Linux Enterprise distributions.

However, it was designed from the beginning to support a wide variety of Linux based platforms. Notably, it can build packages, repositories, and images for Red Hat/Fedora, SUSE, and Debian/Ubuntu systems.

SUSE offers a hosted version as the openSUSE Build Service, and the appliance image is freely available for you to set up your own.

# Software Delivery with OBS

The Open Build Service has several capabilities to be integrated with VCSes (such as Mercurial, Git, Subversion, etc.) that allow it to either pull code or receive code regularly to build.

Once it has the necessary inputs, it can build the software for all enabled targets. After the build, it can fire off events to trigger automated tests and/or review processes.

Failures can also trigger events and notifications, depending on your configuration.

# Unique Features of OBS

- Source input flexibility through “source services” that allow scripted retrieval and processing of sources
- Easy scaling of resources through OBS workers that detect the master and connect
- Multiple build engines for building software in a particular preferred way
- Speed up build environment setup with pre-install images
- Automatic reverse dependency rebuilding on package updates to ensure dependencies are linked correctly
- Fully customizable build environments
  - Even the type of build environment
    - VMs through KVM, Xen, or zVM
    - Emulated foreign architecture VMs
    - Containers using LXC
    - Plain chroots



# So how does Datto use OBS?

# Datto's OBS

- Deployed using the official appliance provided on the website in an all-in-one configuration (master and worker on the same machine)
- Appliance runs in a VM without nested virtualization, so container build environments are used
- We primarily target Ubuntu for most of our software
  - Though we build nearly all software with RPM spec files using OBS's spec builder engine and debbuild!
    - The packages are native, proper Debian packages!

# Datto's OBS: Statistics

- We currently build nearly 500 packages across nearly 350 projects, with over 1,100 repositories
  - The last mass rebuild took ~8 hours to build across 6 builders on the single machine
- Most projects target at least two distribution families and produce native packages for both using the same RPM spec file
- The Datto Linux Agent has been built for over 50 distribution targets across the Red Hat/Fedora, SUSE, and Debian/Ubuntu distribution families.





# Demonstration

# Who else uses OBS?

Notable other users:

- openSUSE: <https://build.opensuse.org/>
- Packman: <https://pmb.s.links2linux.org/>
- Mer: <https://build.merproject.org/>
- Tizen: <https://build.tizen.org/>
- Linaro: <https://obs.linaro.org/>
- Dell (non-public):  
<http://linux.dell.com/repo/community/>
- Collabora (private-ish): <https://build.collabora.co.uk/>
- VideoLAN (non-public):  
<https://www.videolan.org/vlc/download-suse.html>



# OBS is awesome!

**If you're interested, we're hiring DevOps Engineers!**

<https://www.datto.com/careers>

# References

- Open Build Service website: <http://openbuildservice.org/>
- Reference public instance by openSUSE: <https://build.opensuse.org/>
- Open Build Service documentation: <http://openbuildservice.org/help/>
- The debbuild GitHub project: <https://github.com/debbuild/debbuild>



# The End

**Any Questions?**

**Also, we're hiring!!!**

**<https://www.datto.com/careers>**