



# Bleep bloop! A robot workshop.

Aaron Conole <[aconole@redhat.com](mailto:aconole@redhat.com)>

December 6, 2018

My presentation is a lot like yours, only more interesting 'cause it involves robots

### The talk

- What is the 0-day robot?
- How can I avoid being taunted by a machine?
- I want to mess with the robot, how do I do that?
- Community feedback

### Don't forget

Tweet about things here with tag #OVSCon

## Hey ovs-dev, wanna test all patches?

### 0-day means testing before day 1

- Yes... I know... 0-day sometimes means something more ominous
- In this case, it's before day 1 of the patch landing
- Almost 1000 "series" processed since June 2018

### Simple machines that talk to other machines

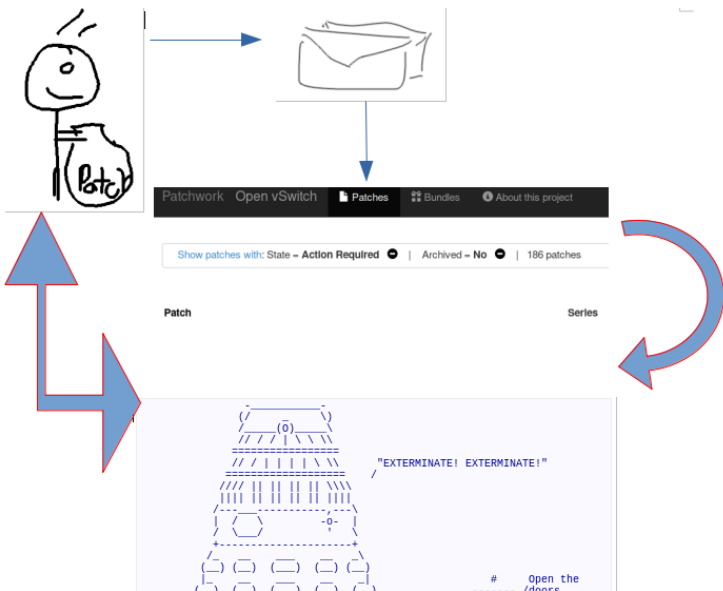
- Idea is straight-forward, we have some mechanical tasks
- No real thought required - just do some basic stuff
- Email everyone on failure
- Informative, not blocking

Oh, I get it— make the robot do all the work.

### All these patches, like tears in a rain

- Developers come up with a bunch of patches
- They send them via email to the mailing list
- Patchwork gets those emails and builds up series information
- The robot monitors patchwork and starts processing

# 95473936754455451545245455478764



# How can I be so bad at everything I try and still be so great?

## Directive 4

- 1 Check out master (or, for certain notation, the appropriate branch)
- 2 Apply the next patch in the series
- 3 Run checkpatch
- 4 Push to `https://github.com/ovsrobot/ovs`
- 5 `'./configure && make'`
- 6 More patches? Goto 2
- 7 `'make distcheck'` - we'll get to it

# Now tragedy! Ha ha ha, that's funny.

## If any of those fail... bleep bloop!

- Previous message: [\[ovs-dev\] \[PATCH\] MAINTAINERS: Move Eiban to emeritus status.](#)
- Next message: [\[ovs-dev\] \[PATCH\] MAINTAINERS: Move Eiban to emeritus status.](#)
- Messages sorted by: [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

Bleep bloop. Greetings Ben Pfaff, I am a robot and I have tried out your patch.  
Thanks for your contribution.

I encountered some error that I wasn't expecting. See the details below.

```
build:
reading sources... [ 95%] tutorials/faucet
reading sources... [ 96%] tutorials/zoox
reading sources... [ 97%] tutorials/ovs-openstack
reading sources... [ 98%] tutorials/ovs-sdnbox
reading sources... [ 99%] tutorials/ovs-advanced
reading sources... [100%] tutorials/ovs-contrack
```

Warning, treated as error:

MAINTAINERS.rst:81: E96R: Error parsing content block for the "list-table" directive: uniform two-level bullet list expected, but row 2 does not contain the same number of items as row 1 (3 vs 2).

```
.. list-table:: OVS Emeritus Maintainers
   :header-rows: 1
```

```
   - None
   - Email
   - Ethan J. Jackson
   - ejl@ecs.berkeley.edu
   -
```

```
make[2]: *** [docs-check] Error 1
make[2]: Leaving directory '/var/lib/jenkins/jobs/upstream_build_from_pu/workspace'
make[1]: *** [all-recurse] Error 1
make[1]: Leaving directory '/var/lib/jenkins/jobs/upstream_build_from_pu/workspace'
make: *** [all] Error 2
```

Please check this out. If you feel there has been an error, please email [axw@redhat.com](mailto:axw@redhat.com)

Thanks,  
0-day Robot

- Previous message: [\[ovs-dev\] \[PATCH\] MAINTAINERS: Move Eiban to emeritus status.](#)

## Shut up and get to the point!

### The logic is undeniable, prevention of bleep bleeps is easy:

- 1 Run checkpatch (Install it as git hook! Then it just happens!)
- 2 Run a `make distcheck` for your series
- 3 Make sure that each patch builds individually (helps with bisecting)
- 4 Make sure each patch will apply to the branch cleanly
- 5 If you use github, push to a branch, and monitor the travis output.



## Now, no one can say I don't own a checkpatch hook

### Attractive! Nice software

```
12:28:40 aconole ~$ cat git/ovs/.git/hooks/pre-commit
#!/bin/sh
if git rev-parse --verify HEAD 2>/dev/null
then
    git diff-index -p --cached HEAD
else
    :
fi | utilities/checkpatch.py -S
```

## Here I am with a brain the size of a planet

### Check your travis branch status

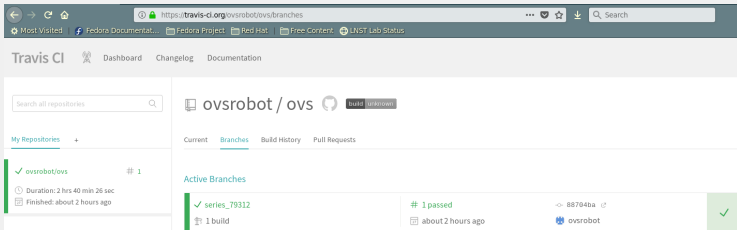
- go to `https://patchwork.ozlabs.org/project/openvswitch/list/`
- Look up the series ID for your patches
- Find them on travis  
`https://travis-ci.org/ovsrobot/ovs/branches`

### Better monitoring / reporting could be a welcome improvement

- Right now, the branch lookup is unweildy
- Probably need some effort to do cleanups in this area
- Remember, the will accomplishes everything

# Test my shiny, metal patch

I could see your patch move



The screenshot shows the Travis CI web interface for the repository `ovsrobot / ovs`. The browser address bar shows `https://travis-ci.org/ovsrobot/ovs/branches`. The interface includes a search bar for repositories, a list of repositories, and a section for active branches.

**My Repositories**

- ✓ ovsrobot/ovs # 1
  - ⌚ Duration: 2 hrs 40 min 26 sec
  - 📅 Finished: about 2 hours ago

**Active Branches**

Branch	Status	Last Commit	Author
✓ series_79312	# 1 passed	88704ba	ovsrobot

## Are you familiar with the old robot saying, DOES NOT COMPUTE?

### Why doesn't make distcheck get reported?

- Some of the tests in the make-check testsuite are unreliable
- Specifically, the most recently failing one I've seen is "ovn – ensure one gw controller restart in HA doesn't bounce the master"
- Could be more... could be an artifact of the system

## Yeah, well I'm gonna go build my own CI, with blackjack

Okay, maybe you want a robot of your very own

Who wouldn't?

Or you want to be as lazy as you can be

- By letting a robot on your laptop test your patches
- Because again, who wouldn't?

Either way, the door opened and you got in

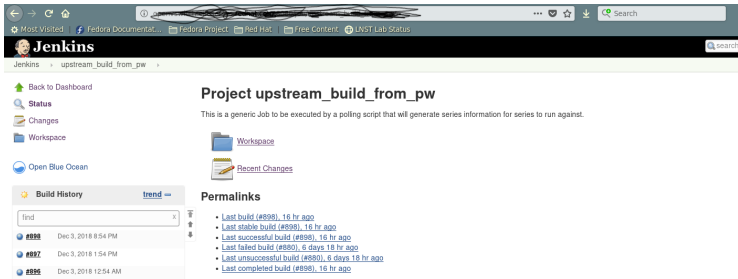
- With <https://github.com/orgcandman/pw-ci>
- You too can launch a robot

Give you a fresh start. All we ask is your cooperation...

### Let's call him clamps

- 1 Clone the repo (<https://github.com/orgcandman/pw-ci>)
- 2 `cd pw-ci/3rd-party/openvswitch/`
- 3 Run `./build.sh`
- 4 Set up `jenkinsrc` and `pwmonrc` files
- 5 Import the config (this is the main job)
- 6 Optionally, set up another jenkins job to run `pwmon` and `cisubmit`

# We launch at six bells



The screenshot shows the Jenkins web interface for a job named 'Project upstream\_build\_from\_pw'. The browser's address bar shows the Jenkins URL. The page title is 'Project upstream\_build\_from\_pw'. Below the title, there is a description: 'This is a generic Job to be executed by a polling script that will generate series information for series to run against.' There are links for 'Workspace' and 'Recent Changes'. A 'Build History' section is visible, showing a table of recent builds. The table has columns for build number, status, and time. The most recent build is #898, which is successful and completed 16 hours ago. Below the build history, there is a 'Permalinks' section with a list of links to specific build details.

**Project upstream\_build\_from\_pw**

This is a generic Job to be executed by a polling script that will generate series information for series to run against.

[Workspace](#)

[Recent Changes](#)

**Build History** [trend](#)

Build Number	Status	Time
#898	Success	Dec 3, 2018 8:54 PM
#897	Success	Dec 3, 2018 1:54 PM
#896	Success	Dec 3, 2018 12:54 AM

**Permalinks**

- [Last build \(#898\), 16 hr ago](#)
- [Last stable build \(#898\), 16 hr ago](#)
- [Last successful build \(#898\), 16 hr ago](#)
- [Last failed build \(#880\), 6 days 18 hr ago](#)
- [Last unsuccessful build \(#880\), 6 days 18 hr ago](#)
- [Last completed build \(#898\), 16 hr ago](#)

## Now, all I have to do is spell check it

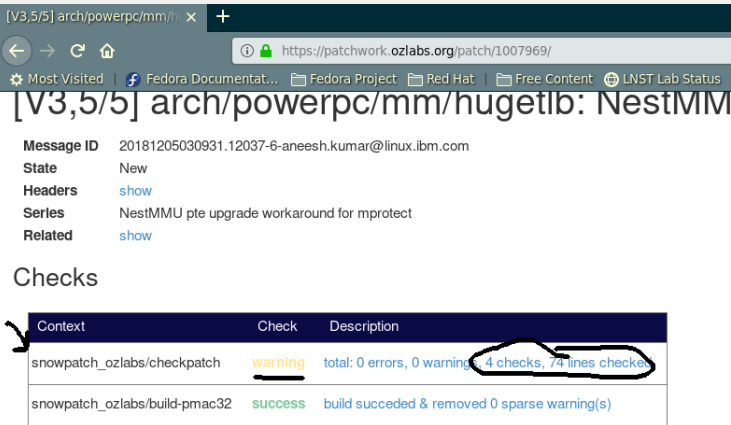
### patchwork reporting

- Patchwork supports 'check' reporting
- These are basically, test name, test status, link to extra data
- Could be useful for having an all-in-one spot to push CI status.
- 'Easy' to report (can be done with curl)



# I better use my special robot vision to see what's inside

## A view of checks



[V3,5/5] arch/powerpc/mm/hugetlb: NestMMU

Message ID 20181205030931.12037-6-aneesh.kumar@linux.ibm.com  
State New  
Headers [show](#)  
Series NestMMU pte upgrade workaround for mprotect  
Related [show](#)

### Checks

Context	Check	Description
snowpatch_ozlabs/checkpatch	<b>warning</b>	total: 0 errors, 0 warnings, <b>4 checks, 74 lines checked</b>
snowpatch_ozlabs/build-pmac32	<b>success</b>	build succeeded & removed 0 sparse warning(s)

## That was the other guy. My name's boiler

### Partnering with other groups

- UNH interoperability lab currently runs tests for DPDK
- Working with Jeremy Plsek to incorporate some NFV and ovs-dpdk testing
- UNH IOL currently does performance testing and is building a public dashboard

# I need your nics, your machines, and your patches

## Send hardware!

- Mellanox is the first up to ship some HW that we'll incorporate
- Other famous hardware makers who will remain unnamed have said ...
- Anyone else? Takers?

## Send patches

- Want to enable lots of the check-suites: check-dpdk, check-userspace, check-kmod
- Most need some kind of privileged resource.
- Maybe you have a fix?

# Questions?