


Using Nix in production

@ NixOS London meetup 2016

Domen Kožar

Question: NixOS for an OpenStack CI? #10268

 Closed

lukego opened this issue on Oct 7, 2015 · 3 comments



lukego commented on Oct 7, 2015



Howdy!

We are considering using NixOS as the basis for a Continuous Integration system for an open source project called [Snabb NFV](#). This would run multiple tests and the most complex one is to setup and test a basic single-node OpenStack environment including Nova (w/ Libvirt+QEMU) and Neutron (w/ MySQL database). We need to be able to easily test with different versions/patches of each software component including bootstrapping VM images with specific kernels and awkward-to-compile software (e.g. Intel DPDK that depends on host kernel and CPU at build time).

We are new to NixOS and we would probably need some help to get us moving in the right direction and to have this up and running in a reasonable timeframe (1-2 months).

The main alternative that I see is to run [devstack](#) inside a Docker container. On the positive side this would reuse a lot of work that other people in the OpenStack community are doing. On the negative side this is really messy and easily veers off into becoming unreproducible over time.

I would really appreciate advice! Large or small effort to run Nova+Neutron as above on NixOS? Could we find somebody available now to help us implement this in such a way that we can understand and maintain it? Am I overlooking some important things?

Thanks!

Snabb

How does it work?

Snabb is written using these main techniques:

- Lua, a high-level programming language that is easy to learn.
- LuaJIT, a just-in-time compiler that is competitive with C.
- Ethernet I/O with no kernel overhead ("kernel bypass" mode).

Snabb compiles into a stand-alone executable called `snabb`. This single binary includes multiple applications and runs on any modern [Linux/x86-64](#) distribution. (You could think of it as a [busybox](#) for networking.)

PUBLIC REPOSITORY

eugeneia/snabb-nfv-test

Last pushed: a year ago

Repo Info

Tags

Short Description

Snabb Switch NFV test image

Full Description

Snabb NFV Test Environment including:

QEMU 2.4.0 + c9cea8f (patch to enable vhost_user reconnect)

Guest: Ubuntu 14.04 (kernel 3.13.11-ckt26)

Guest DPDK: "legacy-snabb" (<https://github.com/eugeneia/dpdk/tree/legacy-snabb>)

Build CPU: Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz

VM build log

[Expand all](#) | [Collapse all](#)

```
- running the VM test script
— allinone: starting vm
— allinone: QEMU running (pid 23609)
— + allinone: waiting for unit 'keystone-all.service'
— + allinone: waiting for TCP port 35357
— + allinone: waiting for TCP port 5000
— + allinone: waiting for unit 'glance-api.service'
— + allinone: waiting for TCP port 9191
— + allinone: waiting for TCP port 9292
— + allinone: waiting for unit 'neutron-server.service'
— + allinone: waiting for unit 'nova-api.service'
— + allinone: waiting for TCP port 8774
— + allinone: waiting for unit 'nova-compute.service'
— + allinone: waiting for unit 'nova-conductor.service'
— + allinone: running command: /root/bootstrap.sh
— + subtest: VM with NIC
— + subtest: VM with 2xNIC (low bandwidth)
— + subtest: VM with 2xNIC (high bandwidth)
— + subtest: 2xVM on same physical port
— + subtest: 2xVM on different physical port
— + subtest: 2xVM with security group restrictions
— + subtest: 2xVM bandwidth restriction
— + subtest: VM with L2TPv3
— + subtest: Multiple VMs
— + subtest: VM with NIC, invalid zone
— + subtest: 2xVM on different physical port, restart neutron
— + subtest: 2xVM on different physical port, restart nova
— + subtest: 2xVM on different physical port, restart snabb services
— + subtest: L2TPv3 between 2 VMs
— + subtest: L2TPv3 between 2 VMs add remove tunnel
— + subtest: 2xVM with security group restrictions
— + subtest: 2xVM on same physical port IPv4
— + subtest: 2xVM with security group restrictions and stateless filtering
+ collecting coverage data
+ syncing
18 out of 18 tests succeeded
+ cleaning up
```

snabblab/snabblab-nixos

- Lab machines for Snabb development
- build/test/benchmark Snabb
- Hydra as CI

Sharing PCI resources

```
users.extraUsers.root.openssh.authorizedKeys.keys = pkgs.  
  command="nice -n20 ${pkgs.utillinux}/bin/flock -s /var/l  
  '';
```

Sudo in builds

```
let
  # use sudo without pam (easier in chroots)
  sudoChroot = sudo.overrideDerivation (super: {
    configureFlags = super.configureFlags ++ [ "--without"
    postInstall = ''
      mv $out/bin/sudo $out/bin/sudo-chroot
    '';
  });
in {
  nix.chrootDirs = [
    "/var/setuid-wrappers/sudo=/var/setuid-wrappers/sudo-
    "/var/setuid-wrappers/sudo.real=/var/setuid-wrappers/
    "${sudoChroot}"
    "/etc/sudoers"
    "/etc/passwd"
    "/sys"
  ];
  environment.systemPackages = [ (lowPrio sudoChroot) ];
  security.setuidPrograms = [ "sudo-chroot" ];
```

Benchmarks

- 10 different test cases.
- 5 versions of QEMU.
- 10 different guest VMs (Linux and DPDK).
- 16 combinations of Virtio-net options.
- 2 NUMA setups ("good" and "bad")
- 2 polling modes (engine "busy loop" and sleep/backoff)
- 2 error recovery modes (engine supervising apps vs process restart)
- 3 CPUs (Sandy Bridge, Haswell, Skylake)



```
{ numTimesRunBenchmark ? 1
, nixpkgs ? (fetchTarball https://github.com/NixOS/nixpkgs)
, snabbAsrc
, snabbBsrc ? null
, snabbCsrc ? null
, snabbDsrc ? null
, snabbEsrc ? null
, snabbFsrc ? null
, snabbAname
, snabbBname ? null
, snabbCname ? null
, snabbDname ? null
, snabbEname ? null
, snabbFname ? null
, benchmarkNames ? [ ]
, reports ? []
, kernelVersions ? ["3.18"]
, dpdkVersions ? []
, dpdkAsrc ? null
, dpdkAname ? null
, qemuVersions ? []
, qemuAsrc ? null
, qemuAname ? null
}:
```

Lab Machines

- {Grindelwald,Interlaken} at Luke's home
- Lugano-{1,4} at [Switch.ch](#)
- Snabb2 at [lgalia.es](#)
- Murren-{1,10} at [Hetzner.de](#)

mkDerivation

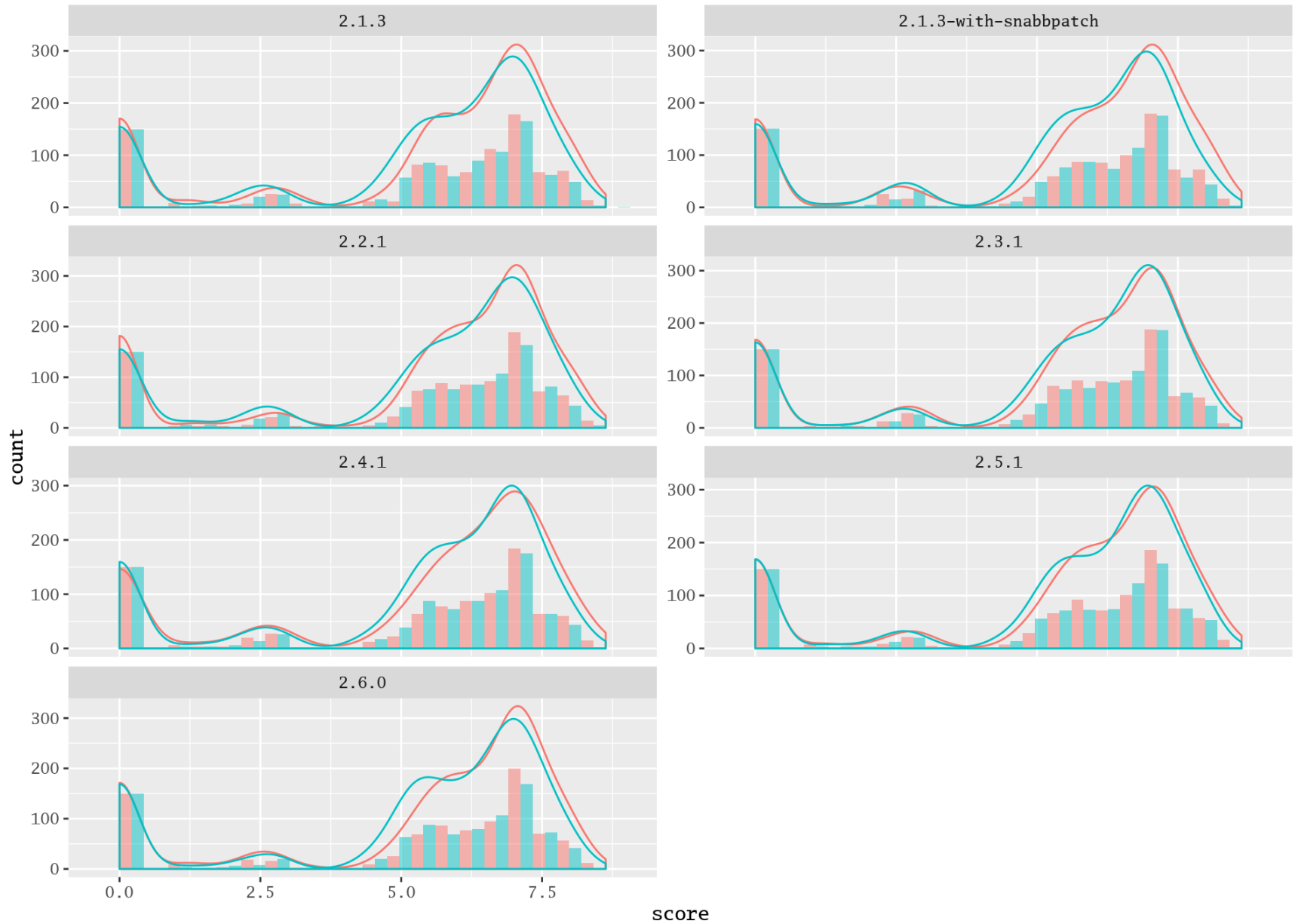
```
requiredSystemFeatures = [ "lugano" ];
```

Hydra

```
maxJobs = 1;  
supportedFeatures = [ "kvm" "nixos-test" ];  
mandatoryFeatures = [ "lugano" ];
```

12fwd by QEMU (MPPS)

snabb master next



Memories

Eats 4G of my ram and then gets killed by OOM.

```
$ nix-build jobsets/snabb-matrix.nix -A benchmarks  
nix-instantiate killed by signal 9
```

With the following patch evaluation takes 2.5G of ram.

```
-      repeatDrv = i: drv.override { numRepeat = i; };  
+      repeatDrv = i: lib.hydraJob (drv.override { numRep
```

814 seconds to evaluate

```
concatMap
  (kPackages:
    concatMap (dpdk:
      concatMap (qemu:
        concatMap (snabb:
          selectBenchmarks
            benchmarkNames
              { inherit snabb qemu dpdk times
                kPackages; }
        ) snabbs
      ) subQemus
    ) subDpdks
  ) subKernelPackages;
```

11 seconds evaluate

```
concatMap
  (kPackages:
    concatMap (dpdk:
      let
        testNixEnv = mkNixTestEnv { inherit kPackages dpdk;
      in
        concatMap (qemu:
          concatMap (snabb:
            selectBenchmarks
              benchmarkNames
                { inherit snabb qemu dpdk times
                  kPackages testNixEnv; }
          ) snabbs
        ) subQemus
      ) subDpdks
    ) subKernelPackages;
```

Determinism is merely a special case of nondeterminism, and not a terribly interesting one at that

Same source, different hash:

- `fetchurl`
- `builtins.fetchTarball`
- `builtins.fetchgit`
- `$ nix-prefetch-git+fetchgit`
- `$ nix-prefetch-git (Hydra)`

domenkozar/nixbot

```
nixbot.github_token = <insert github token>
nixbot.bot_name = nixbot
nixbot.repo = domenkozar/nixpkgs
nixbot.pr_repo = nixos/nixpkgs-pr
nixbot.github_secret = justnotsorandom
nixbot.public_url = http://45b4a7ff.ngrok.io
nixbot.hydra_jobsets_repo = nixos/hydra-jobsets
nixbot.repo_dir = /tmp/nixbot/repositories
```

Requires most recent Nix and Hydra :-)



124 IN PROGRESS

32345 IN QUEUE

19 MACHINES

EVALUATIONS

BUILDS

STEPS



Jobset release-16.03

EVALUATIONS

JOBS

CHANNELS

Lastest check

2016-08-06 12:38:01

Lastest evaluation

[2016-08-06 17:45:55](#)

Lastest finished evaluation

[2016-08-06 17:45:55](#)

#	Input chages	Job status	Time
123	snabbBsrc -- e1fdc74	145 62 23	2016-08-05 13:43:40
123	snabbBsrc -- e1fdc74	145 62 23	2016-08-05 13:43:40
123	snabbBsrc -- e1fdc74	145 62 23	2016-08-05 13:43:40

Thank you!

domen@dev.si