

# Using Nix in production for the last two years

@ NixCon 2017

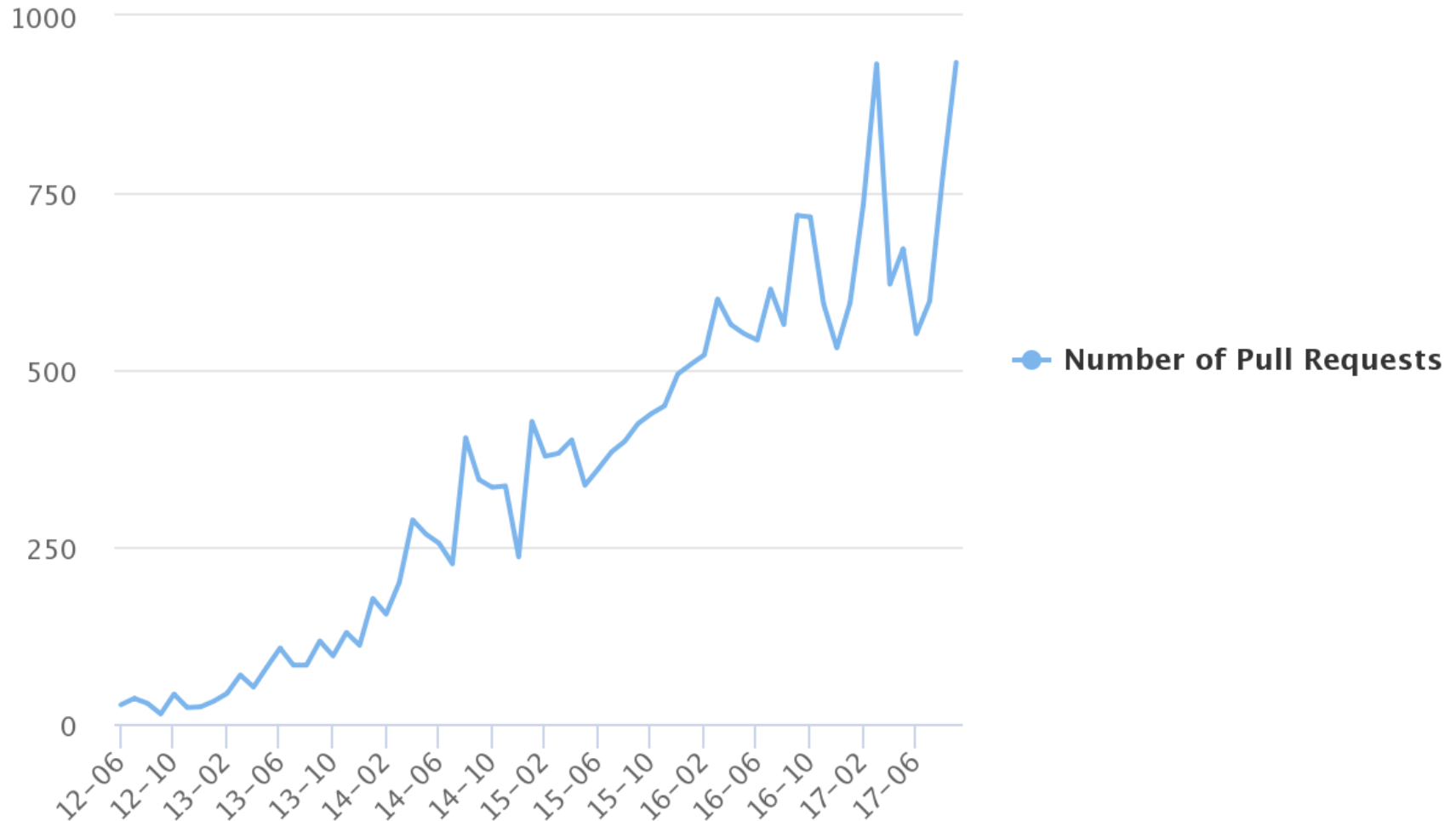
Domen Kožar



ENLAMBDA

# <http://github.com/NixOS/nixpkgs> Pull Requests per month

Source: github.com



Upstreamable nixpkgs forks?

Why Github can't host the Linux Kernel Community

# Fast track documentation

Give a Man a Fish, and You Feed Him for a Day. Teach a Man To Fish, and You Feed Him for a Lifetime

<https://medium.com/@MrJamesFisher/nix-by-example-a0063a1a4c55>

# Gradual process of Nixifying

Impure -> Nix

**Nix caveats**

```
builtins.fetchTarball
```

```
"/nix/store/g43i2h7hh43zr146g9wbq2cs5wj9dac-master"
```

```
builtins.fetchgit
```

```
"/nix/store/fws69vhxrmxjgp83dh9xc86z1c1p8k2j-git-export"
```

```
Nixpkgs nix-prefetch-git
```

```
/nix/store/dxawz07sxcd6m25r982qwljxfz128bak-snabb
```

```
Hydra nix-prefetch-git
```

```
/nix/store/nvn3q89lawxa1ar7x9ywhglnnki3y0dd-git-export
```

```
./.
```

```
/nix/store/3q6qjy0qxswiz8vq7vyi7arrb4shpi55-snabbswitch
```



**Pinning nixpkgs commit**

```
fetchTarball https://github.com/NixOS/nixpkgs/archive/37e7e86  
NIX_PATH=https://github.com/NixOS/nixpkgs/archive/37e7e86  
#!/usr/bin/env nix-shell
```

Leaking `<nixpkgs>` :(

```
with builtins;
let
  script = toFile "builder" ''
    export PATH=$coreutils
    $coreutils/mkdir $out
    cd $out
    $gzip -d < $src | $tar -x --strip-components=1
  '';
  src = let
    spec = fromJSON (readFile ./nixpkgs-src.json);
  in import <nix/fetchurl.nix> {
    url = "https://github.com/${spec.owner}/${spec.repo}";
    inherit (spec) sha256;
  };
  nixcfg = import <nix/config.nix>;
in builtins.derivation {
  system = builtins.currentSystem;
  name = "${src.name}-unpacked";
  builder = nixcfg.shell;
  inherit src;
  args = [ script ];
  coreutils = builtins.storePath nixcfg.coreutils;
  tar = builtins.storePath nixcfg.tar;
  gzip = builtins.storePath nixcfg.gzip;
}
```

- Empty `NIX_PATH`
- Scripts:  
`export NIX_PATH="nixpkgs=${nix-build nixpkgs.nix}"`
- distro installed Nix: `nix-env -iA nix`
- Nix 1.12 has `fetchTarball` that accepts an optional hash

```
nix-repl> :t a: a  
a function
```

```
nix-repl> :t a:a  
a string
```

```
nix-repl> :t http://example.com  
a string
```

```
nix-repl> :t a: a  
a function
```

```
nix-repl> :t a:a  
a string
```

```
nix-repl> :t http://example.com  
a string
```

**error: value is a string while a set was expected**

```
let  
  yaml = ''  
  files:  
    '' + ''  
    - a  
    - b  
    - c  
    '';  
in yaml
```

```
$ nix-instantiate foo.nix --eval  
"files:\n- a\n- b\n- c\n"
```

<https://github.com/NixOS/nix/pull/580>



```
$ nix-instantiate -E --eval './RELEASE'  
/home/ielectric/dev/iohk-ops/RELEASE
```

```
$ nix-instantiate -E --eval '"${./RELEASE}"'  
"/nix/store/v449pi6vy32fq7mxq3kfkjga1d8894w1-RELEASE"
```

```
$ nix-instantiate -E --eval '"${toString ./RELEASE}"'  
"/home/ielectric/dev/iohk-ops/RELEASE"
```

```
nix-repl> :t ./RELEASE  
a path
```

```
nix-repl> :t (toString ./RELEASE)  
a string
```

```
nix-repl> :t (builtins.toPath (toString ./RELEASE))  
a string
```

```
$ nix-instantiate -E --eval '"asd" + ../.'  
"asd/nix/store/lkygp11cxxfppry4d6lk8krpishixvfk-iohk-nixc  
  
$ nix-instantiate -E --eval ' ../. + "/asd" '  
/home/ielectric/dev/iohk-ops/asd
```

**quazi toPath**

```
> builtins.unsafeDiscardStringContext (toString ./RELEASE
"/home/ielectric/dev/iohk-ops/RELEASE"

> toString (builtins.unsafeDiscardStringContext ./RELEASE
"/nix/store/v449pi6vy32fq7mxq3kfkjga1d8894w1-RELEASE"

> :t toString (builtins.unsafeDiscardStringContext ./RELEASE
a string

> :t (builtins.unsafeDiscardStringContext (toString ./RELEASE
a string
```

**"Nix the good parts"**

**<https://github.com/Gabriel439/nixfmt>**

**ambiguous parser :(**

```
$ dd if=/dev/urandom bs=3 count=1 > foo

$ nix-instantiate -E --eval "builtins.readFile ./foo"
"5"

$ dd if=/dev/urandom bs=3 count=1 > foo

$ nix-instantiate -E --eval "builtins.readFile ./foo"
error: the contents of the file '/home/ielectric/dev/foo'
```

<https://github.com/NixOS/nix/issues/1307>

```
$ dd if=/dev/urandom bs=3 count=1 > foo

$ nix-instantiate -E --eval "builtins.readFile ./foo"
"5"

$ dd if=/dev/urandom bs=3 count=1 > foo

$ nix-instantiate -E --eval "builtins.readFile ./foo"
error: the contents of the file '/home/ielectric/dev/foo'
```

<https://github.com/NixOS/nix/issues/1307>

**NixOps: deployment.keys.keyFile**

# NixOS caveats



```
$ \time nix-instantiate --eval -E " \  
  (import <nixpkgs/nixos/lib/eval-config.nix> { \  
    modules = []; \  
  }).config.system.build.toplevel"
```

```
$ \time nix-instantiate --eval -E " \  
  (import <nixpkgs/nixos/lib/eval-config.nix> { \  
    modules = []; \  
  }).config.system.build.toplevel"  
0.51user 0.05system 0:00.57elapsed 100%CPU (0avgtext+0avg  
0inputs+0outputs (0major+45462minor)pagefaults 0swaps
```

```
mergeAttrsMap (kPackages:
  mergeAttrsMap (dpdk:
    mergeAttrsMap (qemu:
      mergeAttrsMap (snabb:
        selectBenchmarks benchmarkNames {
          inherit snabb qemu dpdk kPackages;
        }
      ) snabbs
    ) subQemus
  ) ((selectDpdk dpdkVersions kPackages))
) subKernelPackages;
```

```
mergeAttrsMap (kPackages:
  mergeAttrsMap (dpdk:
    let
      testNixEnv = mkTestEnv { inherit kPackages dpdk; };
    in
    mergeAttrsMap (qemu:
      mergeAttrsMap (snabb:
        selectBenchmarks benchmarkNames {
          inherit snabb qemu dpdk kPackages testNixEnv;
        }
      ) snabbs
    ) subQemus
  ) ((selectDpdk dpdkVersions kPackages))
) subKernelPackages;
```

```
$ NIX_COUNT_CALLS=1 ...
```

```
calls to 26 primops:
```

```
5529 map  
4910 attrNames  
2344 listToAttrs  
1908 foldl'  
1842 toString  
1842 sub  
1839 elemAt  
1838 isFunction
```

```
...
```

```
calls to 919 functions:
```

```
4405 'pushDownProperties' at /nix/var/nix/profiles/  
3796 anonymous function at /nix/var/nix/profiles/pe  
2583 anonymous function at /nix/var/nix/profiles/pe
```

# Do not try at home

\$ cat foo.nix

```
isEnabled:  
{  
  service.foo.enable = isEnabled;  
}
```

\$ cat bar.nix

```
{  
  imports = [ (import ./foo.nix true) ];  
}
```

```
diff --git a/nixos/lib/test-driver/test-driver.pl b/nixos
index 8ad0d67..838fbdd 100644
--- a/nixos/lib/test-driver/test-driver.pl
+++ b/nixos/lib/test-driver/test-driver.pl
@@ -34,7 +34,7 @@ foreach my $vlan (split / /, $ENV{VLANS
    if ($pid == 0) {
        dup2(fileno($pty->slave), 0);
        dup2(fileno($stdoutW), 1);
-       exec "vde_switch -s $socket" or _exit(1);
+       exec "vde_switch -tap tap0 -s $socket" or _exit(
    }
    close $stdoutW;
    print $pty "version\n";
```

**NixOps**



## Declarative NixOps (<https://github.com/input-output-hk/iohk-ops>)

```
$ cat config.yaml
args:
  configurationKey:
    tag: NixStr
    contents: foobar
  systemStart:
    tag: NixInt
    contents: 1506203091
  accessKeyId:
    tag: NixStr
    contents: foobar
name: deployment1
files:
- deployments/keypairs.nix
- deployments/security-groups.nix
- deployments/cardano-nodes.nix
- deployments/cardano-nodes-env-production.nix
- deployments/cardano-explorer.nix
- deployments/cardano-explorer-env-production.nix
- deployments/report-server.nix
- deployments/report-server-env-production.nix
```

```
$ iohk-ops deploy -c config.yaml
```

```
{
  defaults = { config, pkgs, ... }:
    { imports = [ ./ec2-info-example.nix ];
      deployment.targetEnv = "ec2";
      deployment.ec2.region = pkgs.lib.mkDefault "eu-west-1";
      deployment.ec2.instanceType = "t2.large";
    };

  backend2 = { ... }: {
    deployment.ec2.region = "us-east-1";
    deployment.ec2.tags.DummyTag = "some random blabla";
    deployment.ec2.tags.AnotherTag = "more blabla";
  };
}
```

```
{ ..., nodes }:  
  
{  
  backend1 = { config, pkgs, ... }:  
  {  
    services.my.ips = [  
      nodes.backend2.config.networking.publicIPv4  
    ];  
  };  
  
  backend2 = { config, pkgs, ... }:  
  {  
  };  
}
```

## 50+ NixOps machines

```
nixops --max-concurrent-copy N
```

```
deployment.hasFastConnection = true;
```

# Hydra

**[x] Declarative configuration**

**[x] github PRs**

**[x] github status**

**Impure Hydra?**



### *Faster builds*

Run as many build agents as you dare, on any type of machine, and let Buildkite distribute and coordinate the jobs. Configure build steps to target different types of build agents.



### *Powerful pipelines*

Configure build pipelines that run steps in parallel across build agents, wait for previous steps to complete, or pause the pipeline for team intervention, QA or testing.



### *Automation tools*

Per-build key/value and binary data storage allows to easily share data between build steps and across agents. Build assets on one server, retrieve them on another.



### *Easy setup*

Setting up a new build pipeline is no harder than running a build on your local machine, and the agent installs in one line of bash. Let your entire team become the build experts.



### *Beautiful build output*

Log output folding, live updates, emoji, and browsable build artifacts allow everyone on the team to see what's happening and to debug the cause of build failures.



### *Trigger via the web*

Automate manual workflows by allowing team members to manually trigger build pipelines via the web. Enjoy increased visibility, notification and logging.



### *Your code stays secure*

Buildkite never sees your source code or any secret keys. If you connect Buildkite with your GitHub or Bitbucket account we never request permission to read your code.



### *Your build scripts*

Define your build steps in any scripting language your server supports, with almost no Buildkite specific code. Simple, repeatable, versioned and future-proof.




### *Stable infrastructure*

You control the environment. Never worry about your build environment being updated without your control, or becoming out of date.




# Projects


## NixOS (the purely functional Linux distribution)

Jobset 	Description	Job status	Last evaluation
<a href="#">release-16.03</a>	NixOS 16.03 release branch	<span>24315</span> <span>275</span> <span>5</span>	2016-05-21 13:57:13
<a href="#">trunk-combined</a>	Combined NixOS/Nixpkgs unstable	<span>24243</span> <span>406</span> <span>1</span>	2016-05-21 13:57:03

## Nix (the purely functional package manager)

Jobset 	Description	Job status	Last evaluation
<a href="#">master</a>	Master branch	<span>1</span> <span>33</span>	2016-05-21 13:57:13

## Nixpkgs (Nix Packages collection)

Jobset 	Description	Job status	Last evaluation
<a href="#">trunk</a>	Trunk	<span>24006</span> <span>7798</span>	2016-05-21 13:57:13
<a href="#">staging</a>	Staging	<span>63</span> <span>31604</span>	2016-05-21 13:57:03

## NixOps (Deploying NixOS machines)



# Haskell

[x] multiple outputs

[x] stack2nix

## **Bootstrap custom nixpkgs/nixos project?**

- overlays
- pinned nixpkgs
- nixos modules
- custom pkgs
- nixops modules
- lib

# Thanks!

[domen@enlambda.com](mailto:domen@enlambda.com)