Nixing challenges of Kubernetes packaging with Nix

Arik Grahl

SysEleven

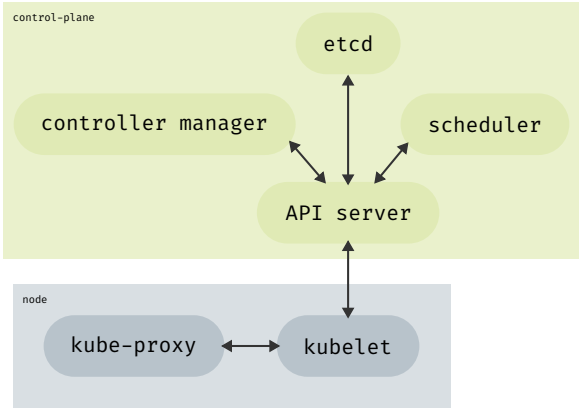
2024-11-13

echo $(whoami)

arik-grahl.de/talks/clc-2024

```yaml
---
apiVersion: v1
kind: Person
metadata:
  name: Arik Grahl
  pronouns: he/him
spec:
  work:
    company: SysEleven
    role: Software Engineer
  contact:
    web: www.arik-grahl.de
    gitlab: gitlab.com/arik.grahl
    github: github.com/arikgrahl
    linkedin: linkedin.com/in/arikgrahl
    mastodon: chaos.social/@arikgrahl
    matrix: @arik:matrix.arik-grahl.de
```

Kubernetes



- Kubernetes is an API
  - operator pattern
    - resources get reconciled
    - system state converges eventually towards defined state
  - strict Go types and protobuf
  - exposed as JSON or YAML

YAML

```yaml
---
a: &anchor
  x: baz
  y: 1.23
b: *anchor
c:
  <<: *anchor
  y: 4.56
d: { "x": "foo", "y": 7.89 }
e:
  - [true, NO]
  - [1, -1, 1.2, 1e6]
  - [1.2.3, foo, "bar"]
```

- verbose and repetitive
- not easily extensible
- challenging inheritance
- type confusion

Tooling to Generate Structured Documents

- instead of solving the root cause
  - just mitigating the problem
  - another level of indirection
- tooling to generate structured documents
  - Jsonnet
  - Kustomize
  - Helm and Helmfile

Helm (1/2)

- defacto standard for packaging
- rudimentary dependency management for downstream charts
- heavily based upon Golang's `text/template`

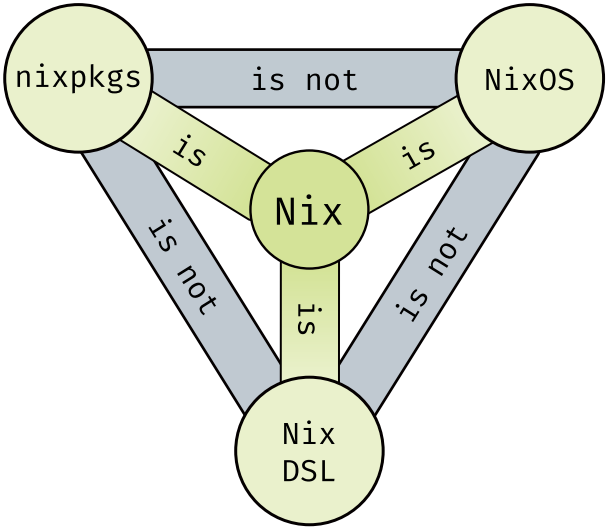
Helm (2/2)

```
{{- if .Values.rbac.create }}
{{- if and .Values.rbac.scope (not .Values.controller.scope.enabled) -}}
  {{ required "Invalid configuration …" (index (dict) ".") }}
{{- end }}
{{- if not .Values.rbac.scope -}}
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    {{- include "ingress-nginx.labels" . | nindent 4 }}
    {{- with .Values.controller.labels }}
    {{- toYaml . | nindent 4 }}
    {{- end }}
  name: {{ include "ingress-nginx.fullname" . }}
# …
{{- end }}
{{- end }}
```

Entering Nix

- Nix enables systems, which are
  - reproducible
  - declarative
  - reliable

What is Nix?



- Nix DSL
  - purely functional
  - lazy evaluated
  - purpose-built
- Nixpkgs
  - package manager
  - large mono repository
  - collection of over 100k packages

Nix as a Generic Builder (1/3)

```
{ pkgs ? import <nixpkgs> { } }:
pkgs.buildGoModule rec {
  pname = "kubectl";
  version = "1.31.2";
  src = pkgs.fetchFromGitHub rec {
    owner = "kubernetes";
    repo = owner;
    rev = "v${version}";
    hash = "sha256-L+x1a9wttu2OBY5T6AY8k91ystu0uZAGd3px4oNVptM=";
  };
  subPackages = [ "cmd/kubectl" ];
  vendorHash = null;
}
```

Nix as a Generic Builder (2/3)

```
{ pkgs ? import <nixpkgs> { } }:
let
  name = "nginx";
  version = "1.27.2";
  labels = { app = name; };
in
pkgs.writeText "deployment.yaml" (pkgs.lib.generators.toYAML { } {
  apiVersion = "apps/v1";
  kind = "Deployment";
  metadata = { inherit name labels; };
  spec = {
    selector.matchLabels = labels;
    template = {
      metadata = { inherit labels; };
      spec.containers = [{
        inherit name;
        image = "docker.io/${name}:${version}-alpine";
      }];
    };
  };
})
```

Nix handles Dependencies

```
{ pkgs ? import <nixpkgs> { } }:
let
  name = "todo-app";
  version = "1.2.3";
  database = import ./postgresql/ha; # pkgs.writeText …
  webapp = import ./golang/todo-app { inherit version; };
in
pkgs.linkFarm name [
  { name = "deployment.yaml"; path = webapp; }
  { name = "statefulset.yaml"; path = database; }
]
```

Profit from the Established Ecosystem

- no need to define low-level Kubernetes manifests
- `kubenix`
  - exposes Kubernetes core API
  - offers Helm wrapper
- bring your own builder script

```
{ pkgs ? import <nixpkgs> { }, system ? builtins.currentSystem }:
derivation rec {
  inherit system;
  name = "kube-prometheus-stack-62.7.0";
  src = fetchTarball { … };
  builder = pkgs.writeShellScript "builder.sh" ''
    ${pkgs.helmfile}/bin/helmfile template -f ${src}/helmfile.yaml > $out
  '';
}
```

Containers (1/2)

```yaml
---
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  containers:
    - name: nginx
      image: docker.io/nginx:v1.27.2-alpine # ⟸ just a reference
```

- generating YAML does not contribute to
  - software supply chain
    - security (SSCS)
    - management (SSCM)
  - software bill of materials (SBOM)

Containers (2/2)

```nix
{ pkgs ? import <nixpkgs> { } }:
let
  name = "nginx";
  version = pkgs.nginx.version;
  container = pkgs.dockerTools.buildLayeredImage {
    inherit name;
    tag = "v${version}";
    config.Entrypoint = [ "${pkgs.nginx}/bin/nginx" ];
    config.Cmd = [ "-g" "daemon off;" ];
  };
in
pkgs.writeText "pod.yaml" (pkgs.lib.generators.toYAML { } {
  apiVersion = "v1";
  kind = "Pod";
  metadata = { inherit name; };
  spec.containers = [{
    inherit name;
    image = "nix.internal/${container}"; # ⟸ served via Nix OCI registry
  }];
})
```

End-To-End Experience

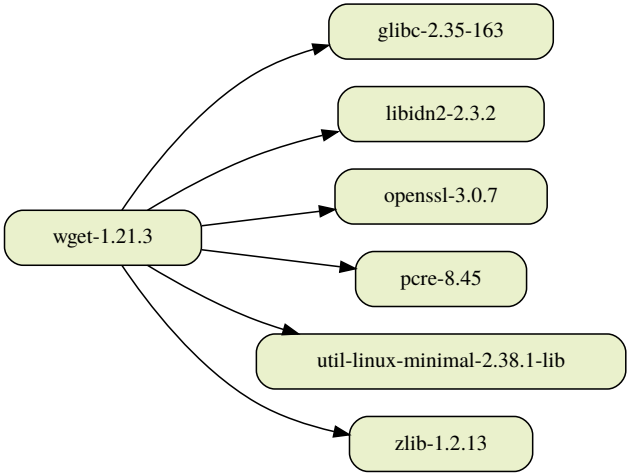
```
$ nix build -o result
$ kubectl apply -f ./result/
```

- one tool to build everything
- reproducible and pure builds
- GitOps friendly
- flexible towards custom tooling

Detailed Insights (1/2)

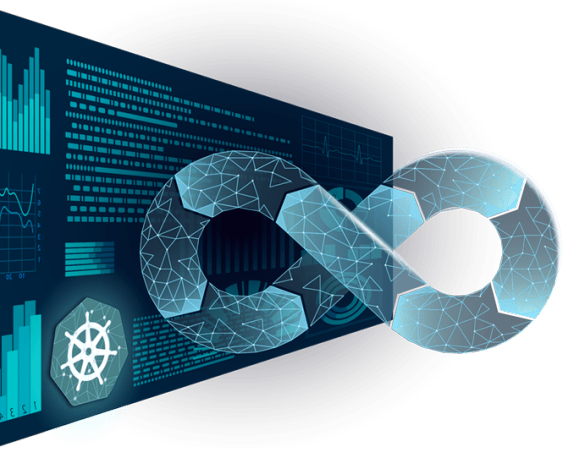
```
$ nix derivation show --recursive ./result \
  | jq '.[].inputDrvs | keys | .[]' -r \
  | sort | uniq -c | sort -r -h
1086 /nix/store/ysv…4sv-bash-5.2p32.drv
 621 /nix/store/rqn…w9j-stdenv-linux.drv
 525 /nix/store/qf9…7zr-mirrors-list.drv
 525 /nix/store/lvb…q95-curl-8.7.1.drv
 458 /nix/store/wql…4pc-stdenv-linux.drv
 149 /nix/store/v9i…9sd-python3-3.11.9.drv
 137 /nix/store/h3n…fmr-bootstrap-tools.drv
 132 /nix/store/3mx…br0-perl-5.38.2.drv
 108 /nix/store/nz9…7fh-pkg-config-wrapper-0.29.2.drv
  94 /nix/store/z46…lsz-wrap-python-hook.drv
 […]
```

Detailed Insights (2/2)



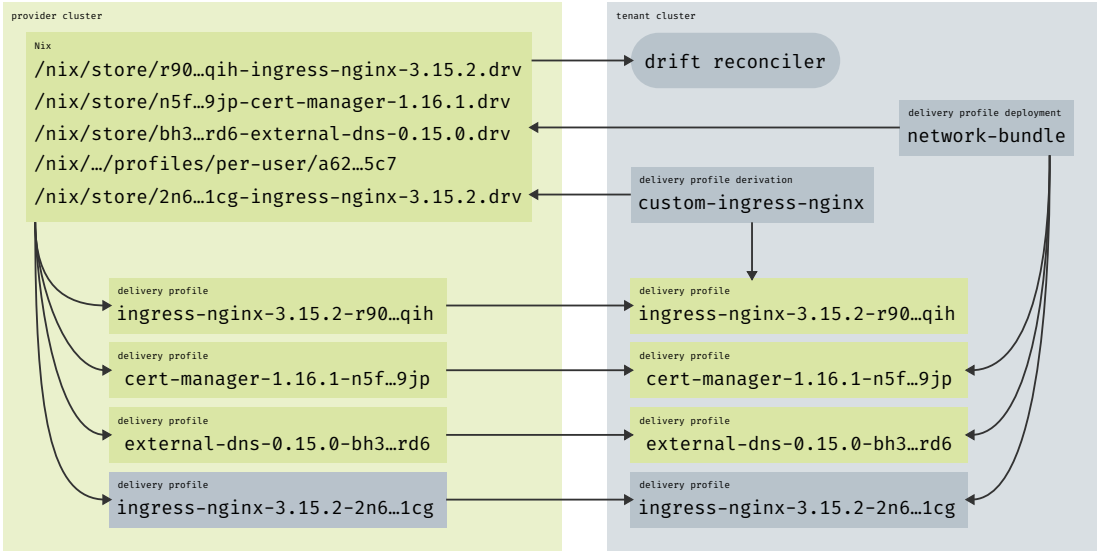
- `sbomnix`
  - generate SBOMs
  - scan for security vulnerabilites
  - query for licenses
  - provenance attestation
- other tools
  - `nix2sbom`
  - `vulnix`

MetaKube Accelerator (1/2)



- SysEleven's vision: Abstract away Nix (for most users)
- MetaKube Accelerator
  - API-centric
  - `kubectl`-first

MetaKube Accelerator (2/2)



Conclusion

- defining Kubernetes manifests with Nix
  - reasonable amount of abstraction
  - sophisticated dependency management
  - building on top of established ecosystem
- building containers with Nix
  - better containers than with Docker
  - simplified tooling with end-to-end experience
  - detailed insights
- MetaKube Accelerator abstracts away Nix

Questions & Answers

arik-grahl.de/talks/clc-2024

sys11.it/mka