NIX AND KUBERNETES

DEPLOYMENTS DONE RIGHT

VOLODYMYR PUZANOV

@FARCALLER

DEPLOYING TO KUBERNETES

WHAT'S WRONG?

YAML

ISSUES WITH YAML

- Verbose & repetitive

ISSUES WITH YAML

- Verbose & repetitive
- Tricky to extend

ISSUES WITH YAML

- Verbose & repetitive
- Tricky to extend
- Inheritance issues

LET'S BUILD TOOLING!

ISSUES WITH TOOLING

- What's Jsonnet?
- Is Kustomize enough for our use case?
- How do we use community code?
- How does that Helm chart work?

HELM

```
{{- if or (and .Values.controller.autoscaling.enabled (gt (.Values.controller.autosca
apiVersion: {{ ternary "policy/v1" "policy/v1beta1" (semverCompare ">=1.21.0-0" .Capa
kind: PodDisruptionBudget
metadata:
  labels:
{{- include "ingress-nginx.labels" . | nindent 4 }}
    app.kubernetes.io/component: controller
{{- with .Values.controller.labels }}
{{- toYaml . | nindent 4 }}
{{- end }}
  name: {{ include "ingress-nginx.controller.fullname" . }}
  namespace: {{ .Release.Namespace }}
{{- if .Values.controller.annotations }}
annotations: {{ toYaml .Values.controller.annotations | nindent 4 }}
{{- end }}
spec:
  selector:
    matchLabels:
{{- include "ingress-nginx.selectorLabels" . | nindent 6 }}
      app.kubernetes.io/component: controller
{{- if and .Values.controller.minAvailable (not (hasKey .Values.controller "maxUnavai
  minAvailable: {{ .Values.controller.minAvailable }}
{{- else if .Values.controller.maxUnavailable }}
  maxUnavailable: {{ .Values.controller.maxUnavailable }}
```

```
{{- if or (and .Values.controller.autoscaling.enabled (gt (.Values.controller.autosca
apiVersion: {{ ternary "policy/v1" "policy/v1beta1" (semverCompare ">=1.21.0-0" .Capa
kind: PodDisruptionBudget
metadata:
  labels:
{{- include "ingress-nginx.labels" . | nindent 4 }}
    app.kubernetes.io/component: controller
{{- with .Values.controller.labels }}
{{- toYaml . | nindent 4 }}
{{- end }}
  name: {{ include "ingress-nginx.controller.fullname" . }}
  namespace: {{ .Release.Namespace }}
{{- if .Values.controller.annotations }}
annotations: {{ toYaml .Values.controller.annotations | nindent 4 }}
{{- end }}
spec:
  selector:
    matchLabels:
{{- include "ingress-nginx.selectorLabels" . | nindent 6 }}
      app.kubernetes.io/component: controller
{{- if and .Values.controller.minAvailable (not (hasKey .Values.controller "maxUnavai
  minAvailable: {{ .Values.controller.minAvailable }}
{{- else if .Values.controller.maxUnavailable }}
  maxUnavailable: {{ .Values.controller.maxUnavailable }}
```

```
{{- if or (and .Values.controller.autoscaling.enabled (gt (.Values.controller.autosca
apiVersion: {{ ternary "policy/v1" "policy/v1beta1" (semverCompare ">=1.21.0-0" .Capa
kind: PodDisruptionBudget
metadata:
  labels:
{{- include "ingress-nginx.labels" . | nindent 4 }}
    app.kubernetes.io/component: controller
{{- with .Values.controller.labels }}
{{- toYaml . | nindent 4 }}
{{- end }}
  name: {{ include "ingress-nginx.controller.fullname" . }}
  namespace: {{ .Release.Namespace }}
{{- if .Values.controller.annotations }}
annotations: {{ toYaml .Values.controller.annotations | nindent 4 }}
{{- end }}
spec:
  selector:
    matchLabels:
{{- include "ingress-nginx.selectorLabels" . | nindent 6 }}
      app.kubernetes.io/component: controller
{{- if and .Values.controller.minAvailable (not (hasKey .Values.controller "maxUnavai
  minAvailable: {{ .Values.controller.minAvailable }}
{{- else if .Values.controller.maxUnavailable }}
  maxUnavailable: {{ .Values.controller.maxUnavailable }}
```

```
{{- if or (and .Values.controller.autoscaling.enabled (gt (.Values.controller.autosca
apiVersion: {{ ternary "policy/v1" "policy/v1beta1" (semverCompare ">=1.21.0-0" .Capa
kind: PodDisruptionBudget
metadata:
  labels:
{{- include "ingress-nginx.labels" . | nindent 4 }}
    app.kubernetes.io/component: controller
{{- with .Values.controller.labels }}
{{- toYaml . | nindent 4 }}
{{- end }}
  name: {{ include "ingress-nginx.controller.fullname" . }}
  namespace: {{ .Release.Namespace }}
{{- if .Values.controller.annotations }}
annotations: {{ toYaml .Values.controller.annotations | nindent 4 }}
{{- end }}
spec:
  selector:
    matchLabels:
{{- include "ingress-nginx.selectorLabels" . | nindent 6 }}
      app.kubernetes.io/component: controller
{{- if and .Values.controller.minAvailable (not (hasKey .Values.controller "maxUnavai
  minAvailable: {{ .Values.controller.minAvailable }}
{{- else if .Values.controller.maxUnavailable }}
  maxUnavailable: {{ .Values.controller.maxUnavailable }}
```

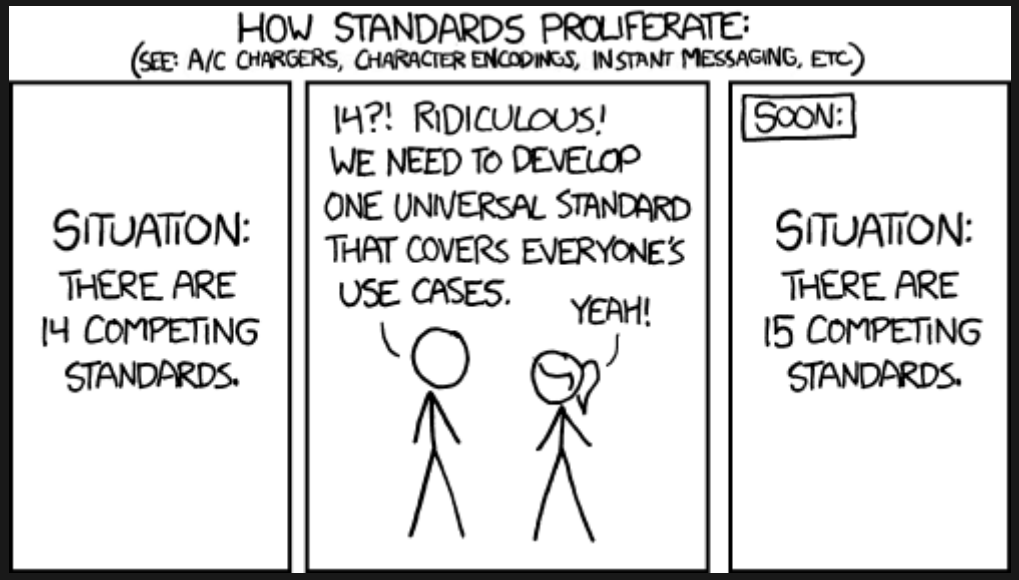
a **string templating language** used for **structured data**

— Xe

NIX

AN OBVIOUS SOLUTION!

AN OBVIOUS SOLUTION!



DERIVATION
THAT OUTPUTS YAML

```
let
  name = "mailpusher";
  labels = { "app.kubernetes.io/name" = name; };
in
{
  apiVersion = "apps/v1";
  kind = "Deployment";
  metadata = {
    inherit name labels;
  };
  spec = {
    revisionHistoryLimit = 3;
    selector.matchLabels = labels;
    template.metadata = { inherit labels; };
```

```
let
  name = "mailpusher";
  labels = { "app.kubernetes.io/name" = name; };
in
{
  apiVersion = "apps/v1";
  kind = "Deployment";
  metadata = {
    inherit name labels;
  };
  spec = {
    revisionHistoryLimit = 3;
    selector.matchLabels = labels;
    template.metadata = { inherit labels; };
```

```
let
  name = "mailpusher";
  labels = { "app.kubernetes.io/name" = name; };
in
{
  apiVersion = "apps/v1";
  kind = "Deployment";
  metadata = {
    inherit name labels;
  };
  spec = {
    revisionHistoryLimit = 3;
    selector.matchLabels = labels;
    template.metadata = { inherit labels; };
```

```
let
  name = "mailpusher";
  labels = { "app.kubernetes.io/name" = name; };
in
{
  apiVersion = "apps/v1";
  kind = "Deployment";
  metadata = {
    inherit name labels;
  };
  spec = {
    revisionHistoryLimit = 3;
    selector.matchLabels = labels;
    template.metadata = { inherit labels; };
```

ONE TOOL

```
gatewayResources = pkgs.fetchurl {
  url = "https://github.com/kubernetes-sigs/gateway-api/" +
        "releases/download/v0.7.1/standard-install.yaml";
  hash = "sha256-SOhUhEYTngjUec66SrhWGSQmSkZENw/7Lo+9tUsIgWQ=";
};
```

```
configconnector-operator = pkgs.fetchurl {
  url = "https://storage.googleapis.com/configconnector-operator/" +
        "latest/release-bundle.tar.gz";
  downloadToTemp = true;
  recursiveHash = true;
  hash = "sha256-2WI4vNmbD/cbNn8MEU6fAgxE6FckcgsWihACead0MIE=";
  postFetch = ''
    mkdir $out
    tar xzvf $downloadedFile -C $out
  '';
};
```


RUN KUSTOMIZE THROUGH NIX?
RUN JSONNET THROUGH NIX?
RUN HELM THROUGH NIX?
imgflip.com

```
mkDerivation {
    ...

    installPhase = ''
      ${pkgs.kubernetes-helm}/bin/helm pull \
      --repo "${repo}" --version "${version}" \
      "${chart}" -d $OUT_DIR --untar
      mv $OUT_DIR/${chart}/* "$out"
    '';

    outputHashMode = "recursive";
    outputHashAlgo = "sha256";
    outputHash = ...;
};
```

```
mkDerivation {
    ...

    installPhase = ''
      ${pkgs.kubernetes-helm}/bin/helm pull \
      --repo "${repo}" --version "${version}" \
      "${chart}" -d $OUT_DIR --untar
      mv $OUT_DIR/${chart}/* "$out"
    '';

    outputHashMode = "recursive";
    outputHashAlgo = "sha256";
    outputHash = ...;
};
```

```
mkDerivation {
    ...

    installPhase = ''
      ${pkgs.kubernetes-helm}/bin/helm pull \
      --repo "${repo}" --version "${version}" \
      "${chart}" -d $OUT_DIR --untar
      mv $OUT_DIR/${chart}/* "$out"
    '';

    outputHashMode = "recursive";
    outputHashAlgo = "sha256";
    outputHash = ...;
};
```

BUT WHY?

- Re-use the thousands of existing charts

BUT WHY?

- Re-use the thousands of existing charts
- Cache at the nix level

BUT WHY?

- Re-use the thousands of existing charts
- Cache at the nix level
- Easy[1] modification of resources


nix helm

NIXPKGS FOR HELM CHARTS

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
let charts = nixhelm.chartsDerivations.${system}; in
pkgs.lib.pipe
{
  name = "ghost";
  chart = charts.groundhog2k.ghost;
  namespace = "farcaller-net";
  values = {
    ingress.enabled = false;
    storage.persistentVolumeClaimName = "farcaller-ghost-content";
  };
} [
kubelib.buildHelmChart
builtins.readFile
kubelib.fromYAML
(foldResources [
  ./namespace.yaml
  ./storage.yaml
  ./secretstore.yaml
  ./mailgun.yaml
  ./mysql.yaml
])
(map patchDeployment)
kubelib.mkList
kubelib.toYAMLFile
];
```

```
patchDeployment = obj:
  if obj.kind == "Deployment" then
    (lib.recursiveUpdate object {
      spec.replicas = 2;
    })
  else obj;
```

SOME ACTUALLY AMAZING THINGS

```
let
  isGlobalService = obj: obj.kind == "Service"
  && obj.metadata.annotations."io.cilium/global-service" or "" == "true";
in pipe app [
  (app: ./../${app})
  toString
  getFlake
  (f: f.outputs.packages.${system}.kubernetesConfiguration)
  readFile
  kubelib.fromYAML
  head
  (y: y.items)
  (filter isGlobalService)
  (map (updateManyAttrsByPath [
    {
      path = [ "metadata" "namespace" ];
      update = old: app;
    }
    {
      path = [ "metadata" "labels" ];
      update = old: { ingress-reflector = "true"; };
    }
  ]))
]
```

```
let
  isGlobalService = obj: obj.kind == "Service"
  && obj.metadata.annotations."io.cilium/global-service" or "" == "true";
in pipe app [
  (app: ./../${app})
  toString
  getFlake
  (f: f.outputs.packages.${system}.kubernetesConfiguration)
  readFile
  kubelib.fromYAML
  head
  (y: y.items)
  (filter isGlobalService)
  (map (updateManyAttrsByPath [
    {
      path = [ "metadata" "namespace" ];
      update = old: app;
    }
    {
      path = [ "metadata" "labels" ];
      update = old: { ingress-reflector = "true"; };
    }
  ]))
]
```

```
let
  isGlobalService = obj: obj.kind == "Service"
  && obj.metadata.annotations."io.cilium/global-service" or "" == "true";
in pipe app [
  (app: ./../${app})
  toString
  getFlake
  (f: f.outputs.packages.${system}.kubernetesConfiguration)
  readFile
  kubelib.fromYAML
  head
  (y: y.items)
  (filter isGlobalService)
  (map (updateManyAttrsByPath [
    {
      path = [ "metadata" "namespace" ];
      update = old: app;
    }
    {
      path = [ "metadata" "labels" ];
      update = old: { ingress-reflector = "true"; };
    }
  ]))
]
```

```
let
  isGlobalService = obj: obj.kind == "Service"
  && obj.metadata.annotations."io.cilium/global-service" or "" == "true";
in pipe app [
  (app: ./../${app})
  toString
  getFlake
  (f: f.outputs.packages.${system}.kubernetesConfiguration)
  readFile
  kubelib.fromYAML
  head
  (y: y.items)
  (filter isGlobalService)
  (map (updateManyAttrsByPath [
    {
      path = [ "metadata" "namespace" ];
      update = old: app;
    }
    {
      path = [ "metadata" "labels" ];
      update = old: { ingress-reflector = "true"; };
    }
  ]))
]
```

```
let
  isGlobalService = obj: obj.kind == "Service"
  && obj.metadata.annotations."io.cilium/global-service" or "" == "true";
in pipe app [
  (app: ./../${app})
  toString
  getFlake
  (f: f.outputs.packages.${system}.kubernetesConfiguration)
  readFile
  kubelib.fromYAML
  head
  (y: y.items)
  (filter isGlobalService)
  (map (updateManyAttrsByPath [
    {
      path = [ "metadata" "namespace" ];
      update = old: app;
    }
    {
      path = [ "metadata" "labels" ];
      update = old: { ingress-reflector = "true"; };
    }
  ]))
]
```

- Any source

- Any source
- Combinations

- Any source
- Combinations
- Modifications

- Any source
- Combinations
- Modifications
- All in nix

THE GRAND DESIGN

ARTEFACTS

```
nix build -o result
```

```
kubectl apply -f result
```

UPDATES

```
nix flake update
```

HOW DOES IT GET INTO THE CLUSTER?

ARGOCD

NIXOS/NIX:LATEST

```
if [ "$PARAM_VALUES" != "" ]; then
  echo -ne "With values\n" >/dev/stderr
  echo "$PARAM_VALUES" > values.json
  nix-shell -p git --run 'git add values.json'
fi

nix build ".#kubernetesConfiguration"
```

```
if [ "$PARAM_VALUES" != "" ]; then
  echo -ne "With values\n" >/dev/stderr
  echo "$PARAM_VALUES" > values.json
  nix-shell -p git --run 'git add values.json'
fi

nix build ".#kubernetesConfiguration"
```

```
if [ "$PARAM_VALUES" != "" ]; then
  echo -ne "With values\n" >/dev/stderr
  echo "$PARAM_VALUES" > values.json
  nix-shell -p git --run 'git add values.json'
fi

nix build ".#kubernetesConfiguration"
```

IN THE END, YOU GET A DIFF

apps/StatefulSet/mysql/mysql

```
    app.kubernetes.io/name: mysql                                                                    |    app.kubernetes.io/name: mysql
    argocd.argoproj.io/instance: web-mysql                                                           |    argocd.argoproj.io/instance: web-mysql
    helm.sh/chart: mysql-9.11.0                                                                      |    helm.sh/chart: mysql-9.12.1
managedFields:                                                                                       | managedFields:
  - apiVersion: apps/v1                                                                              |   - apiVersion: apps/v1
  metadata:                                                                                          |   metadata:
    annotations:                                                                                     |     annotations:
      checksum/configuration: cf6a9f3129ed4245a6988020d32be9576d17cc21ccec3f11d5f47d99c11e9387       |       checksum/configuration: 2b82a476ea229525ffa0fb537c626b3dd2a119a08b8e21ae14239d18273f3bae
      kubectl.kubernetes.io/restartedAt: '2023-07-31T11:01:58Z'                                      |       kubectl.kubernetes.io/restartedAt: '2023-07-31T11:01:58Z'
    creationTimestamp: null                                                                          |     creationTimestamp: null
      app.kubernetes.io/managed-by: Helm                                                             |       app.kubernetes.io/managed-by: Helm
      app.kubernetes.io/name: mysql                                                                  |       app.kubernetes.io/name: mysql
      helm.sh/chart: mysql-9.11.0                                                                    |       helm.sh/chart: mysql-9.12.1
  spec:                                                                                              |   spec:
    affinity:                                                                                        |     affinity:
                key: mysql-root-password                                                             |                 key: mysql-root-password
                name: mysql-passwords                                                                |                 name: mysql-passwords
        image: 'docker.io/bitnami/mysql:8.0.34-debian-11-r8'                                         |       image: 'docker.io/bitnami/mysql:8.0.34-debian-11-r31'
        imagePullPolicy: IfNotPresent                                                                |       imagePullPolicy: IfNotPresent
        livenessProbe:                                                                               |       livenessProbe:
                key: mysql-root-password                                                             |                 key: mysql-root-password
                name: mysql-passwords                                                                |                 name: mysql-passwords
        image: 'docker.io/bitnami/mysqld-exporter:0.15.0-debian-11-r0'                               |       image: 'docker.io/bitnami/mysqld-exporter:0.15.0-debian-11-r24'
        imagePullPolicy: IfNotPresent                                                                |       imagePullPolicy: IfNotPresent
        livenessProbe:                                                                               |       livenessProbe:
            ".snapshot" -not -name "lost+found" | xargs -r chown -R                                  |           ".snapshot" -not -name "lost+found" | xargs -r chown -R
            "1001:1001"                                                                              |           "1001:1001"
        image: 'docker.io/bitnami/os-shell:11-debian-11-r16'                                         |       image: 'docker.io/bitnami/os-shell:11-debian-11-r43'
        imagePullPolicy: IfNotPresent                                                                |       imagePullPolicy: IfNotPresent
        name: volume-permissions                                                                     |       name: volume-permissions
```

CAKE

https://github.com/farcaller/cake

SIMPLE DEPLOYMENTS

```
{ nixhelm, lib, ... }: {
  apps.istio-base = {
    kind = "helm";
    release = nixhelm.chartsMetadata.istio.base;
    namespace = "istio-system";
    createNamespace = true;
  };
};
```

PASSING VALUES

```nix
{ nixhelm, ... }: {
  apps.victoria-metrics-operator = {
    kind = "helm";
    release = nixhelm.chartsMetadata.victoriametrics.victoria-metrics-operator;
    createNamespace = true;
    values = {
      operator.prometheus_converter_add_argocd_ignore_annotations = true;
      operator.enable_converter_ownership = true;
    };
  };
};
```

FULL VALUES SCHEMA SUPPORT

```nix
{ ... }: {
  apps.private-ghcr = {
    kind = "nix";
    namespace = "external-secrets";
    valuesSchema = with lib; {
      cluster = mkOption { type = types.str; };
    };
  };
}
```

CLUSTER-SPECIFIC GENERATION

```nix
{ nixhelm, lib, ... }: {
  apps.promtail = {
    kind = "helm";
    release = nixhelm.chartsMetadata.grafana.promtail;
    namespace = "monitoring-system";
    namespaces = [ "monitoring-system" "kube-system" ];

    valuesGenerator = { clusterName, cluster, values, config, ... }: {
      extraArgs = [ "-client.external-labels=cluster=${clusterName}" ];
      serviceMonitor.enabled = true;
    };
  };
}
```

DEFINING THE CLUSTERS

```nix
{ config, lib, options, ... }: {
  clusters.nixbox = {
    ciliumID = 11;
    apiServer = "10.224.1.41";

    enableMonitoring = true;

    apps = {
      cilium-bgp.enable = true;
      cert-manager-cilium-issuer = {
        enable = true;
        values.vault_mount = "k8s-nixbox";
      };
    };
  };
}
```

DEFINING THE CLUSTERS

```nix
{ config, lib, options, ... }: {
  clusters.nixbox = {
    ciliumID = 11;
    apiServer = "10.224.1.41";

    enableMonitoring = true;

    apps = {
      cilium-bgp.enable = true;
      cert-manager-cilium-issuer = {
        enable = true;
        values.vault_mount = "k8s-nixbox";
      };
    };
  };
}
```

DEFINING THE CLUSTERS

```nix
{ config, lib, options, ... }: {
  clusters.nixbox = {
    ciliumID = 11;
    apiServer = "10.224.1.41";

    enableMonitoring = true;

    apps = {
      cilium-bgp.enable = true;
      cert-manager-cilium-issuer = {
        enable = true;
        values.vault_mount = "k8s-nixbox";
      };
    };
  };
}
```

DEFINING THE CLUSTERS

```nix
{ config, lib, options, ... }: {
  clusters.nixbox = {
    ciliumID = 11;
    apiServer = "10.224.1.41";

    enableMonitoring = true;

    apps = {
      cilium-bgp.enable = true;
      cert-manager-cilium-issuer = {
        enable = true;
        values.vault_mount = "k8s-nixbox";
      };
    };
  };
}
```

the cake is a lie!

THANKS!

Xe cadey@pony.social

Raito Bezarius raito@nixos.paris

https://vfp.in/nixcon2023

www.farcaller.net

farcaller@hdev.im

mail@farcaller.net