

FREEDOM THROUGH BOUNDARIES:

Building Configurations That Age Well

Bogdan Stancu, Adobe



Who's that dog?

- He is Terry
- 3 Years old
- Training for almost 3 years



Dog training and freedom

- Freedom isn't the absence of rules
- Clear boundaries create trust
- Strict training enables off-leash freedom



What is Helm?

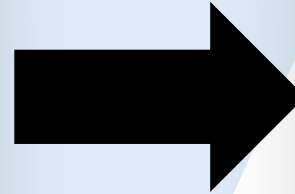
A templating engine for Kubernetes manifests



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: goodboy
data:
  goodboy: {{ .Values.config.goodboy }}
```



```
config:
  goodboy: "Terry"
```



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: goodboy
data:
  goodboy: "Terry"
```

What is an OpenTelemetry Collector?

An agent that collects, transforms and forwards telemetry signals



```
receivers:
  otlp:
    protocols:
      grpc:
        endpoint: 0.0.0.0:4317

exporters:
  otlp:
    endpoint: otelcol.observability.svc.cluster.local:443

service:
  extensions: []
  pipelines:
    traces:
      receivers: [otlp]
      processors: []
      exporters: [otlp]
```



Every field is a promise

Every value is a contract

It starts small

- Small Helm Chart
- Deploys 3 sets of pre-configured collectors
- Global -> all collectors
- Config for individual signals



```
global:
    //...
metrics:
    //...
traces:
    //...
logs:
    //...
```



User request: existingSecret

- Users want to deploy their own secrets
- They need a way to mount existing secrets

```
global:  
  secrets:  
    ingestionKey: ""  
    imagePullSecret: ""  
  
metrics:  
  useExistingSecret: ""  
  existingSecretName: ""  
  
traces:  
  //...  
  
logs:  
  //...
```

Security request: externalSecrets

- **Security:** All secrets are kept in secret management solutions
- **We use the External Secrets Operator**

```
global:
  secrets:
    ingestionKey: ""
    imagePullSecret: ""
    externalSecrets:
      enabled: "true"
      imagePullSecretPath: ""

metrics:
  useExistingSecret: ""
  existingSecretName: ""
traces:
  //...
logs:
  //...
```

Internal testing

- Feature flags for internal testing
- Users have access to internal implementation details

```
global:
  secrets:
    ingestionKey: ""
    imagePullSecret: ""
    externalSecrets:
      enabled: true
      imagePullSecretPath: ""
      mountSecretAsFile: false

metrics:
  useExistingSecret: ""
  existingSecretName: ""
traces:
  //...
logs:
  //...
```

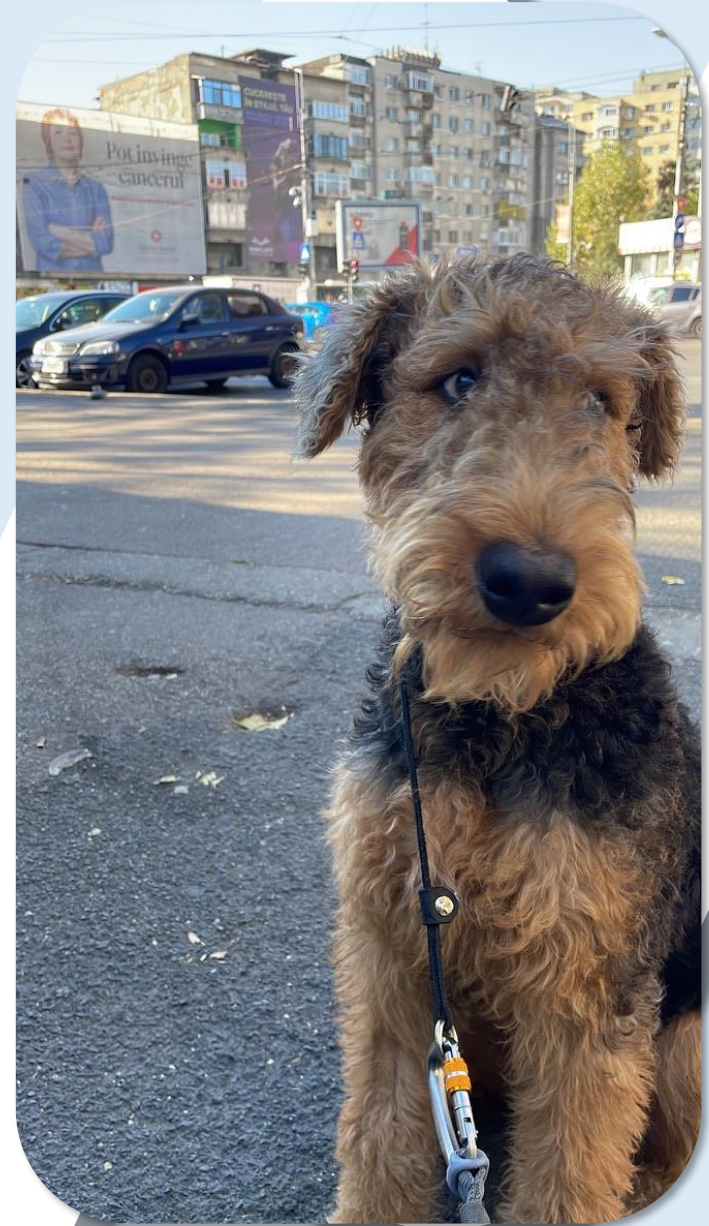
Flooded with edge cases

```
global:
  secrets:
    imagePullSecret: ""          # testing only
    ingestionSecret: ""         # testing only
    useExternalSecrets: true
    externalSecretsProvider: ""  # vault | aws | gcp
    externalSecretsNamespace: ""
    imagePullSecretPath: ""
    ingestionSecretNamePath: ""
    mountSecretAsFile: false
    mountSecretAsEnv: true
    allowInsecureSecrets: false
    skipSecretValidation: false
    legacySecretMode: false
    legacySecretPrefix: ""

  metrics:
    secrets:
      useExistingSecret:
      inheritGlobalSecrets: true
      allowPlaintextMetricsAuth: false

  traces:
    //...

  logs:
    //...|
```



Incompatibility allowed to coexist

- Auto-generated or pre-existing secret?
- Should throw error

```
global:
  secrets:
    imagePullSecret: ""      # testing only
    ingestionSecret: ""     # testing
    autoGenerateSecrets: true
    existingSecretName: "ingestion-secret"

metrics:
  //...

traces:
  //...

logs:
  //...
|
```

Version v1.0.x forever

- No breaking changes meant no real fixes
- The config grew, but never healed
- Temporary decisions became permanent
- Fear dictates the roadmap



```
## v1.0.27
```

- Fix edge case in secret resolution
- Improve compatibility
- Minor refactor

```
## v1.0.34
```

- Another fix
- Another exception
- Another workaround

Not testing the whole config matrix

- Too many options to test manually
- No automated coverage of combinations
- Bugs depended on configuration shape
- We stopped touching risky areas



featureA	featureB	featureC	
✓	✓	✓	← tested
✓	✓	x	
✓	x	✓	
x	✓	✓	
x	x	✓	
...			

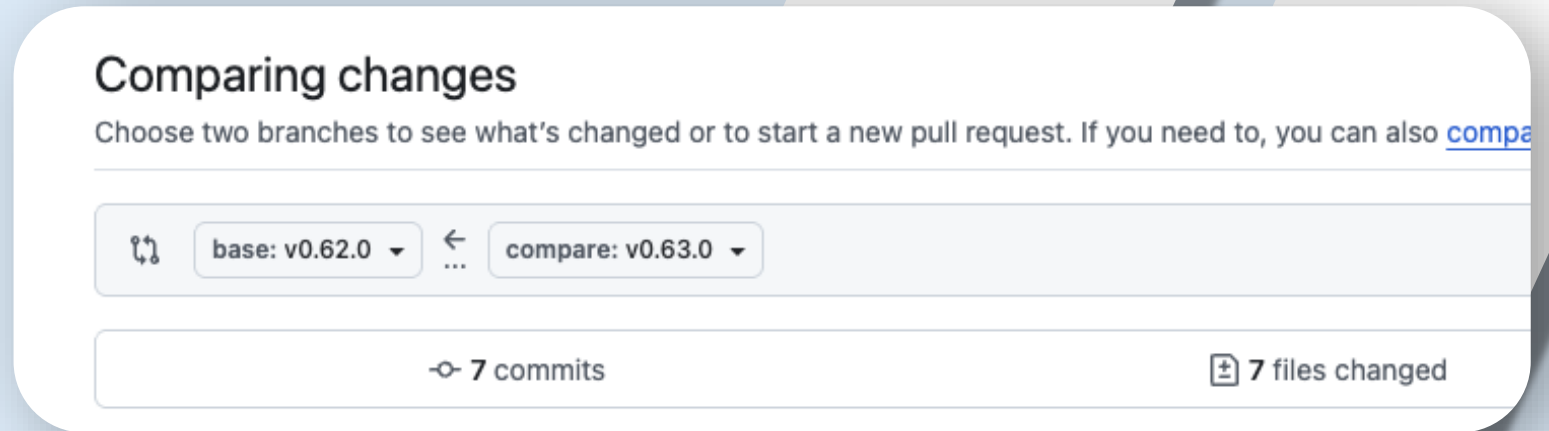


Some checks were not successful

1 failing, 25 skipped, 166 successful checks

Commit history is the communication

- No changelog
- No release notes
- Users left to guess



Documentation replaced design

- Docs compensated for poor design
- Comments became the real API
- Misuse was expected, then criticized



```
# DO NOT use with externalSecrets  
# Unless you also set legacyMode  
# This will break in some cases  
# See docs section 4.3.2
```

```
| useExistingSecret: true
```

Create boundaries

- Allowed configurations define the contract
- Everything else is explicitly rejected
- If we accept it, we support it



Make invalid states unavailable

- **Mutually exclusive options enforced**
- **Errors over silent behavior**
- **Broken combinations fail fast**



```
$ helm install otel ./chart -f values.yaml
```

```
Error: INSTALLATION FAILED:
```

```
Invalid configuration:
```

```
- secret.source=external is incompatible  
  with legacySecretFallback
```

Design for the user not the maintainer

- Users need to know what happens not how it happens
- Hide implementation details
- Internals remain changeable



```
# Maintainer-friendly  
useLegacyResolver: true  
enableExternalProvider: false  
fallbackToSecretManager: true
```

```
# User-friendly  
secretSource: kubernetes
```

Explicit over implicit defaults

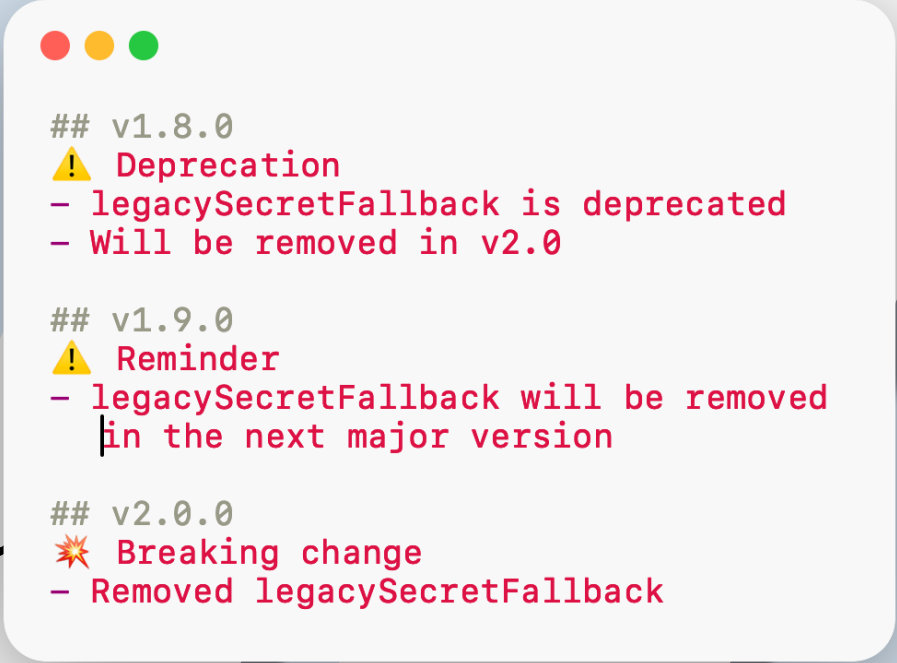
- Defaults are visible and well documented
- Behavior never changes silently
- Same config = same outcome



```
# Explicit intent always wins  
secretSource: kubernetes  
  
# If this ever behaves differently,  
# we shipped a breaking change.
```

Breaking changes are planned events

- Never fear breaking changes
- Communicate early and clearly
- Provide migration paths
- Treat backwards compatibility as a burden



```
## v1.8.0
⚠ Deprecation
- legacySecretFallback is deprecated
- Will be removed in v2.0

## v1.9.0
⚠ Reminder
- legacySecretFallback will be removed
  in the next major version

## v2.0.0
💥 Breaking change
- Removed legacySecretFallback
```

Test the whole config matrix

- Test all combinations not just default ones
- Don't try to predict behavior
- All combinations should either work or throw errors



Fail on unknown fields

- Helm ignores typos
- Field not recognised, reject it
- Validate early, fail fast

A white rounded rectangle with a shadow, containing a code snippet. At the top left of the rectangle are three colored circles: red, yellow, and green. The code is in a monospaced font, with some parts in red and some in teal.

```
metrics:  
  exprters: // ← typo  
  sendToCortex: true
```

Observability of the resolved config

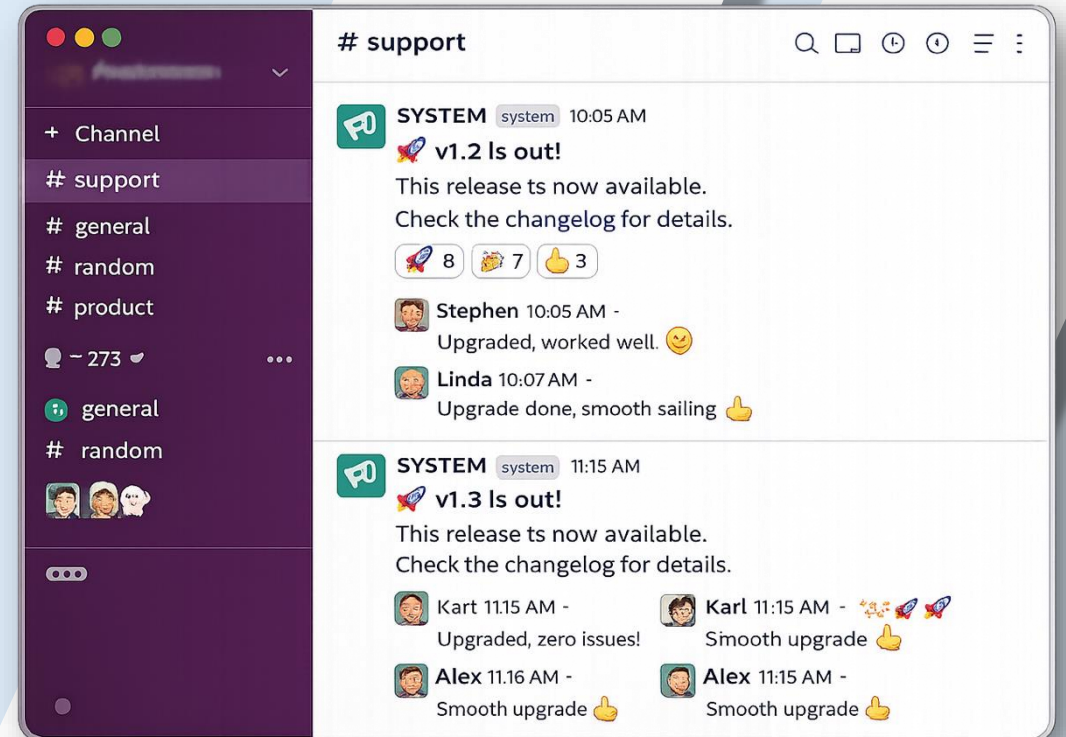
- Users see what they configure, not what is getting applied
- Provide a way to see the final merged config
- Users should not require in depth knowledge to debug



```
$ collector --dump-config  
receivers: otlp (grpc:4317, http:4318)  
exporters: otlp (endpoint: tempo:4317)  
secret: mounted from vault/otel-secret
```

Upgrades must be boring

- Upgrade guides
- Config migration paths
- Compatibility matrix between config versions



Boundaries enable freedom

- **Not just for the users**
- **Not just for maintainers**
- **For the whole ecosystem
using that product**





Thank you!