YAMLをテストする

```
%YAML 1.2
---
YAML: YAML Ain't Markup Language

What It Is: YAML is a human friendly data serialization
  standard for all programming languages.

YAML Resources:
  YAML 1.2 (3rd Edition): http://yaml.org/spec/1.2/spec.html
  YAML 1.1 (2nd Edition): http://yaml.org/spec/1.1/
  YAML 1.0 (1st Edition): http://yaml.org/spec/1.0/
  YAML Issues Page: https://github.com/yaml/yaml/issues
  ...
```

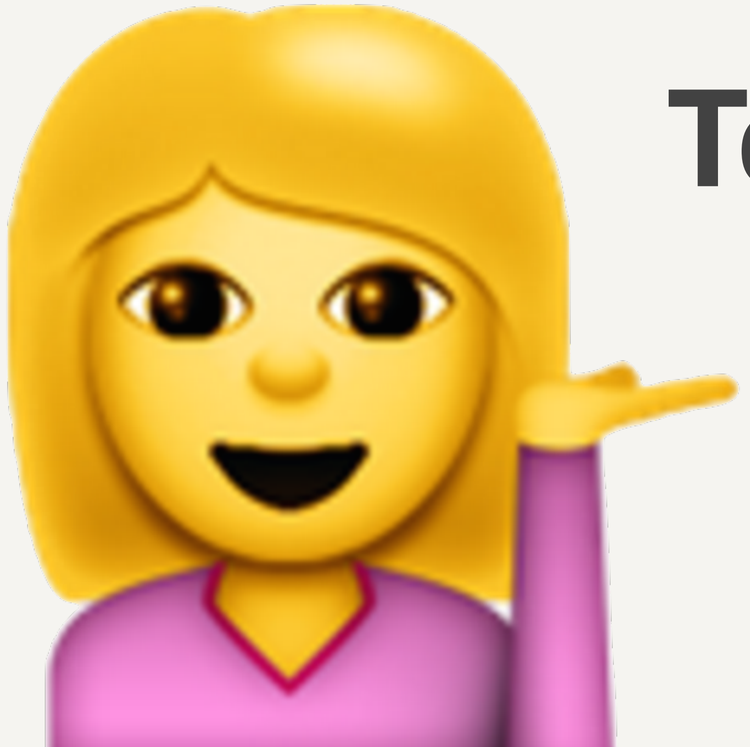
@b4b4r07 (Feb 25, 2019) / mercari.go #6

BABAROT / @b4b4r07

Mercari, Inc.
SRE, Microservices Platform

Blog / **tellme.tokyo**

Question:



Kubernetes YAML や
Terraform は書きますか？

Infrastructure as Code の浸透

＊以下「IaC」と記す

IaC の浸透

- Terraform や Kubernetes の普及で状態・定義をコードにすることが多くなった
- インフラ領域以外においても、ソフトウェアの状態やその設定を JSON や YAML といった言語で持つことが多くなった

https://trends.google.co.jp/trends/explore?date=today%205-y&q=infrastructure%20as%20code

IaC とは

- インフラの状態を設定ファイルで書く
- ソフトウェア開発の手法を応用できる
  - レビュー
  - テスト
  - etc

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
      - containerPort: 80
```

Kubernetes Pod の YAML

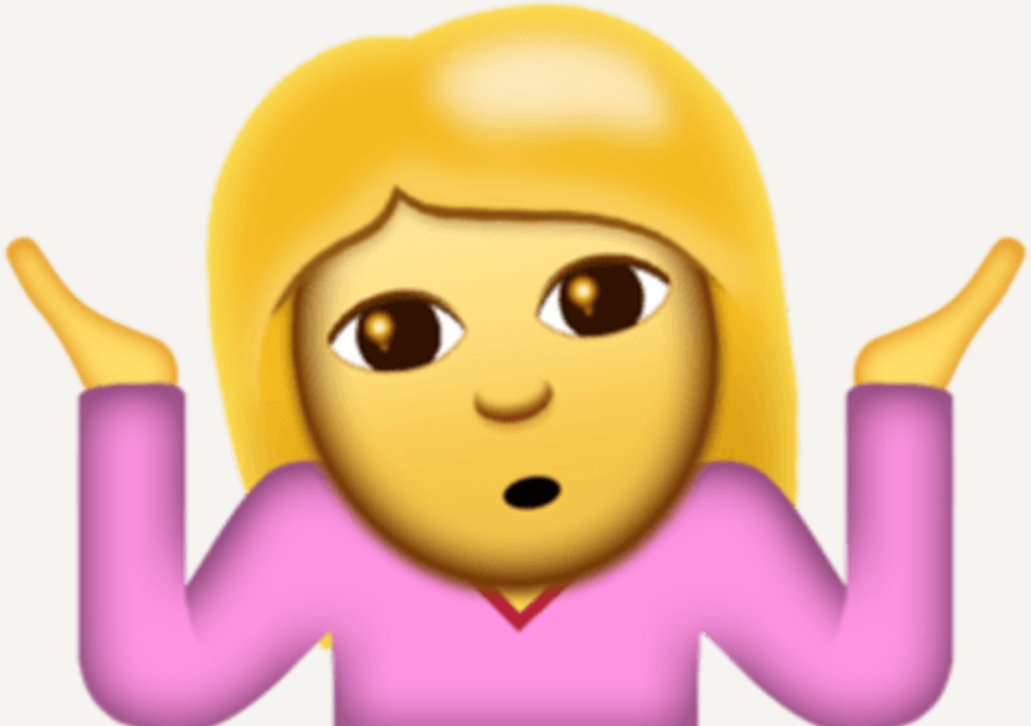
IaC とは

- インフラの状態を設定ファイルで書く
- ソフトウェア開発の手法を応用できる
  - レビュー
  - テスト
  - etc

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
      - containerPort: 80
```

Kubernetes Pod の YAML

YAML をテストする




HashiCorp

Policy as Code

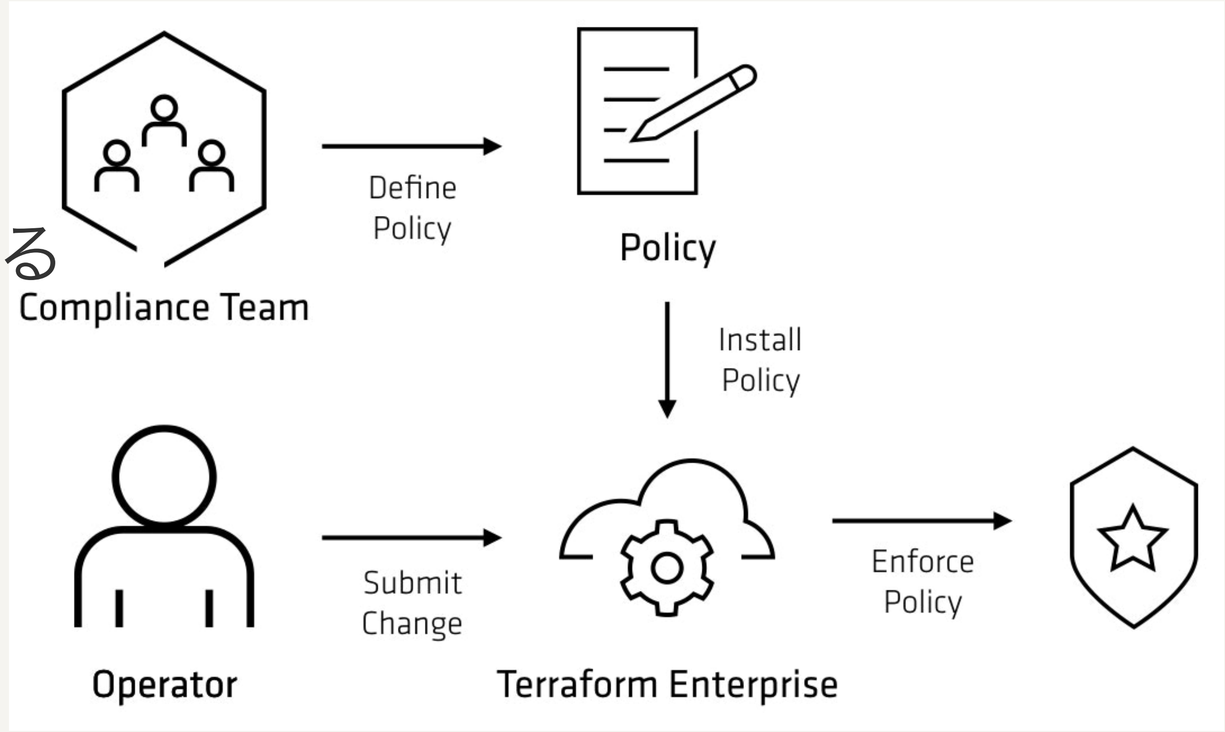
- HashiCorp が提唱した考え方
- 設定ファイルにおける “こうあるべき” をポリシーとして記す
  - 制約項目 (deploy region, etc)
  - レビュー項目 (like style guide)

Why Policy as Code? - HashiCorp Blog

Policy as Code

- HashiCorp Sentinel
  - HashiCorp 製品で使うことができる

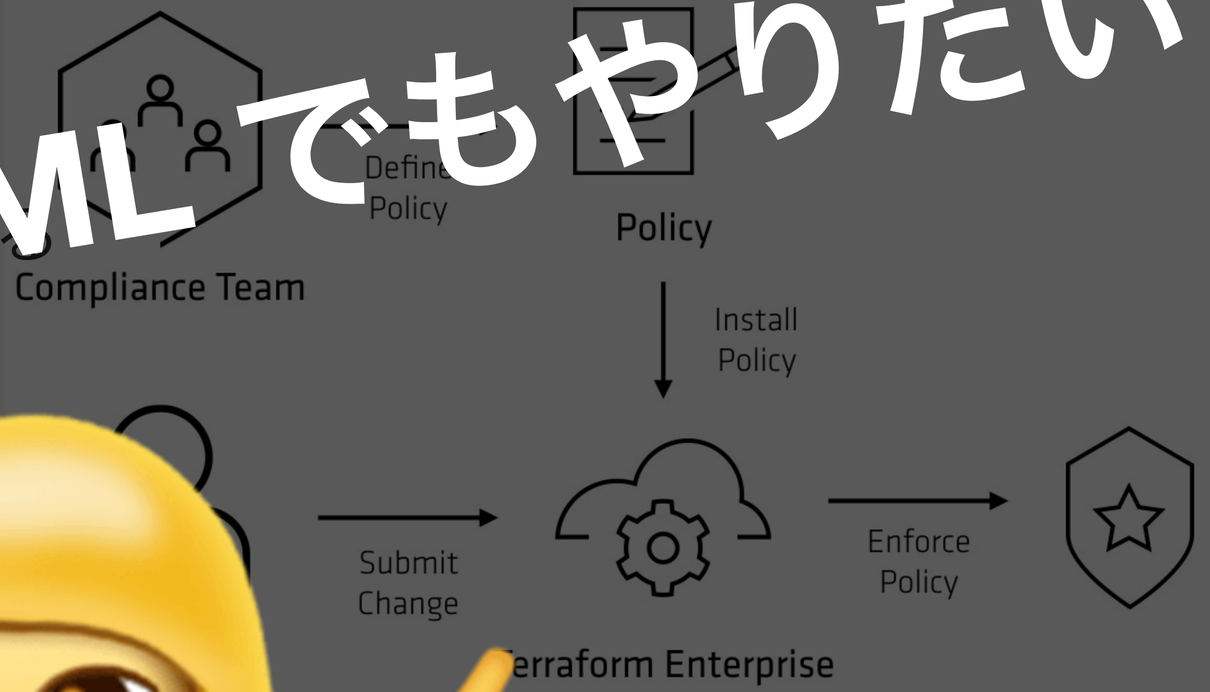
    ツール / 連携ができる
- 例えば Terraform の設定、
  - どこの Region にデプロイするか
  - Instance は最低何台確保されるか
  - などをポリシーとしてコード化できる
  - それをチェックできる



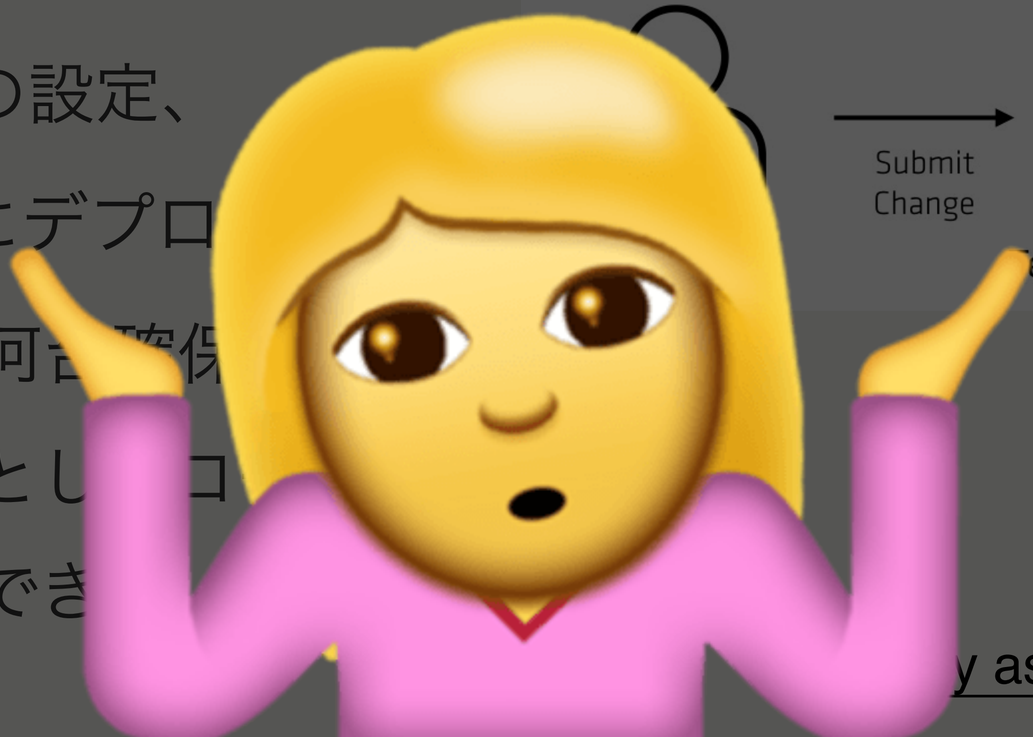
Policy as Code - Sentinel by HashiCorp

Policy as Code

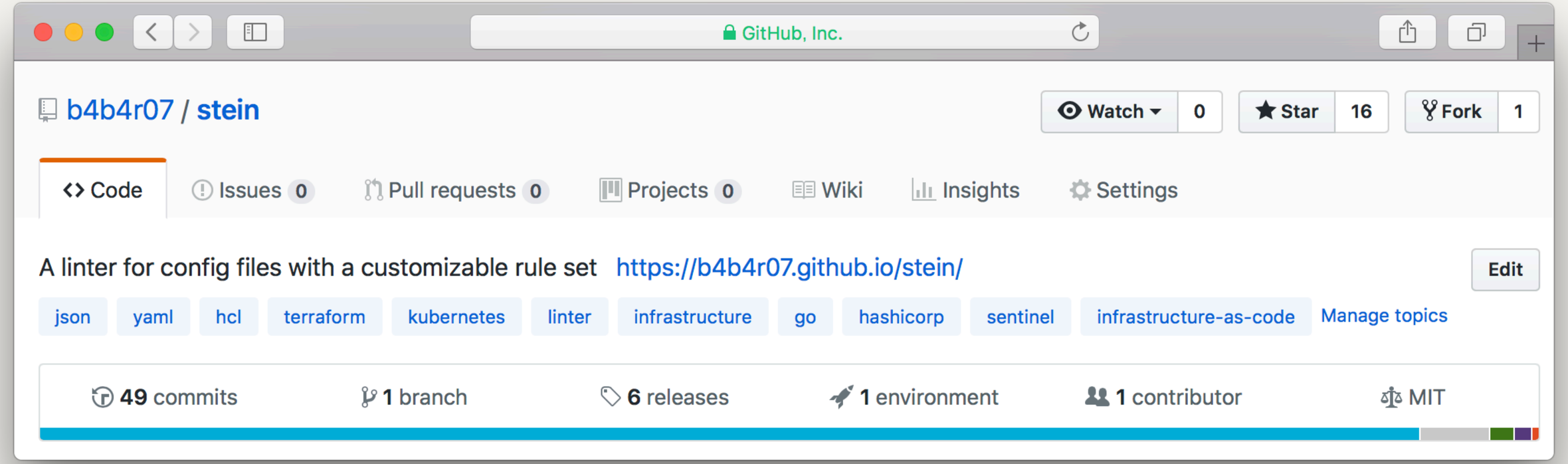
- HashiCorp Sentinel
  - HashiCorp 製品で使うことが多い
  - ～～と連携ができる
- 例えば Terraform の設定、
  - どこの Region にデプロ
  - Instance は最低何台確保
  - などをポリシーとしてコ
  - それをチェックでき



Kubernetes YAML でもやりたい



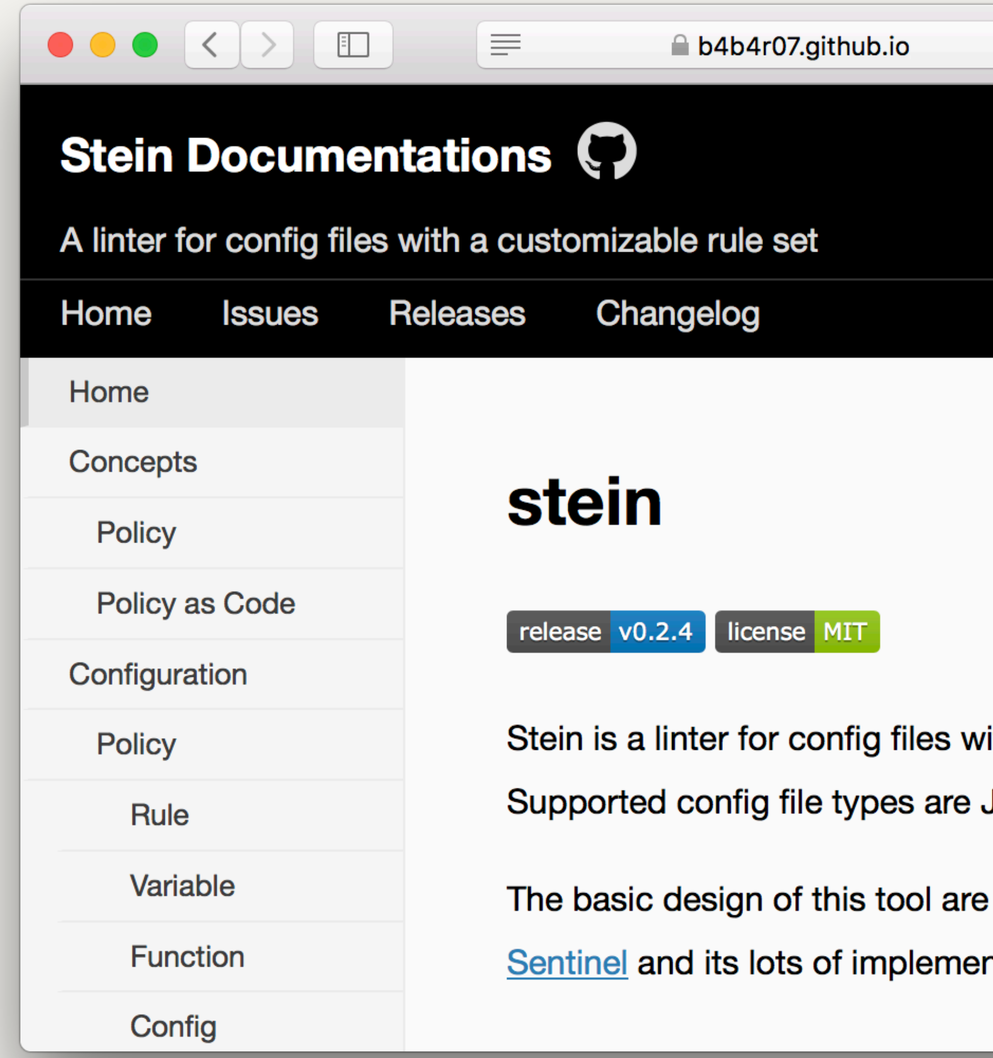
y as Code - Sentinel by HashiCorp


GitHub, Inc.
b4b4r07 / stein
Watch 0
Star 16
Fork 1
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
A linter for config files with a customizable rule set https://b4b4r07.github.io/stein/
Edit
json
yaml
hcl
terraform
kubernetes
linter
infrastructure
go
hashicorp
sentinel
infrastructure-as-code
Manage topics
49 commits
1 branch
6 releases
1 environment
1 contributor
MIT

Stein

Stein

- 設定ファイルのポリシーをコード化できる
  - JSON, YAML, HCL
- Policy as Code を実践する Linter
- Terraform のように HCL でルール作成できる
  - 豊富な Interpolations
  - ドキュメント



Stein Documentations

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: x-echo-jp-dev
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
      - containerPort: 80
```

例えば

省略できる

けどさせたくない

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: x-echo-jp-dev
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
      - containerPort: 80
```

```
rule "namespace_specification" {
  description = "Check namespace name is not empty”

  conditions = [
    "${jsonpath("metadata.namespace") != ""}",
  ]

  report {
    level   = "ERROR"
    message = "Namespace is not specified"
  }
}
```

ルールの定義

```
rule "namespace_specification" {
  description = "Check namespace name is not empty”

  conditions = [
    "${jsonpath("metadata.namespace") != ""}",
  ]

  report {
    level   = "ERROR"
    message = "Namespace is not specified"
  }
}
```

```
rule "namespace_specification" {
  description = "Check namespace name is not empty”

  conditions = [
    "${jsonpath("metadata.namespace") != ""}",
  ]

  report {
    level   = "ERROR"
    message = "Namespace is not specified"
  }
}
```

ルールが成功するか失敗するかの条件

```
rule "namespace_specification" {
  description = "Check namespace name is not empty”

  conditions = [
    "${jsonpath("metadata.namespace") != ""}",
  ]

  report {
    level   = "ERROR"
    message = "Namespace is not specified"
  }
}
```

ルールが失敗したらこのフォーマットに従って

エラーがレポートされる (終了コード1)

```
$ stein apply
x-echo-jp/development/Pod/test.yaml
  [ERROR]  rule.namespace_specification  Namespace is not specified

======================
7 error(s), 2 warn(s)
```

- **Stein を使うことで、Sentinel のように Policy as Code を実践できる**
  - Sentinel は HashiCorp 製品に、Stein は任意の設定ファイルに
- **制約項目の検証やレビュー観点の指摘を機械的にできる**
  - 「注意深く見なければいけない」「毎回指摘する」などは
    機械的にチェックしてポリシーをルール化するべき

Goで作った経緯

ブログに書いた

- hashicorp/hcl2 を使って独自 DSL を定義する | tellme.tokyo
- Kubernetes などの YAML を独自のルールをもとにテストする | tellme.tokyo

Thank you