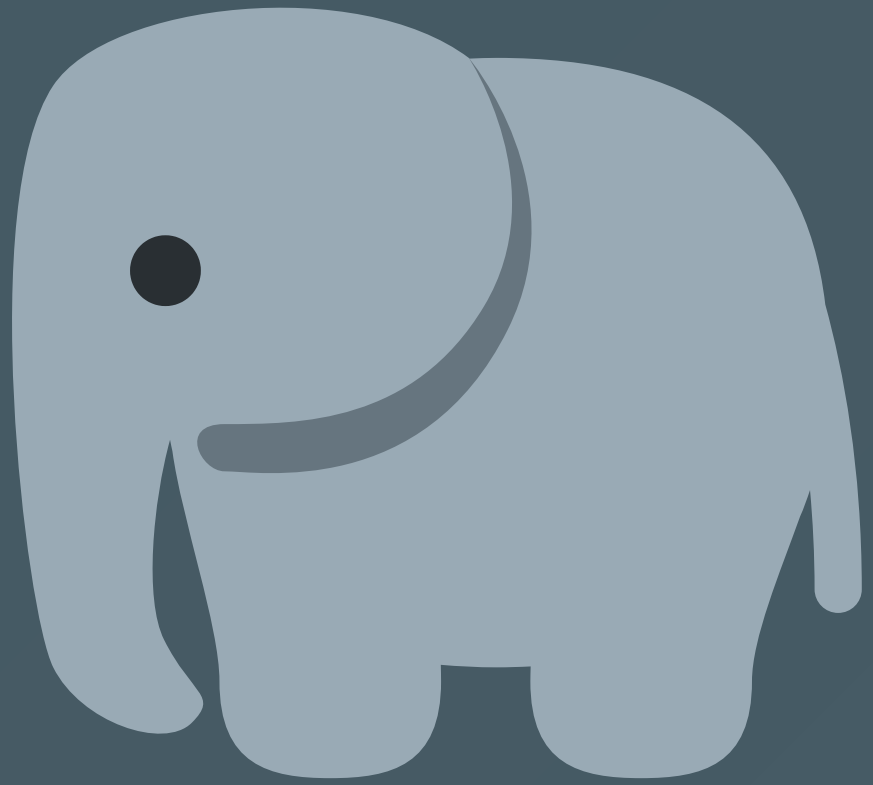
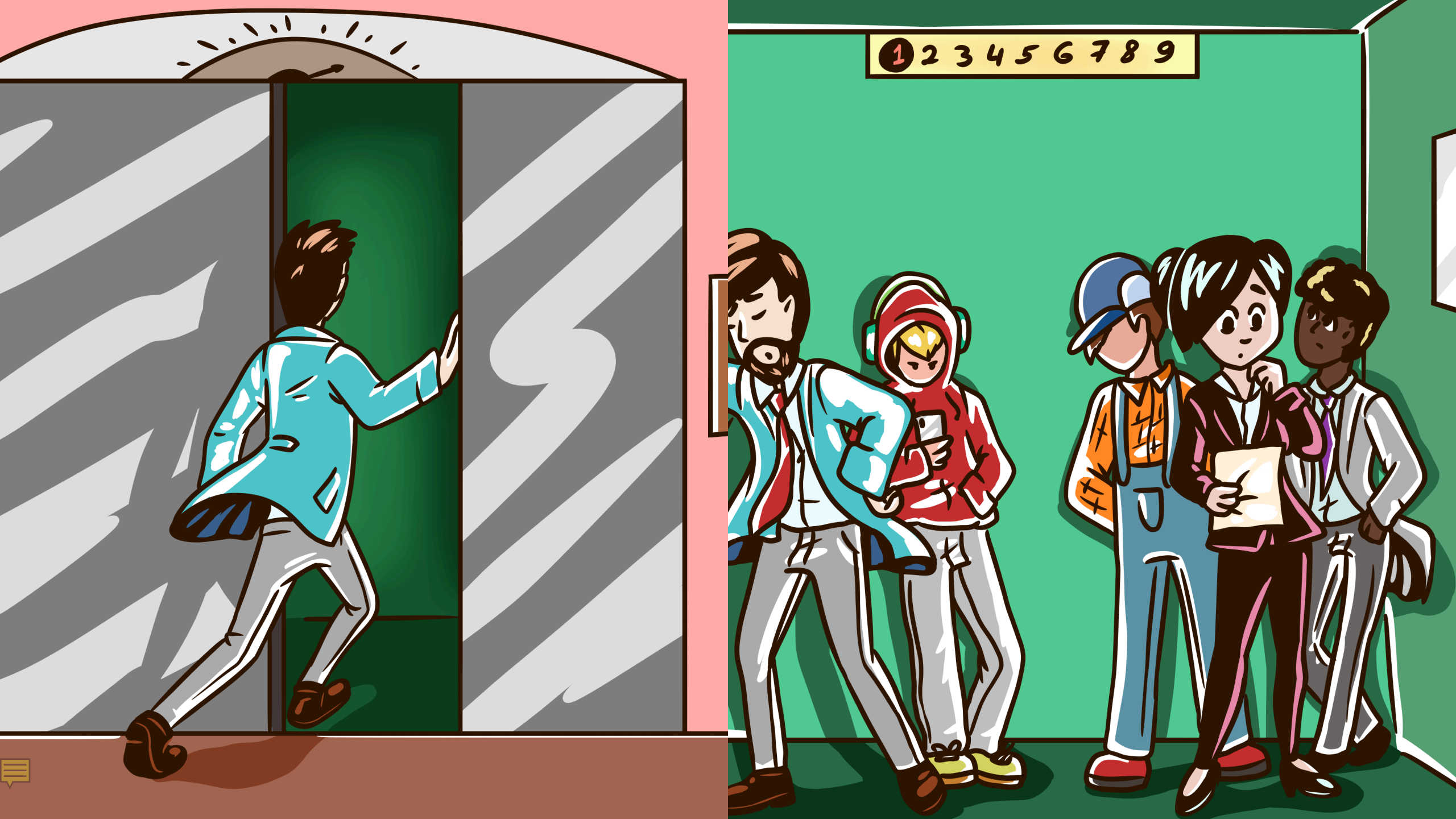
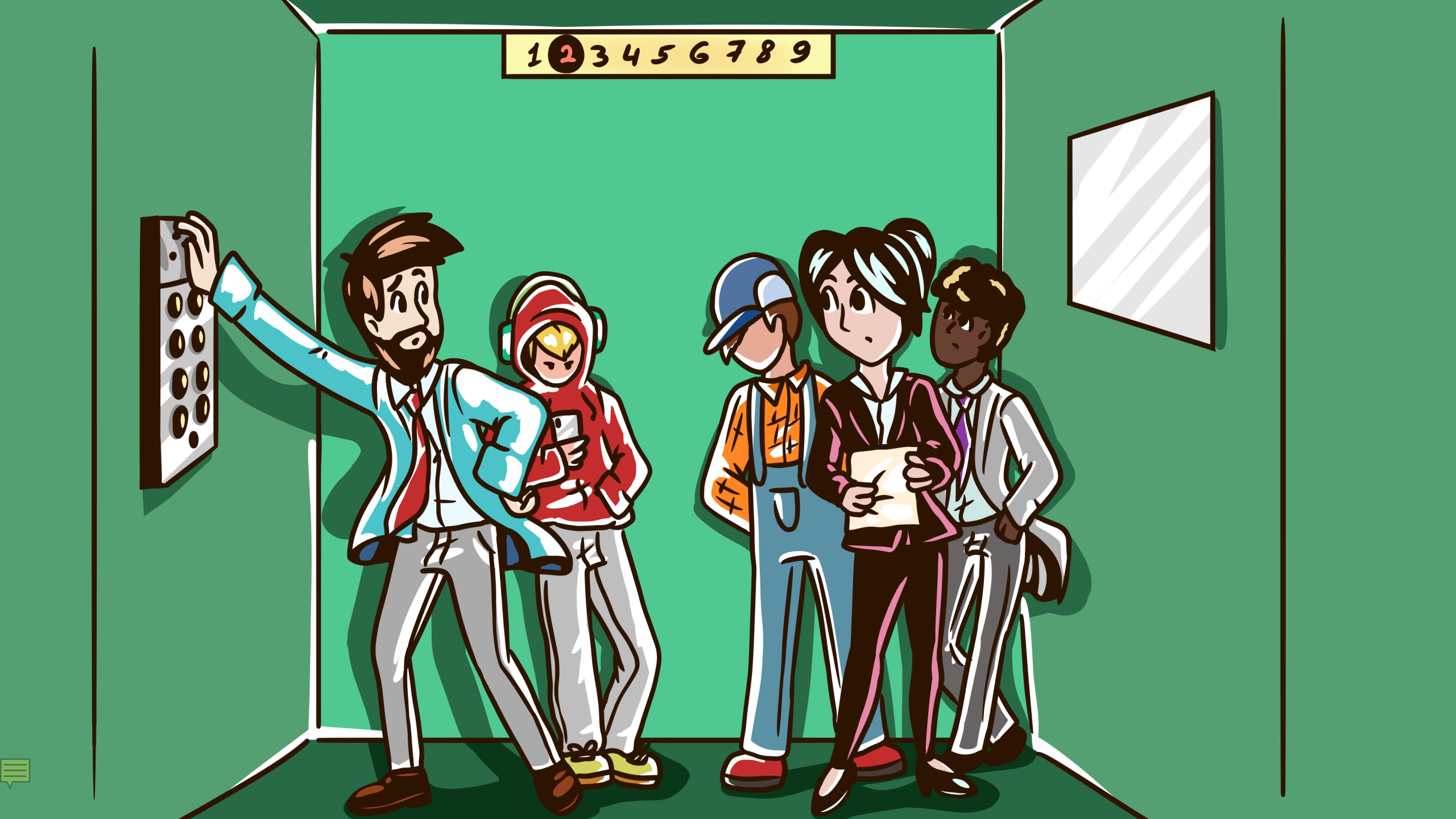
⚡ Policy as *[versioned]* Code

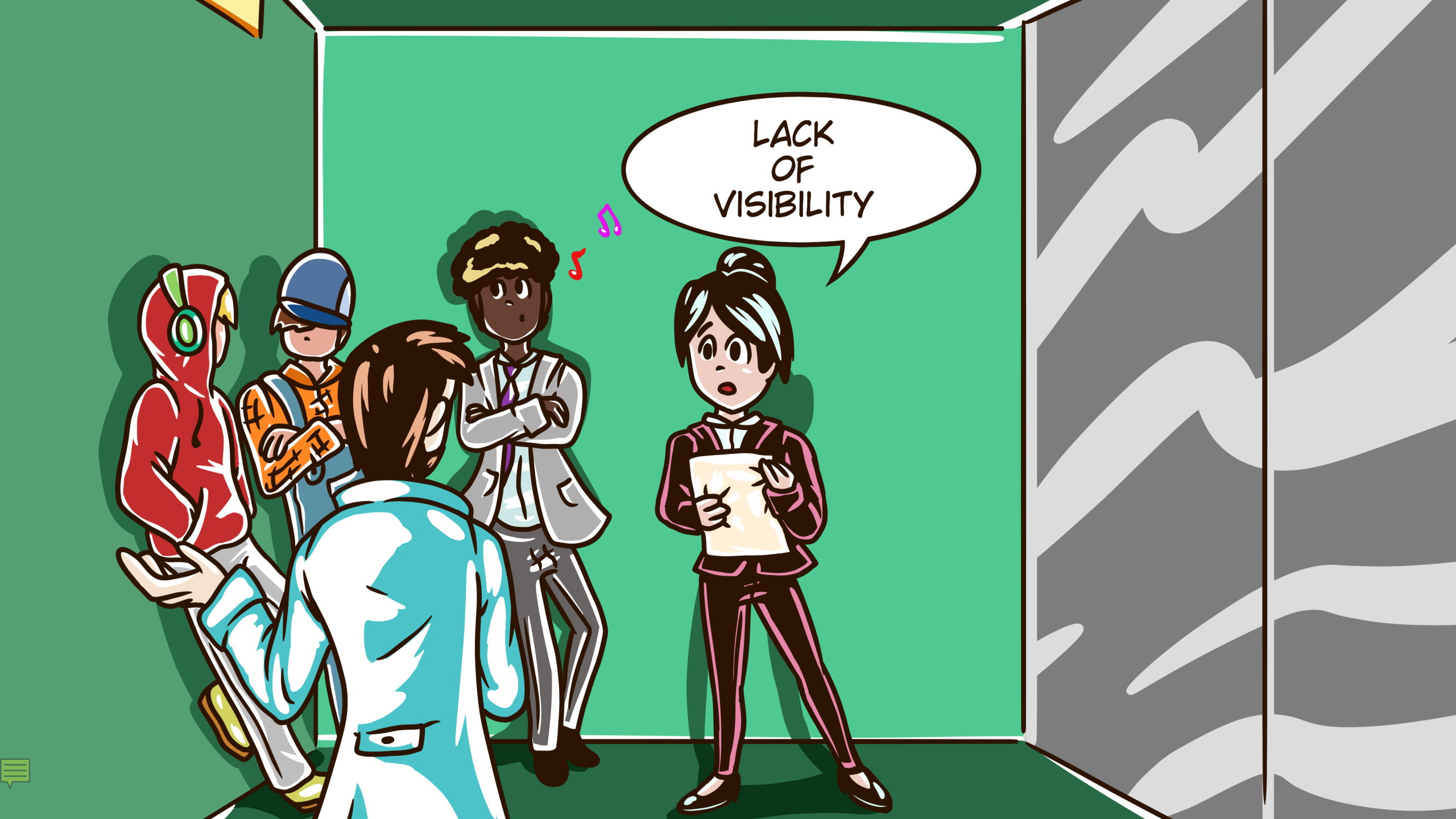
Chris Nesbitt-Smith

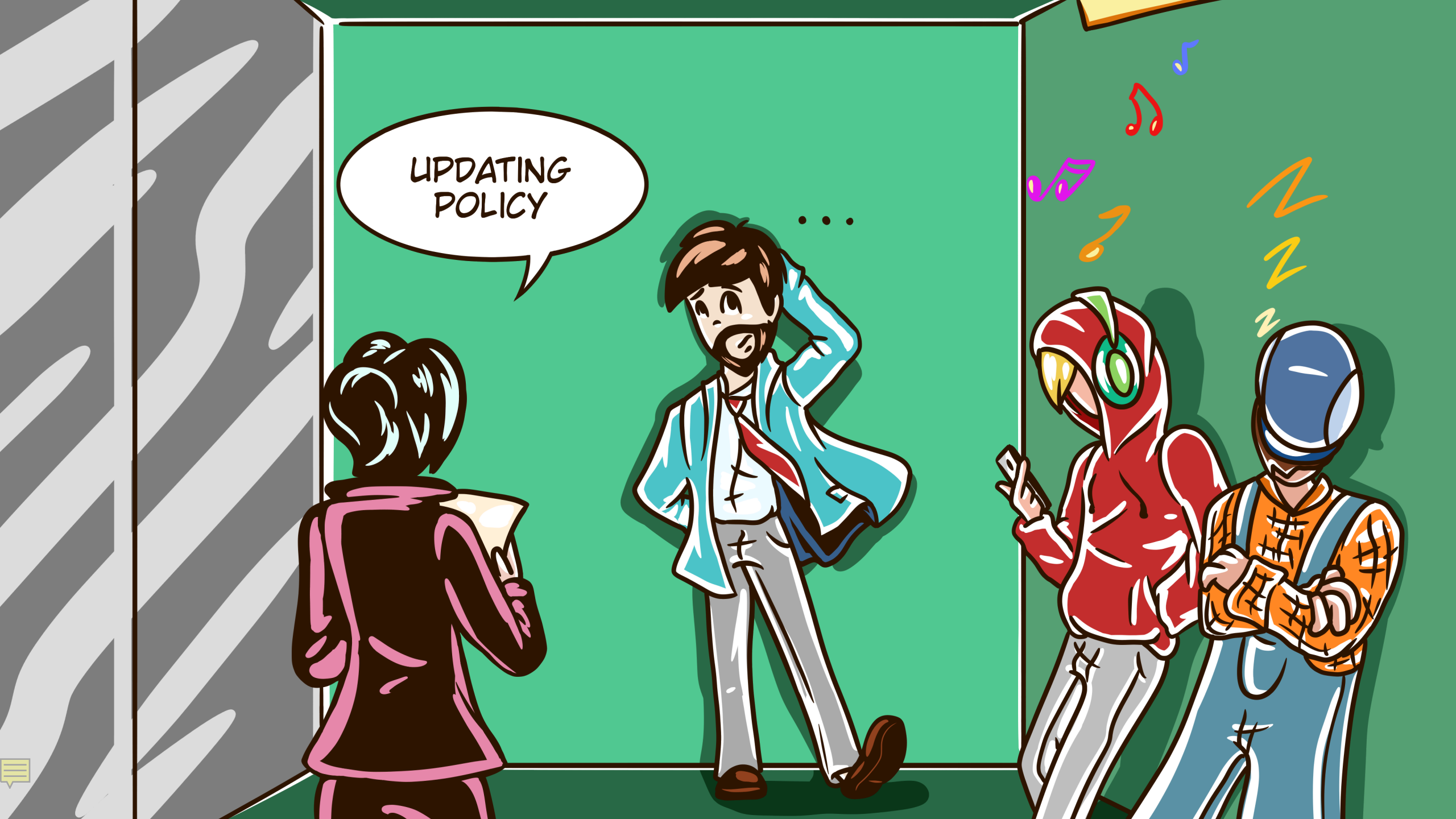
UK Gov | esynergy | Control Plane | LearnK8s | lots of open source



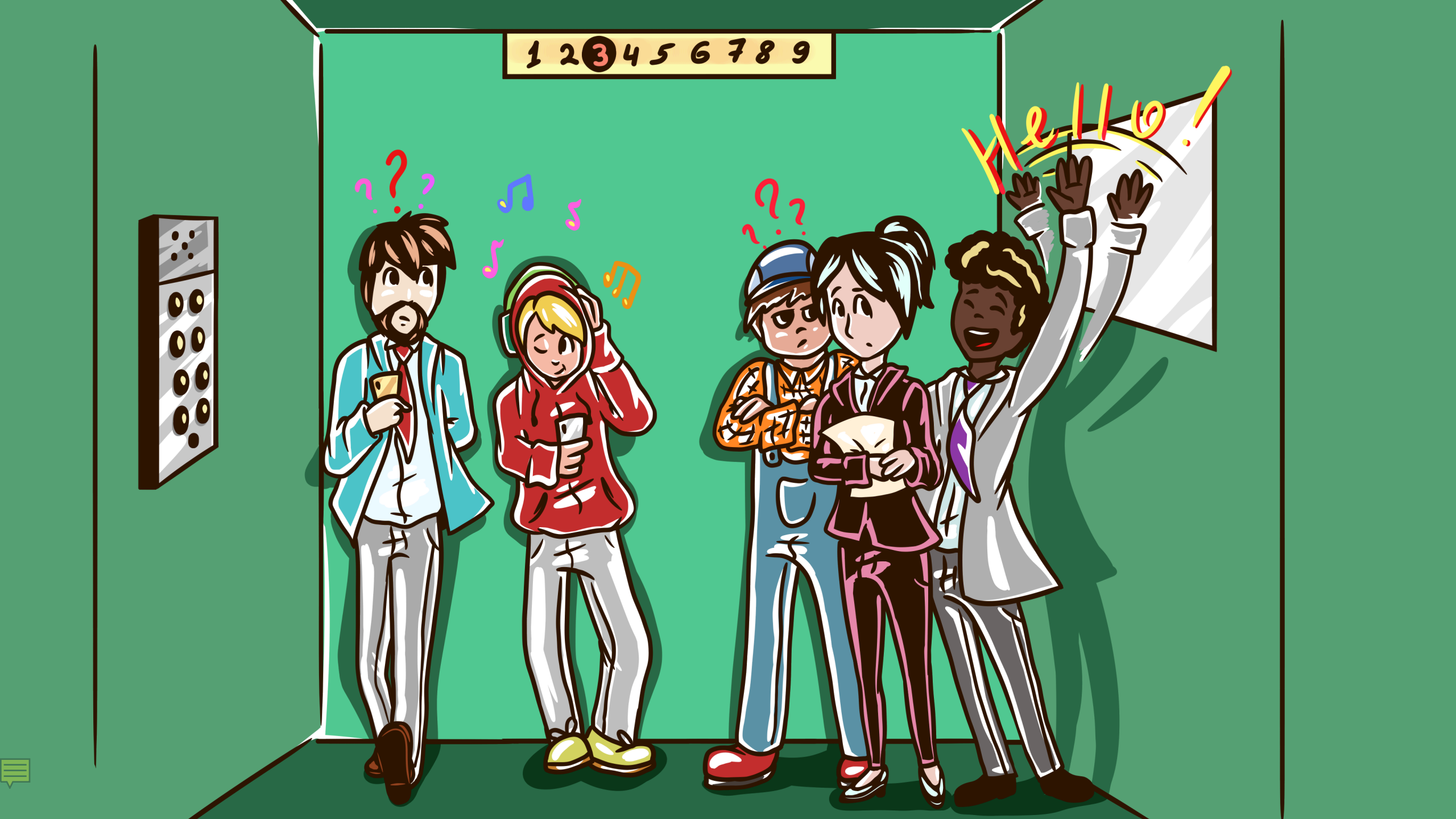

1 2 3 4 5 6 7 8 9


1 2 3 4 5 6 7 8 9


LACK
OF
VISIBILITY


UPDATING POLICY
. . .


1 2 3 4 5 6 7 8 9
LACK OF CONTROL

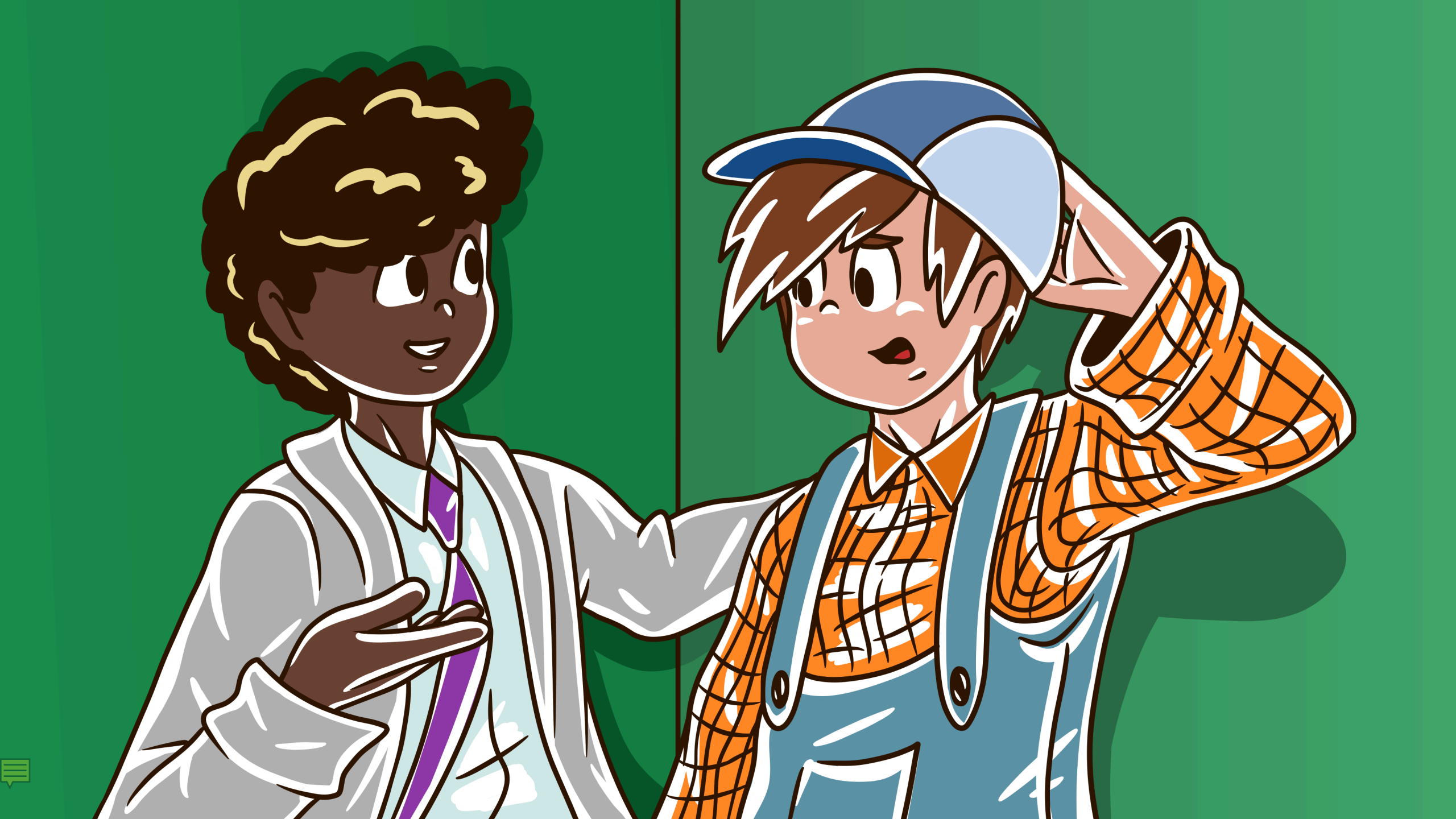

1 2 3 4 5 6 7 8 9
Hello!


MANAGING RISK


CLEANER


CODE


QUALITY



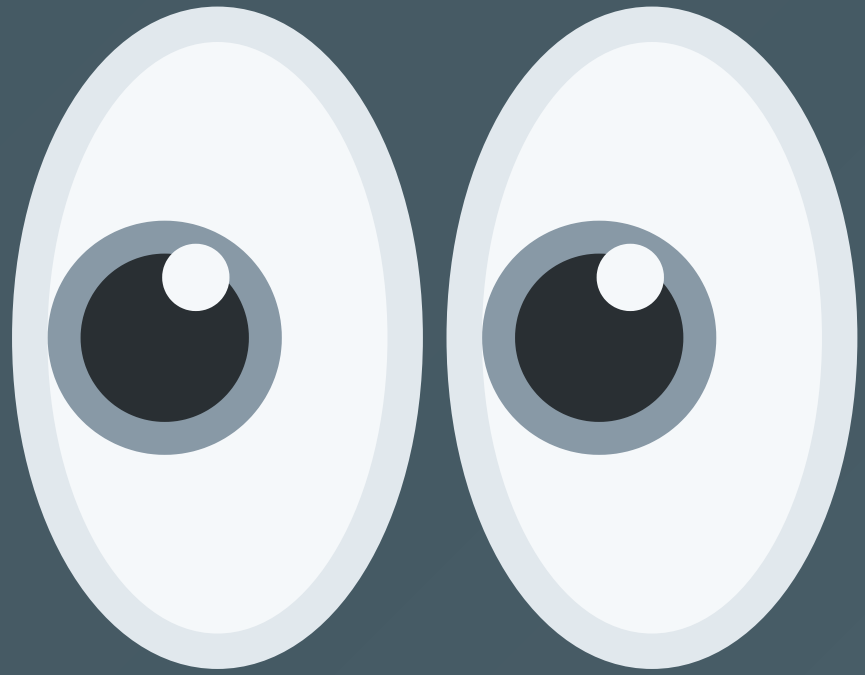

Ha
Ha
Ha
Ha
Ha
YEA

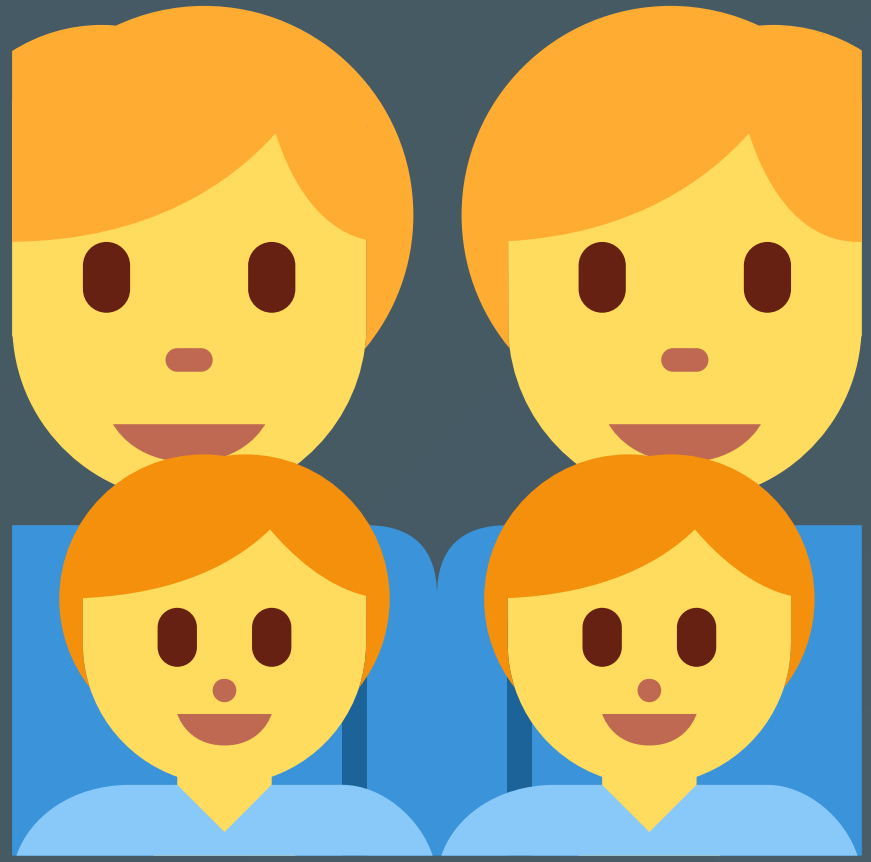
What if...

Update policy...

Update policy...

Daily!?





Chris Nesbitt-Smith

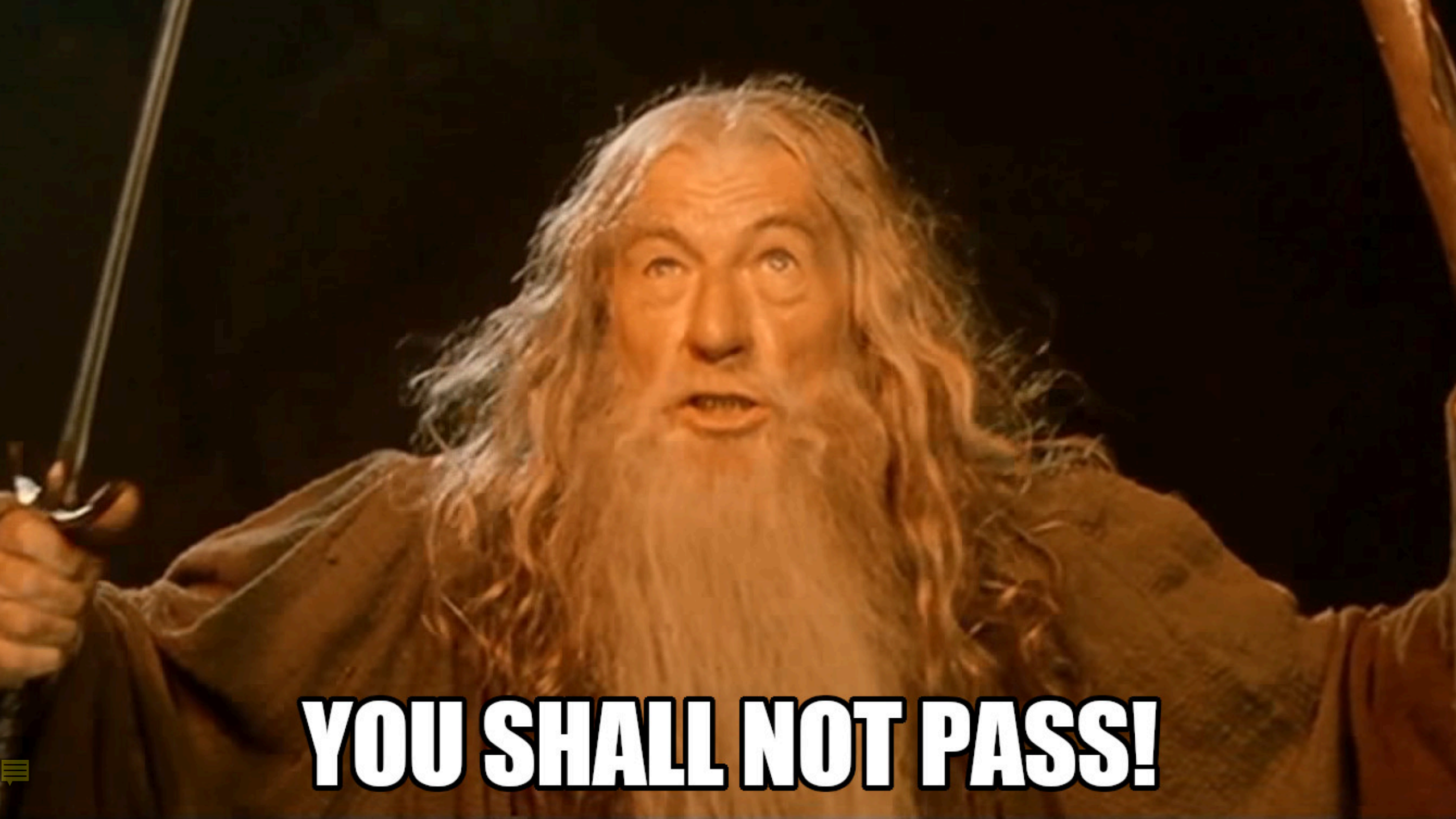
- Learnk8s & ControlPlane - Instructor+consultant
- esynergy - Digital Transformation Consultant
- Crown Prosecution Service (UK gov) - Consultant
- Opensource

policy

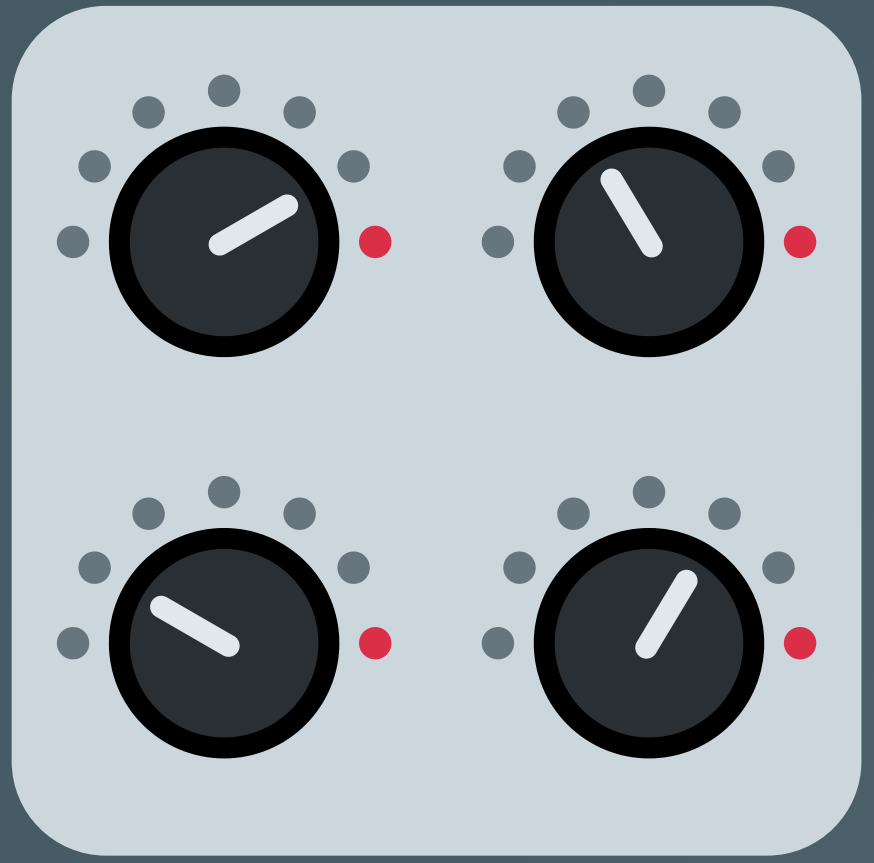
noun [C]

UK /ˈpɒl.ə.si/ US /ˈpɑː.lə.si/

a set of ideas or a plan of what to do in particular situations that has been agreed to officially by a group of people, a business organization, a government, or a political party

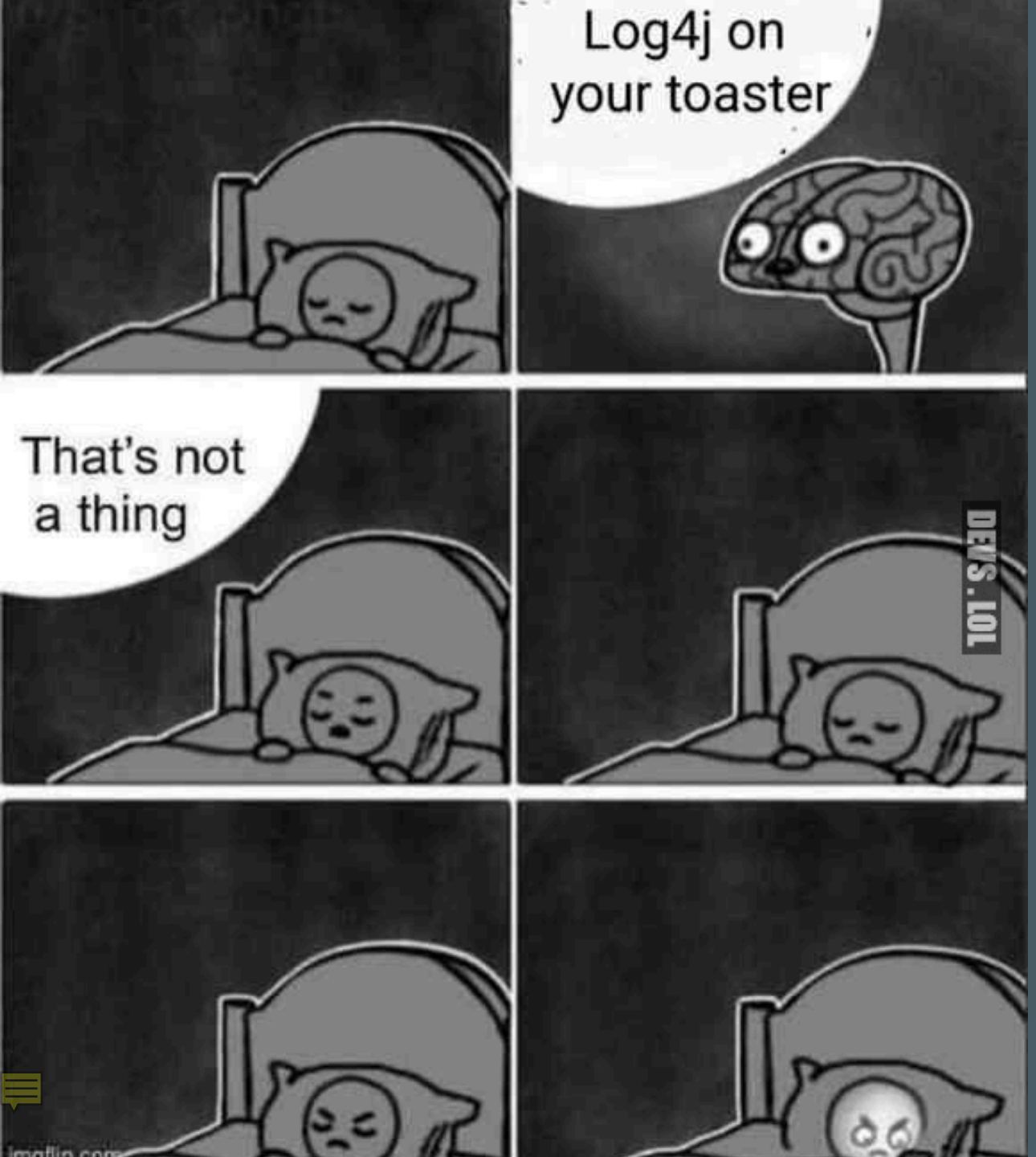

YOU SHALL NOT PASS!


7
7
7



Admission Control | Anchore | Apparmor | Azure Policy | Checkov | Istio | jspolicy | K-rail | Kopf | Kubewarden | Kyverno | Network Policy | OPA Gatekeeper | Opslevel | Polaris | Prisma Cloud | Qualys | Rego | Regula | Seccomp | SeLinux | Sentinel | ServiceControlPolicy | Sysdig | TiDB

But, we just provide `warn`ings not `error`s?



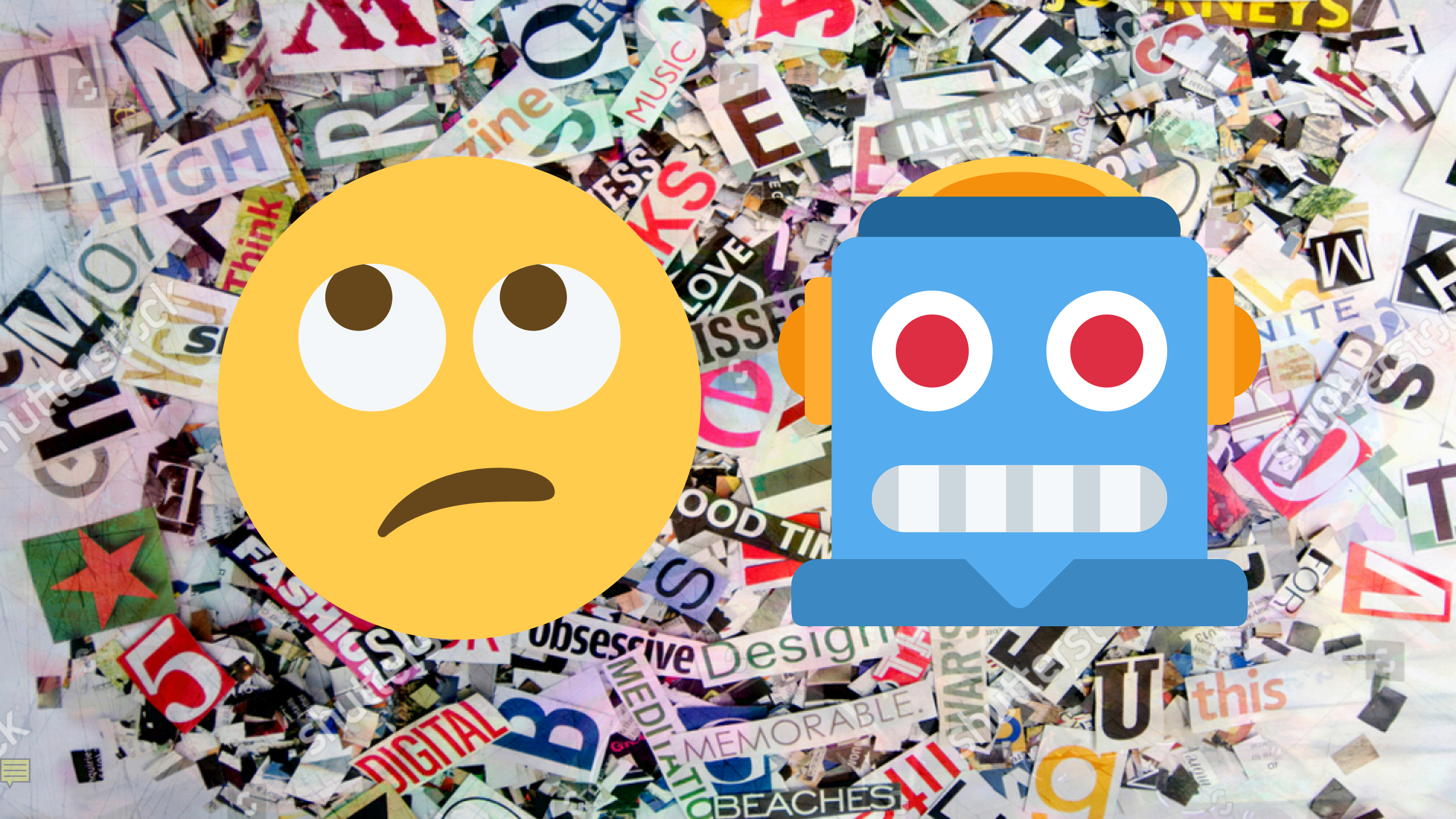
CVE-2021-44228

CVE-2021-45046

CVE-2021-45105

SCIENCE FICTION DAY

Here is where your presentation begins


Obsessive Design
MEMORABLE.
BEACHES
DIGITAL
this

+

(Terraform + Kubernetes)

Admission Control | Anchore |
Apparmor | Azure Policy | **Checkov** |
Istio | jspolicy | K-rail | Kopf |
Kubewarden | **Kyverno** | Network
Policy | OPA Gatekeeper | Opslevel |
Polaris | Prisma Cloud | Qualys | Rego |
Regula | Seccomp | SeLinux | Sentinel |
ServiceControlPolicy | Sysdig | TiDB

github.com/policy-as-versioned-code

Policy as Versioned Code demo

United Kingdom chris@cns.me.uk

Overview Repositories 10 Projects Packages People

policy-as-versioned-code / policy

Public

Company Policy

Releases 4

v2.1.1 Latest
18 hours ago

+ 3 releases

Policy as [versioned] code

This repo contains the company policy that has been codified into kyverno and checkov policies.

v1.0.0 policy

```yaml
# checkov terraform
metadata:
  name: >-
    Check that all resources are tagged with
    the key - department"
  id: "CUSTOM_AWS_1"
  category: "CONVENTION"
scope:
  provider: aws
definition:
  and:
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "exists"
```

```yaml
# kyverno kubernetes
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-department-label
  annotations:
    policies.kyverno.io/title: Require Department Label
    policies.kyverno.io/category: Example Org Policy
    policies.kyverno.io/description: >-
      It is important we know the department that resources
      belong to, so you need to define a 'mycompany.com/department'
      label on all your resources.
    pod-policies.kyverno.io/autogen-controllers: none
spec:
  validationFailureAction: enforce
  background: false
  rules:
    - name: require-department-label
      validate:
        message: >-
          The label `mycompany.com/department` is
          required.
        pattern:
          metadata:
            labels:
              "mycompany.com/department": "?*"
```

v1.0.0 policy tests

```
// fail0.tf
resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
}
---
// pass0.tf
resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  tags = {
    mycompany.com.department = "finance"
  }
}
```

```
# fail0.yaml
apiVersion: v1
kind: Pod
metadata:
  name: require-department-label-fail0
spec: ...
---
# pass0.yaml
apiVersion: v1
kind: Pod
metadata:
  name: require-department-label-pass0
  labels:
    mycompany.com/department: finance
spec: ...
```

V1.0.0 Initial Policy Release

 **chrisns** released this 20 hours ago · 8 commits to main since this release 1.0.0 9b360cb

What's Changed

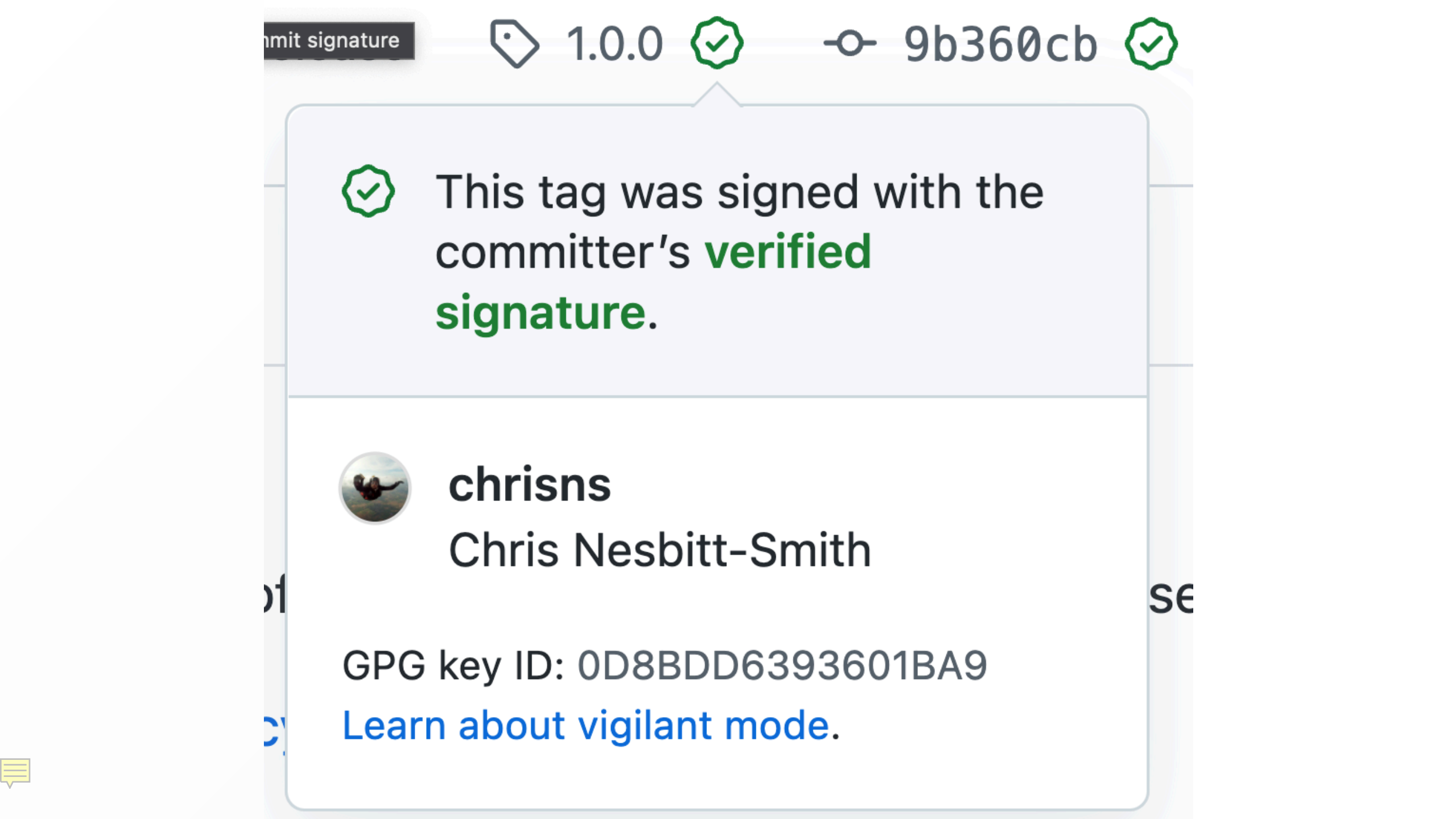
Initial release of policy

All resources require a `label` (Kubernetes) or `tag` (terraform) of `mycompany.com/department` to be set

Full Changelog: https://github.com/policy-as-versioned-code/policy/commits/v1.0.0

▾ Assets 2

- Source code (zip)
- Source code (tar.gz)


mit signature
1.0.0
9b360cb
This tag was signed with the committer's verified signature.
chrisns
Chris Nesbitt-Smith
GPG key ID: 0D8BDD6393601BA9
Learn about vigilant mode.

v2.0.0 policy

```yaml
# checkov terraform
metadata:
  name: >-
    Check that all resources are tagged with the key - department"

  id: "CUSTOM_AWS_1"
  category: "CONVENTION"
scope:
  provider: aws
definition:
  or:
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: hr
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: acounts
```

```yaml
# kyverno kubernetes
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-department-label
  annotations:
    policies.kyverno.io/title: Require Department Label
    policies.kyverno.io/category: Example Org Policy
    policies.kyverno.io/description: >-
      It is important we know the department that resources
      belong to, so you need to define a 'mycompany.com/department'
      label on all your resources.
    pod-policies.kyverno.io/autogen-controllers: none
spec:
  validationFailureAction: enforce
  background: false
  rules:
    - name: require-department-label
      validate:
        message: >-
          The label `mycompany.com/department` is required to be one
          of [acounts|hr]"
        pattern:
          metadata:
            labels:
              "mycompany.com/department": "acounts|hr"
```

v2.1.0 policy

```yaml
# checkov terraform
metadata:
  name: >-
    Check that all resources are tagged with the key - department"

  id: "CUSTOM_AWS_1"
  category: "CONVENTION"
scope:
  provider: aws
definition:
  or:
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: hr
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: accounts
```

```yaml
# kyverno kubernetes
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-department-label
  annotations:
    policies.kyverno.io/title: Require Department Label
    policies.kyverno.io/category: Example Org Policy
    policies.kyverno.io/description: >-
      It is important we know the department that resources
      belong to, so you need to define a 'mycompany.com/department'
      label on all your resources.
    pod-policies.kyverno.io/autogen-controllers: none
spec:
  validationFailureAction: enforce
  background: false
  rules:
    - name: require-department-label
      validate:
        message: >-
          The label `mycompany.com/department` is required to be one
          of [accounts|hr]"
        pattern:
          metadata:
            labels:
              "mycompany.com/department": "accounts|hr"
```

v2.1.1 policy

```yaml
# checkov terraform
metadata:
  name: >-
    Check that all resources are tagged with
    the key - department"
  id: "CUSTOM_AWS_1"
  category: "CONVENTION"
scope:
  provider: aws
definition:
  or:
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: hr
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: accounts
    - cond_type: "attribute"
      resource_types: "all"
      attribute: "tags.mycompany.com.department"
      operator: "equals"
      value: tech
```

```yaml
# kyverno kubernetes
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-department-label
  annotations:
    policies.kyverno.io/title: Require Department Label
    policies.kyverno.io/category: Example Org Policy
    policies.kyverno.io/description: >-
      It is important we know the department that resources
      belong to, so you need to define a 'mycompany.com/department'
      label on all your resources.
    pod-policies.kyverno.io/autogen-controllers: none
spec:
  validationFailureAction: enforce
  background: false
  rules:
    - name: require-department-label
      validate:
        message: >-
          The label `mycompany.com/department` is required to be one
          of [tech|accounts|hr]"
        pattern:
          metadata:
            labels:
              "mycompany.com/department": "tech|accounts|hr"
```

app1 (k8s) | infra1 (tf)

1.0.0	2.0.0	2.1.0	2.1.1
✅	❌	❌	❌

```
$schema: "https://docs.renovatebot.com/renovate-schema.json",
labels: ["policy"],
regexManagers: [{
  fileMatch: ["kustomization.yaml"],
  matchStrings: ['mycompany.com/policy-version: "(?<currentValue>.*)"\\s+'],
  datasourceTemplate: "github-tags",
  depNameTemplate: "policy",
  packageNameTemplate: "policy-as-versioned-code/policy",
  versioningTemplate: "semver",
},{
  fileMatch: [".*tf$"],
  matchStrings: [
    '#\\s*renovate:\\s*policy?\\s*default = "(?<currentValue>.*)"\\s',
  ],
  datasourceTemplate: "github-tags",
  depNameTemplate: "policy",
  lookupNameTemplate: "policy-as-versioned-code/policy",
  versioningTemplate: "semver",
}],
```

Update dependency policy to v2 #10

Edit | Code

Open **renovate** wants to merge 1 commit into `main` from `renovate/policy-2.x`

Conversation 0 | Commits 1 | Checks 2 | Files changed 1 | +1 −1

renovate bot commented 29 minutes ago

This PR contains the following updates:

Package	Update	Change
policy	major	`1.0.0` -> `2.1.1`

Release Notes

▼ policy-as-versioned-code/policy

`v2.1.1`

Compare Source

What's Changed

- add sales department as available option of departments by @chrisns in https://github.com/policy-as-versioned-code/policy/pull/7

Full Changelog: policy-as-versioned-code/policy@`v2.1.0...v2.1.1`

`v2.1.1`

Compare Source

`v2.1.0`

Reviewers

Suggestions

chrisns Request

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

policy

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications Customize

Subscribe

You're not receiving notifications from this thread.

policy

failed 29 minutes ago in 43s

Search logs

```
"/home/runner/work/_temp/_runner_file_commands":"/github/file_commands" -v
"/home/runner/work/app1/app1":"/github/workspace" ghcr.io/policy-as-versioned-code/policy-
checker:latest
Found kustomization.yaml
Checking policy version...
Policy version: 2.1.1
Fetching Policy...
Policy fetched.
Running policy checker...

Applying 2 policies to 1 resource...
(Total number of result count may vary as the policy is mutated by Kyverno. To check the mutated
policy please try with log level 5)

policy require-known-department-label -> resource default/Deployment/app1 failed:
1. require-known-department-label: validation error: The label `mycompany.com/department` is
required to be one of [tech|accounts|servicedesk|hr|sales]. Rule require-known-department-label
failed at path /metadata/labels/mycompany.com/department/

pass: 1, fail: 1, warn: 0, error: 0, skip: 0
```

4 Open  3 Closed

policy-as-versioned-code/infra1 **Update dependency policy to v2**  policy

#8 opened 7 minutes ago by renovate bot

policy-as-versioned-code/infra2 **Update dependency policy to v2.1.1**  policy

#8 opened 7 minutes ago by renovate bot

policy-as-versioned-code/app2 **Update dependency policy to v2.1.1**  policy

#6 opened 23 minutes ago by renovate bot

policy-as-versioned-code/app1 **Update dependency policy to v2**  policy

#10 opened 25 minutes ago by renovate bot

app2 (k8s) | infra2 (tf)

1.0.0	2.0.0	2.1.0	2.1.1
-	✅	☑️	☑️

All checks have passed

2 successful checks

This branch has no conflicts with the base branch

Merging can be performed automatically.

Squash and merge

You can also open this in GitHub Desktop

app3 (k8s) | infra3 (tf)

1.0.0	2.0.0	2.1.0	2.1.1
-	-	-	✅

main | 1 branch | 0 tags | Go to file

renovate-bot and **renovate[bot]** Update ghcr.io/kyverno/kyverno-cli digest t... ✓ 13f8349 5 hours ago 21 commits

.github	Update docker/setup-buildx-action action to v2	17 hours ago
CODE_OF_CONDUCT.md	Update templated files (#1)	3 days ago
Dockerfile	Update ghcr.io/kyverno/kyverno-cli digest to 113485b	5 hours ago
README.md	added readme	8 hours ago
SECURITY.md	Update templated files (#1)	3 days ago
requirements.txt	Update dependency checkov to v2.0.1140	yesterday
run.sh	remove the v again	17 hours ago

README.md

Policy checker

This is a tool that can be used locally and in CI by tooling to determine if the repository is compliant with policy.

The version of policy is determined by:

- Kubernetes: reads the `kustomization.yaml` file and retrieves the `commonLabels['mycompany.com/policy-version']`
- Terraform: reads the variable default_value of `mycompany.com/policy-version`

If theres any `.tf` files it'll check terraform and check Kubernetes if theres a `kustomization.yaml`.

About

Tool that can be used locally and in CI to check for compliance with policy.

Readme
Code of conduct
0 stars
0 watching
0 forks

Packages 1

policy-checker

Languages

Shell 72.4% Dockerfile 27.6%

```
$ docker run --rm -ti \
  -v $(pwd):/apps \
  ghcr.io/policy-as-versioned-code/policy-checker

Found kustomization.yaml
Checking policy version...
Policy version: 1.0.0
Fetching Policy...
Policy fetched.
Running policy checker...

Applying 1 policy to 1 resource...
(Total number of result count may vary as the
policy is mutated by Kyverno. To check the
mutated policy please try with log level 5)

pass: 1, fail: 0, warn: 0, error: 0, skip: 0
```

terraform

kubernetes

many-to-many

main 1 branch 0 tags

Go to file Code

the-repository-manager[bot] and chrisns Update templated files (#1) ... e72a36f 7 hours ago 3 commits

.github	Update templated files (#1)	7 hours ago
CODE_OF_CONDUCT.md	Update templated files (#1)	7 hours ago
README.md	E2e (#2)	7 hours ago
SECURITY.md	Update templated files (#1)	7 hours ago

README.md

All versions of policy co-existing on a single Kubernetes Cluster

Demo

```
# Create a cluster
$ kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.23.4) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹
```

About

All versions of policy co-existing on a single Kubernetes Cluster

Readme

Code of conduct

0 stars

0 watching

0 forks

main | 1 branch | 0 tags | Go to file | Code

chrisns cluster2 — d346e1a 4 minutes ago — 4 commits

.github	cluster2	4 minutes ago
CODE_OF_CONDUCT.md	Update templated files (#1)	7 hours ago
README.md	cluster2	4 minutes ago
SECURITY.md	Update templated files (#1)	7 hours ago

README.md

>=2.0.0 versions of policy co-existing on a single Kubernetes Cluster

Demo

```
# Create a cluster
$ kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.23.4) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹
```

policy-as-versioned-code / **cluster1** Public

Code | Pull requests | **Actions** | Security | Insights | Settings

Update templated files (#1)

CI #7

deploy

succeeded 7 hours ago in 2m 5s

- Set up job — 2s
- Run actions/checkout@2541b1294d2704b0964813337f33b291d3f8596b — 1s
- Run container-tools/kind-action@fdfd7e7ffb39e532f9de844dba8f1a29ae7e0f75 — 1m 14s
- Install kyverno — 4s
- Wait for kyverno to be installed — 20s
- **Apply Policy** — 6s

```
1  ▶ Run kubectl apply -k "github.com/policy-as-versioned-code/policy/kubernetes/kyverno?
   ref=1.0.0"
9  clusterpolicy.kyverno.io/require-department-label-1.0.0 created
10 clusterpolicy.kyverno.io/require-department-label-2.0.0 created
11 clusterpolicy.kyverno.io/require-known-department-label-2.0.0 created
12 clusterpolicy.kyverno.io/require-department-label-2.1.0 created
13 clusterpolicy.kyverno.io/require-known-department-label-2.1.0 created
14 clusterpolicy.kyverno.io/require-department-label-2.1.1 created
15 clusterpolicy.kyverno.io/require-known-department-label-2.1.1 created
```

- Wait for policy to be available — 6s
- **Deploy apps** — 5s

```
1  ▶ Run kubectl apply -k github.com/policy-as-versioned-code/app1
8  deployment.apps/app1 created
9  deployment.apps/app2 created
10 deployment.apps/app3 created
```

- Check all apps are deployed — 3s
- Post Run container-tools/kind-action@fdfd7e7ffb39e532f9de844dba8f1a29ae7e0f75 — 1s
- Post Run actions/checkout@2541b1294d2704b0964813337f33b291d3f8596b — 0s

policy-as-versioned-code / **cluster2** Public

Code | Issues | Pull requests | **Actions** | Security | Insights | ...

cluster2

CI #1

deploy

succeeded 6 minutes ago in 1m 58s

- Set up job — 1s
- Run actions/checkout@2541b1294d2704b0964813337f33b291d3f8596b — 0s
- Run container-tools/kind-action@fdfd7e7ffb39e532f9de844dba8f1a29ae7e0f75 — 1m 12s
- Install kyverno — 3s
- Wait for kyverno to be installed — 19s
- **Apply Policy** — 4s

```
1  ▶ Run kubectl apply -k "github.com/policy-as-versioned-code/policy/kubernetes/kyverno?
   ref=2.0.0"
8  clusterpolicy.kyverno.io/require-department-label-2.0.0 created
9  clusterpolicy.kyverno.io/require-known-department-label-2.0.0 created
10 clusterpolicy.kyverno.io/require-department-label-2.1.0 created
11 clusterpolicy.kyverno.io/require-known-department-label-2.1.0 created
12 clusterpolicy.kyverno.io/require-department-label-2.1.1 created
13 clusterpolicy.kyverno.io/require-known-department-label-2.1.1 created
```

- Wait for policy to be available — 4s
- **Deploy apps** — 7s

```
1  ▶ Run kubectl apply -k github.com/policy-as-versioned-code/app2
7  deployment.apps/app2 created
8  deployment.apps/app3 created
```

- Check all apps are deployed — 2s
- Post Run container-tools/kind-action@fdfd7e7ffb39e532f9de844dba8f1a29ae7e0f75 — 1s
- Post Run actions/checkout@2541b1294d2704b0964813337f33b291d3f8596b — 0s
- Complete job — 0s

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

Purposeless policy is potentially practically pointless policy.

🙏 Thanks 🙏

- cns.me
- talks.cns.me
- github.com/chrisns
- github.com/policy-as-versioned-code
- learnk8s.io

Chris Nesbitt-Smith

cns.me

github.com/policy-as-versioned-code

Chris Nesbitt-Smith