FULL STACKS

CONTAINER days LONDON

Security as Code

From Rules to Reality

Jan Hoepfner

@jeypie

Senior DevOps Engineer

Co-Founder @ Burn4IT 🎙️

🍳 🎹 🧩 | 🏃 | 🚵 🍖 🍻

SUSE | VMware | ArgoCD | GitLab | Kubernetes

Daniel Drack

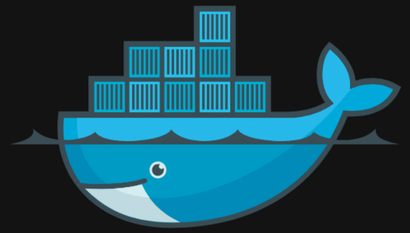
@DrackThor

Senior DevOps Engineer

Host @ Cloud Native Days Austria
Founder @ Cloud Native Austria
Organizer @ Cloud Native Chapter Graz

BSc | MA | MBA
CNCF Ambassador | Kubestronaut
SUSE | Exoscale | Snyk | GitLab | Scrum





* Features =

+ X =

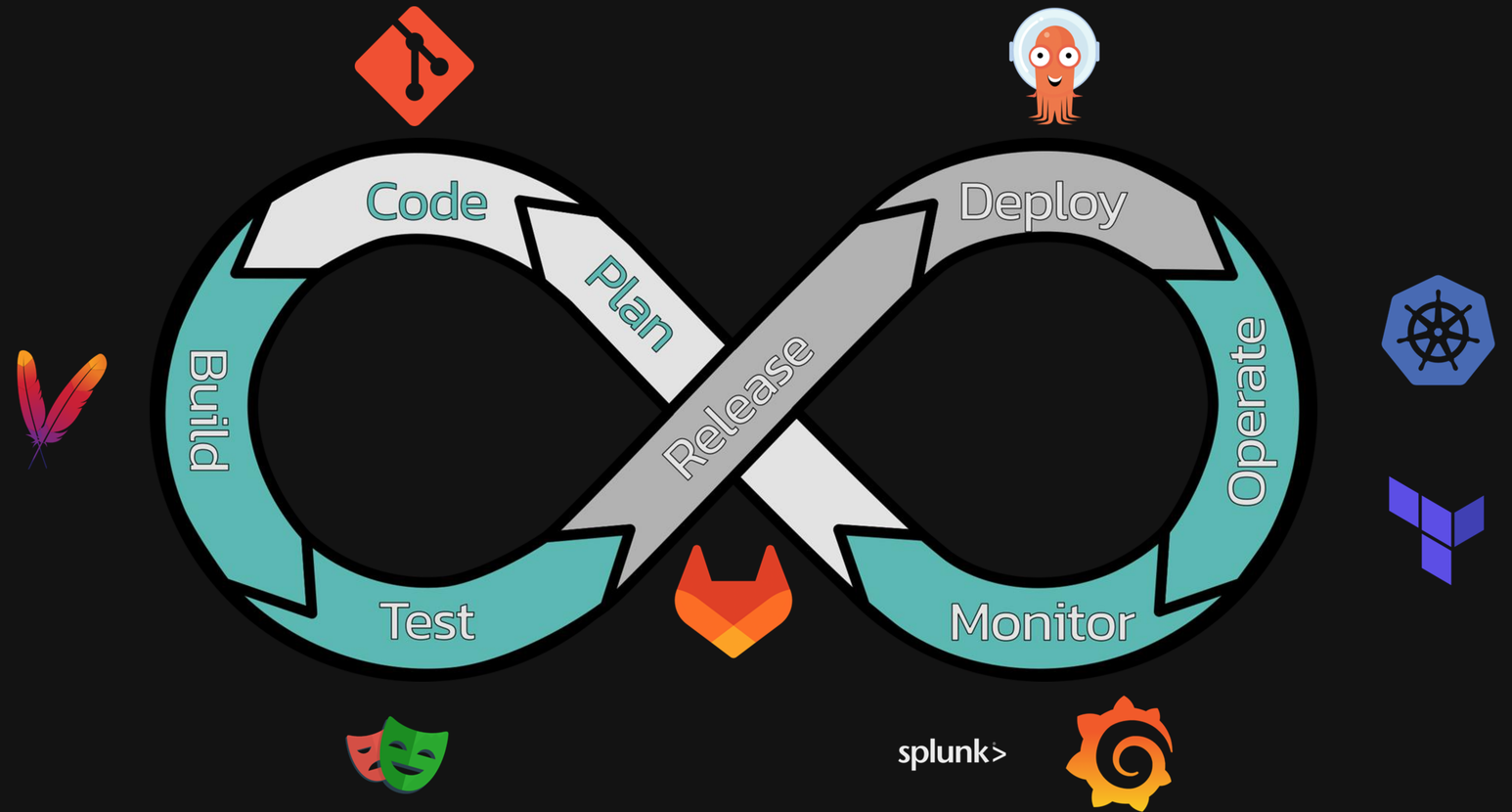
The compliance hierarchy

CRA in a 🥥

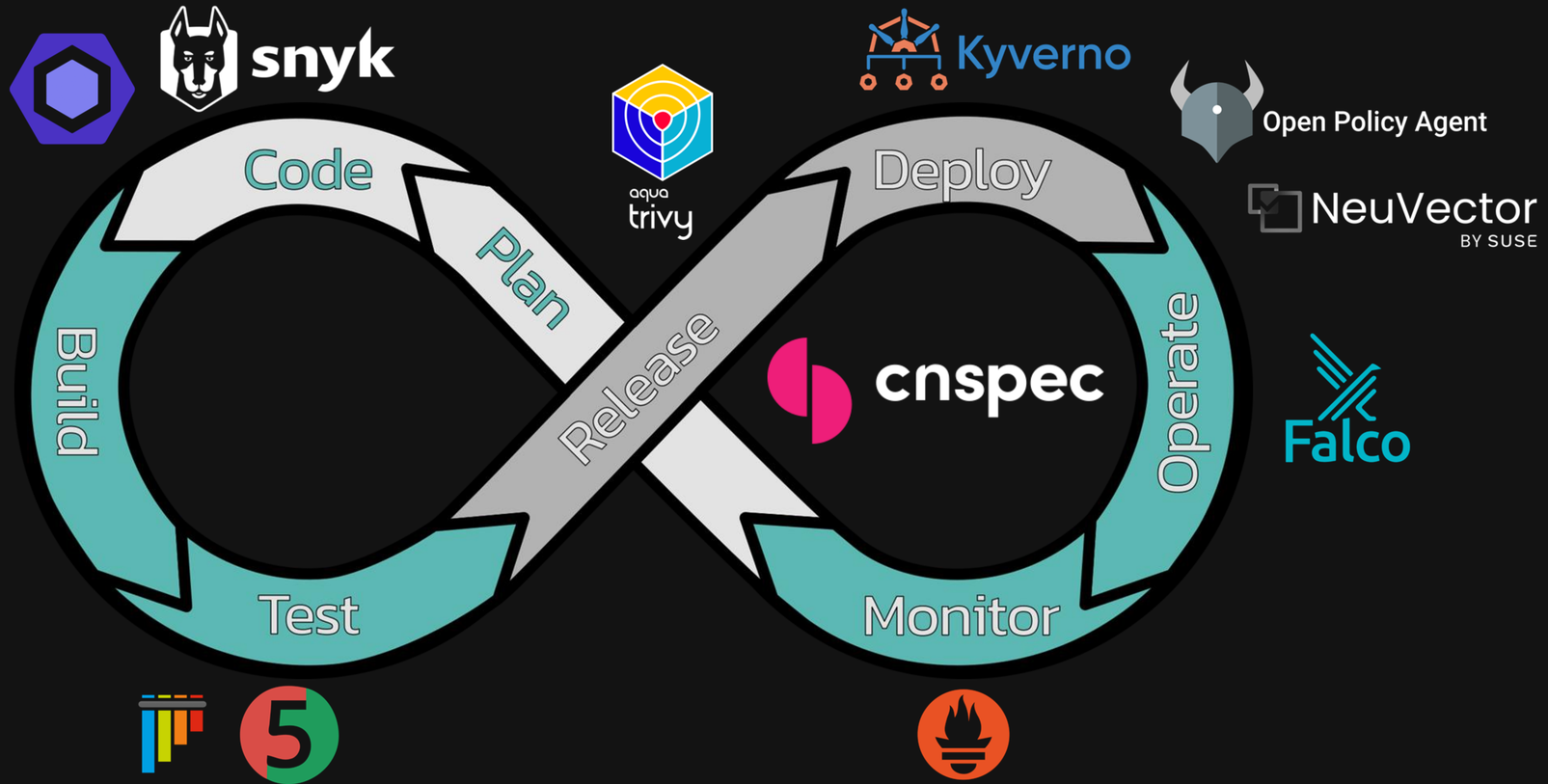
- [..] without known exploitable vulnerabilities.
- [..] authentication, identity or access management systems [..].
- [..] limit attack surfaces.
- [..] software bill of materials.
- [..] effective and regular tests.

So, in reality..
How can we comply?

Delivery Cycle



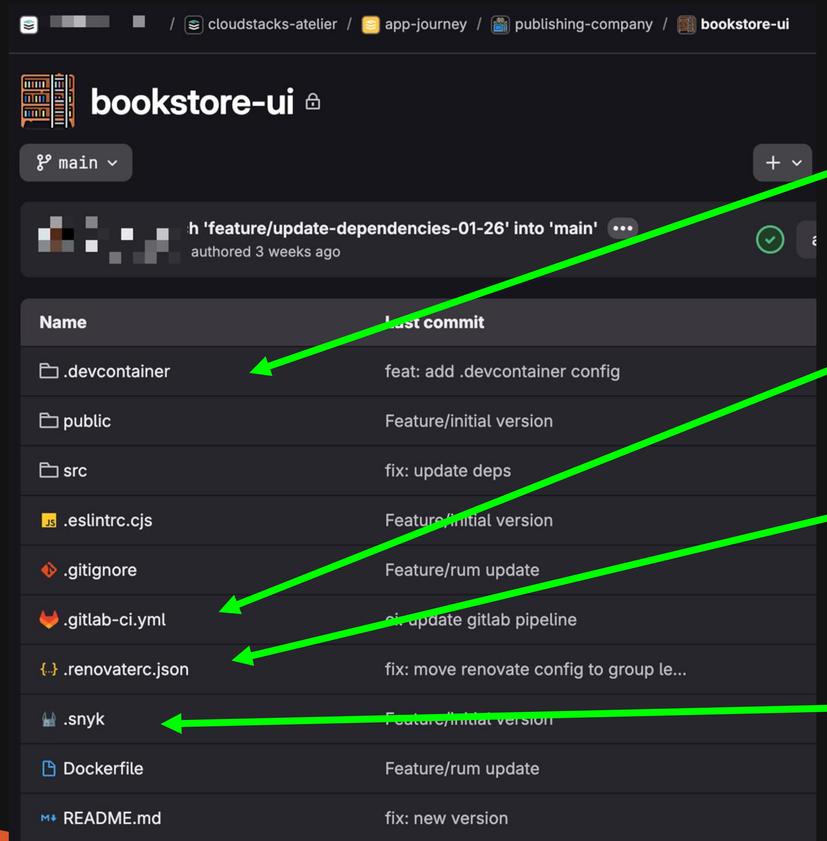
Security Cycle



“**mindset** is the idea,
policy rules are the intention,
enforcing makes it reality”

- Janiel

Automate and Standardize

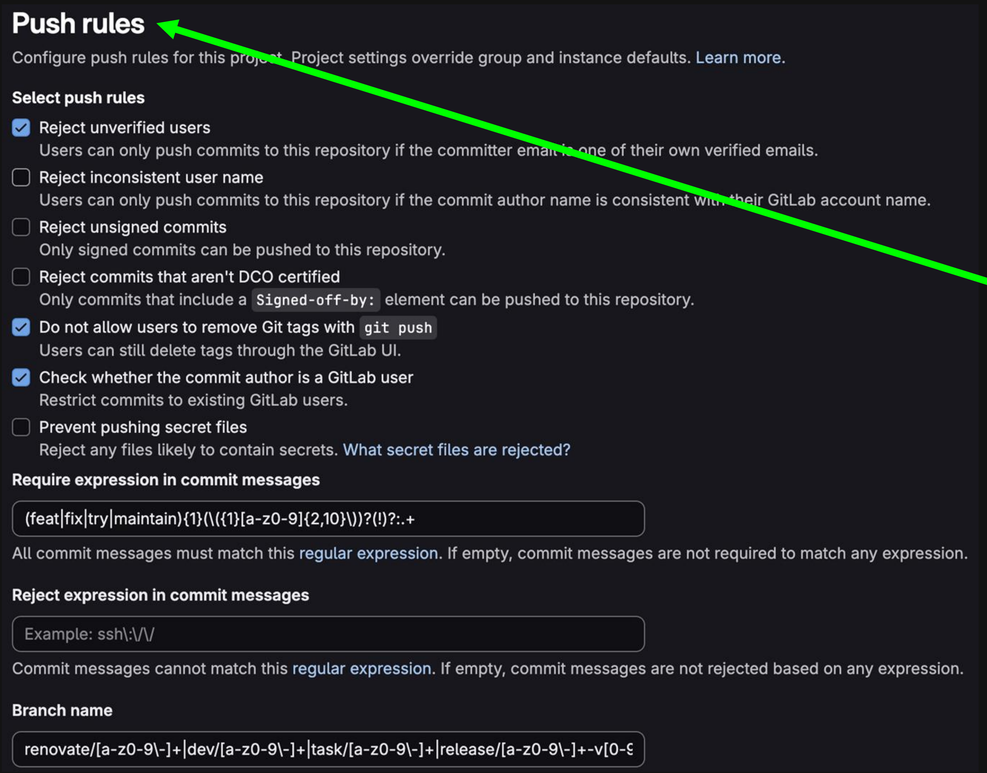


unified dev envs

CI pipelines

autom. dependency updates

active vulnerability management



Repo level
push policy

```hcl
resource "gitlab_project" "module" {
  for_each = { for m in var.modules : m.name => m }

  name             = each.value["name"]
  namespace_id     = each.value["type"]
  ci_config_path   = ".fullstacks-ci.yml"
  visibility_level = "private"

  archive_on_destroy     = true
  request_access_enabled = false

  default_branch        = "main"
  merge_commit_template = file("files/mr-commit-template.md")
  merge_method          = "ff"

  squash_commit_template                           = file("files/squash-template.md")
  suggestion_commit_message                        = "fix: suggestion added"
  merge_requests_template                          = file("files/mr-description-template.md")
  only_allow_merge_if_all_discussions_are_resolved = true

  approvals_before_merge                = 2
  initialize_with_readme                = true
  merge_pipelines_enabled               = true
  merge_requests_enabled                = true
  only_allow_merge_if_pipeline_succeeds = true
  restrict_user_defined_variables       = true
  shared_runners_enabled                = false

  push_rules {
    author_email_regex     = "@(fullstacks.io|noreply.gitlab.com)$"
    branch_name_regex      = "renovate/[a-z0-9\\-]+|dev/[a-z0-9\\-]+...+gm"
    commit_committer_check = true
    # https://www.conventionalcommits.org/en/v1.0.0/
    # "(feat|fix|try|maintain)!?(\\(.*\\))?:.+|^Merge branch.*"
    commit_message_regex = "(feat|fix|try|maintain){1}(\\({1}[a-z0-
9]{2,10}\\))?(!)?:.+"
    deny_delete_tag      = true
    file_name_regex      = "(jar|exe)$"
    max_file_size        = 1 # in MB
    member_check         = true
    # this would also prevent pushing possibly wanted *.pem files (eg. certificates)
    prevent_secrets = false
  }
```

Scan + Guidelines

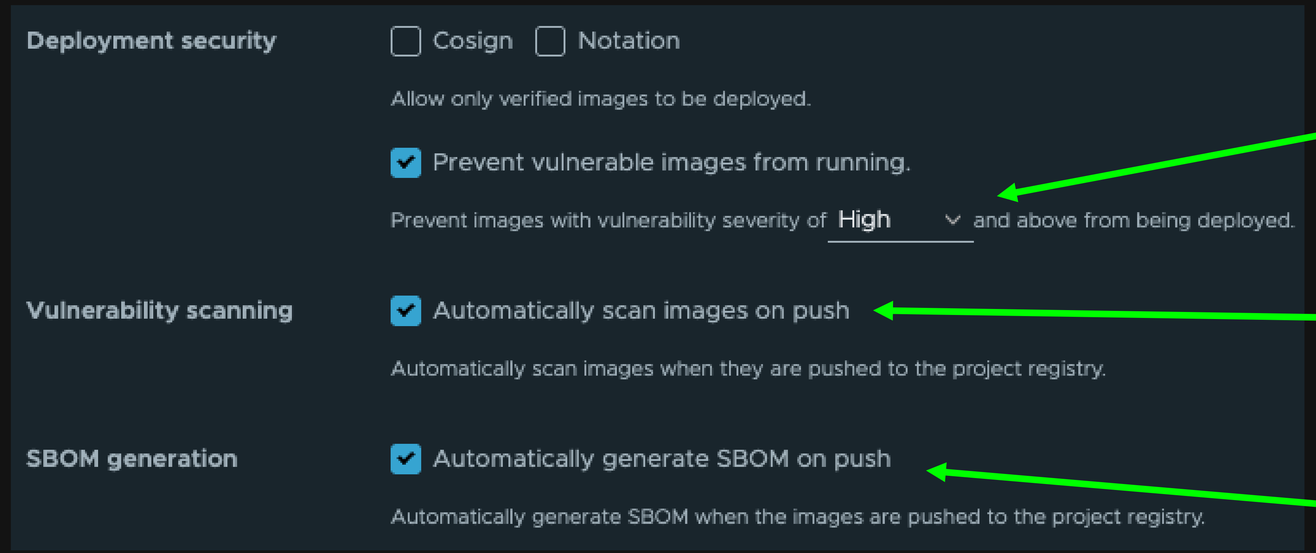
```dockerfile
# Use an official node image as the base image
FROM node:lts-alpine AS build

# Set the working directory
WORKDIR /app

# Copy the package.json and package-lock.json
files
COPY package*.json ./

# Install the dependencies
RUN npm install

# Copy the rest of the application files
COPY . .

# Build the application
RUN npm run build

# Use an official nginx image as the base image
FROM nginx:alpine AS production
```

No pinned version 😯

npm ci ?!

How about rootless image?!

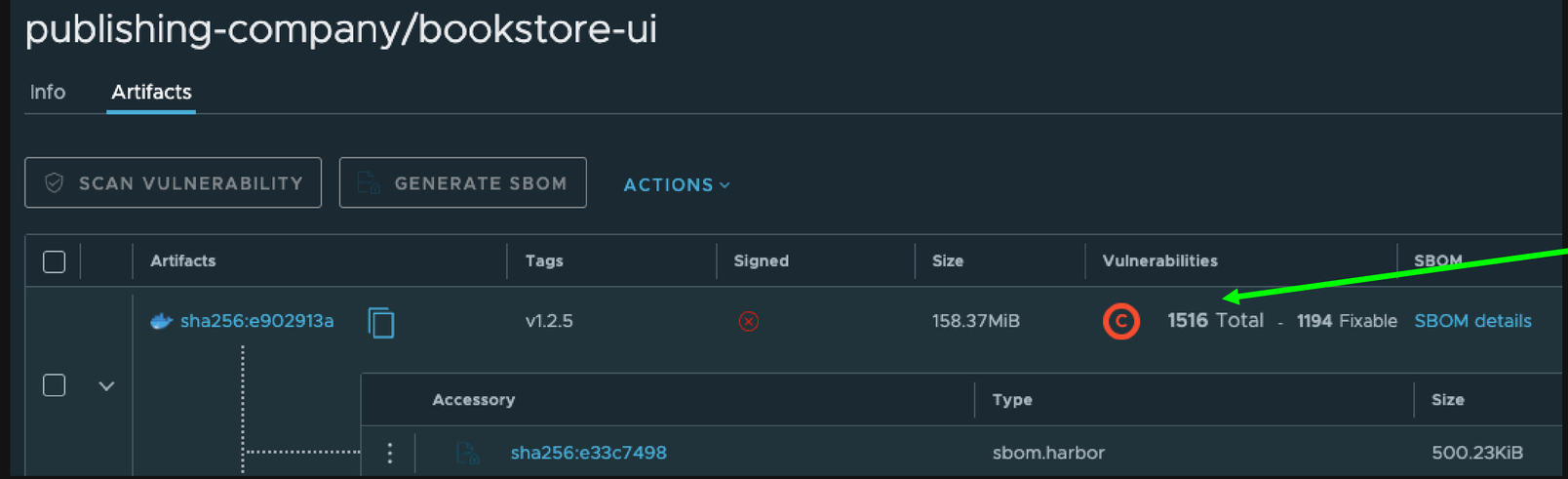
```dockerfile
# Copy the built files from the previous stage
COPY --from=build /app/dist /usr/share/nginx/html

# Copy the nginx configuration file
COPY nginx.conf /etc/nginx/nginx.conf

# Expose port 80
EXPOSE 80

# Start nginx
CMD ["nginx", "-g", "daemon off;"]
```



enforced container registry pull policy

one-click automation

as requested by CRA (cheap option)



don't use this!

```
$ podman pull registry.lab.cloudstacks.eu/ddrack-public/publishing-company/bookstore-ui:v1.2.5
Trying to pull registry.lab.cloudstacks.eu/ddrack-public/publishing-company/bookstore-ui:v1.2.5...
Error: unable to copy from source docker://registry.lab.cloudstacks.eu/ddrack-public/publishing-company/bookstore-ui:v1.2.5: \
    initializing source docker://registry.lab.cloudstacks.eu/ddrack-public/publishing-company/bookstore-ui:v1.2.5: \
    reading manifest v1.2.5 in registry.lab.cloudstacks.eu/ddrack-public/publishing-company/bookstore-ui: \
    unknown: current image with 1516 vulnerabilities cannot be pulled due to configured policy in \
    'Prevent images with vulnerability severity of "High" or higher from running.' \
    To continue with pull, please contact your project administrator to exempt matched \
    vulnerabilities through configuring the CVE allowlist.
```

SCM and images are just the beginning..

enforce policies everywhere

... but **which** policies?

Start with
CIS Benchmarks & internal BP

cnspec

NeuVector
BY SUSE

Open Policy Agent

snyk

cnspec

cnspec

```
policies:
  - uid: mondoo-linux-security
    name: Mondoo Linux Security
    version: 2.5.0
    require:
      - provider: os
    docs:
      desc: |-
        The Mondoo Linux Security policy identifies misconfigurations that could leave
        Linux systems vulnerable to unauthorized access, privilege escalation,
        and data exfiltration. Improperly hardened systems can enable attackers to gain
        initial access, move laterally across networks, and maintain persistent
        access to critical infrastructure.
    groups:
      - title: Users and Groups
        filters: |
          asset.family.contains('linux')
        checks:
          - uid: mondoo-linux-security-root-group-is-empty
    queries:
      - uid: mondoo-linux-security-root-group-is-empty
        title: Ensure root group is empty
        impact: 100
        mql: |
          groups.where(name == "root").all(members == empty || members.all(name == 'root'))
        docs:
          desc: |
            This check ensures that the `root` group, which allows system programs or defined users
            the ability to read and write configurations and files on the system, is properly secured
            by ensuring no users other than the `root` user are assigned to it.
```

```
$ cnspec scan local \
    --sudo \
    --incognito \
    -f policy.mql.yaml
```

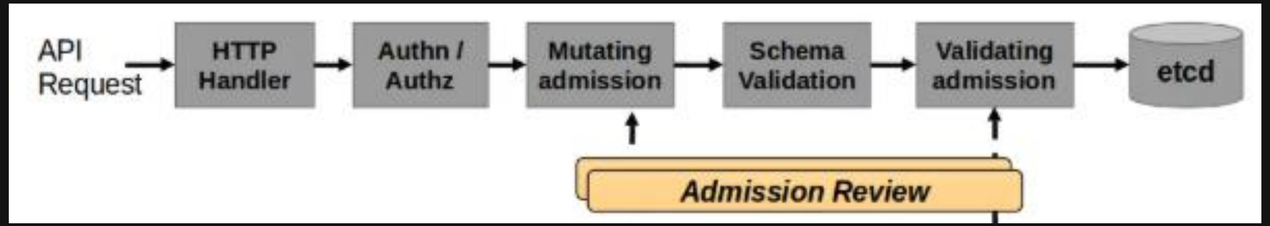
cnspec in Action

NeuVector
BY SUSE

NeuVector
BY SUSE

```yaml
apiVersion: neuvector.com/v1
kind: NvAdmissionControlSecurityRule
metadata:
  name: local
spec:
  rules:
  - action: deny
    comment: only allow custom registry
    criteria:
    - name: imageRegistry
      op: notContainsAny
      path: imageRegistry
      value: "https://registry.fullstacks.io"
    disabled: false
```

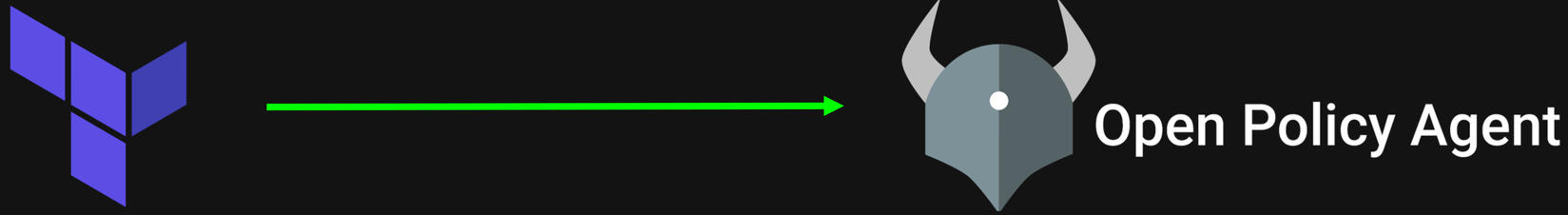
```yaml
  - action: deny
    comment: deny hostIPC
    criteria:
    - name: namespace
      op: notContainsAny
      path: namespace
      value: kube-system,cattle-system,trident,calico-
system,cattle-fleet-system,...
    - name: shareIpcWithHost
      op: =
      path: shareIpcWithHost
      value: "true"
```



NeuVector
BY SUSE

```yaml
apiVersion: neuvector.com/v1
kind: NvSecurityRule
metadata:
  name: nv.bookstore-ui.publishing-company
  namespace: publishing-company
spec:
  dlp:
    settings:
      - action: allow
        name: sensor.creditcard
    status: true
  egress:
    - action: allow
      applications:
        - Consul
        - SSL
      name: nv.consul-server.consul-egress-0
      ports: any
      selector:
        comment: ""
        name: nv.consul-server.consul
        name_referral: true
        original_name: ""
  file: [ ]
  ingress: [
    # many rules
  ]
  process:
    - action: allow
      allow_update: false
      name: consul-dataplane
      path: /usr/local/bin/consul-dataplane
    - action: allow
      allow_update: false
      name: nginx
      path: /usr/sbin/nginx
    # some more processes allowed
  target:
    policymode: Monitor
    selector:
      name: nv.bookstore-ui.publishing-company
```

NeuVector BY SUSE in Action



Open Policy Agent

```
package terraform.rke2_vsphere

import rego.v1

deny contains msg if { msg := prod_guardrails[_] }

var(name) := v if {
  v := input.variables[name].value
}

lower_str(x) := y if {
  y := lower(sprintf("%v", [x]))
}

prod_guardrails contains msg if {
  cn := lower_str(var("cluster_name"))
  contains(cn, "prod")
  msg := sprintf("Refusing: cluster_name looks like production (%v).", [var("cluster_name")])
}
```

```
❯ conftest test tfplan.prod.json -p opa/policy
FAIL - tfplan.prod.json - main - Refusing: cluster_name looks like
production (london-prod).

1 test, 0 passed, 0 warnings, 1 failure, 0 exceptions
```

Open Policy Agent in Action

```yaml
scan_code_snyk:
  stage: scan
  image:
    name: snyk/snyk:node
    entrypoint: [""]
  script:
    - set -x
    - npm ci
    - npm install -g snyk-to-html
    - snyk code test
      --all-projects
      --org=${SNYK_ORG_ID}
      --project-environment=onprem,external
      --project-lifecycle=production
      --severity-threshold=high
      --sarif-file-output=result-sast.sarif.json
      --json | snyk-to-html -o results-code.html
  artifacts:
    expose_as: 'Snyk Code Scan'
    when: always
    paths: [
      'results-code.html',
      'result-sast.sarif.json'
    ]
```

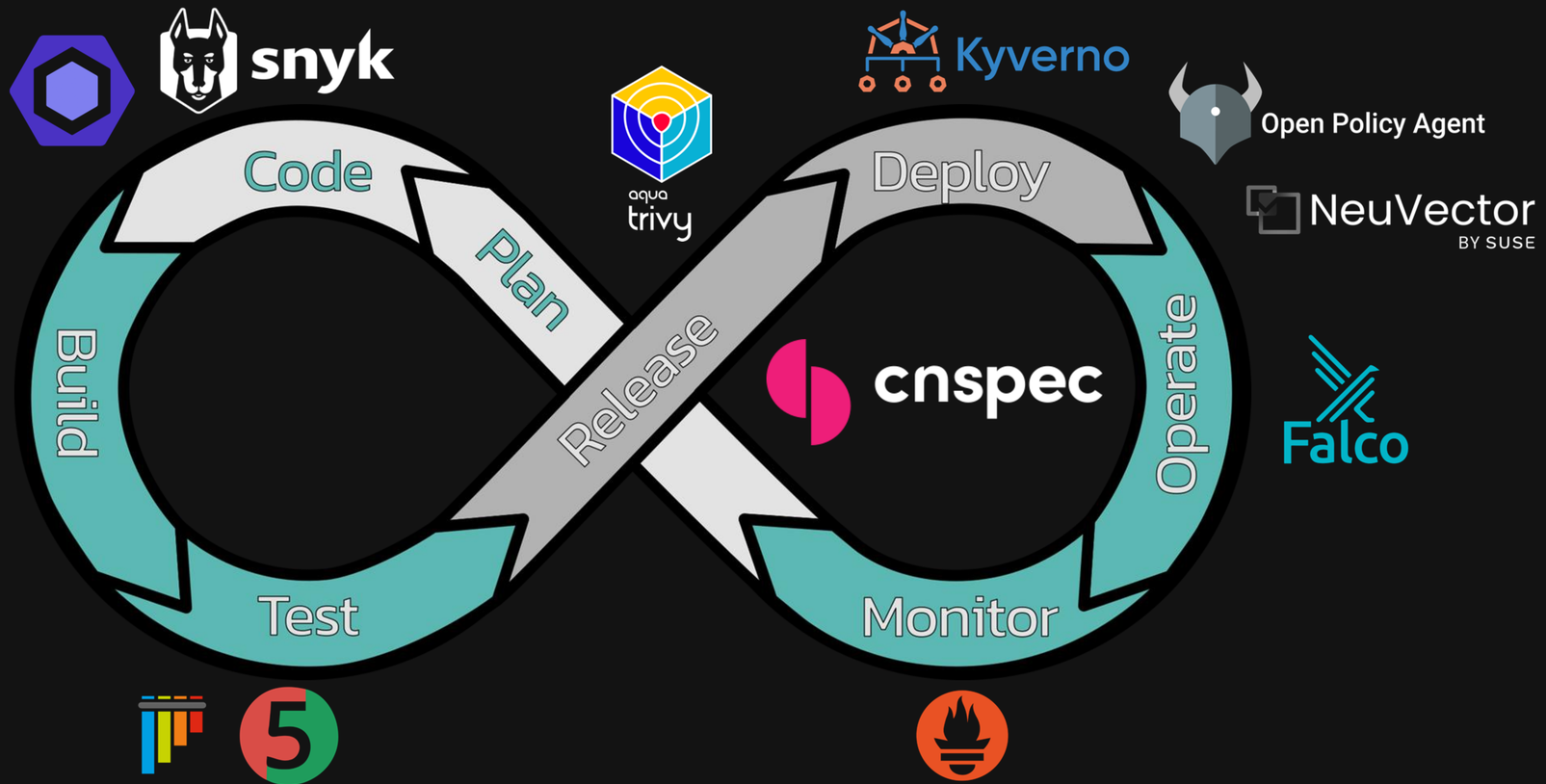
SCAN
POLICY

```yaml
# Snyk (https://snyk.io) policy file,
# patches or ignores known vulnerabilities.
version: v1.25.0
# ignores vulnerabilities until expiry date;
# change duration by modifying expiry date
ignore:
  SNYK-DEBIAN12-ZLIB-6008963:
    - "*":
        reason: None Given
        created: 2024-03-26T14:17:58.892Z
        expires: 2099-03-26T14:17:58.892Z
  SNYK-DEBIAN11-ZLIB-6008961:
    - "*":
        reason: None Given
        created: 2024-03-26T14:17:58.892Z
        expires: 2099-03-26T14:17:58.892Z
patch: {}
```

in Action

Delivery Cycle

Security Cycle



Policies everywhere, always test, dare to enforce.

Security becomes reality when “should” turns into “cannot”.

If it’s not enforced, it’s merely a suggestion.

Daniel Drack

@DrackThor

Jan Hoepfner

@jeypie