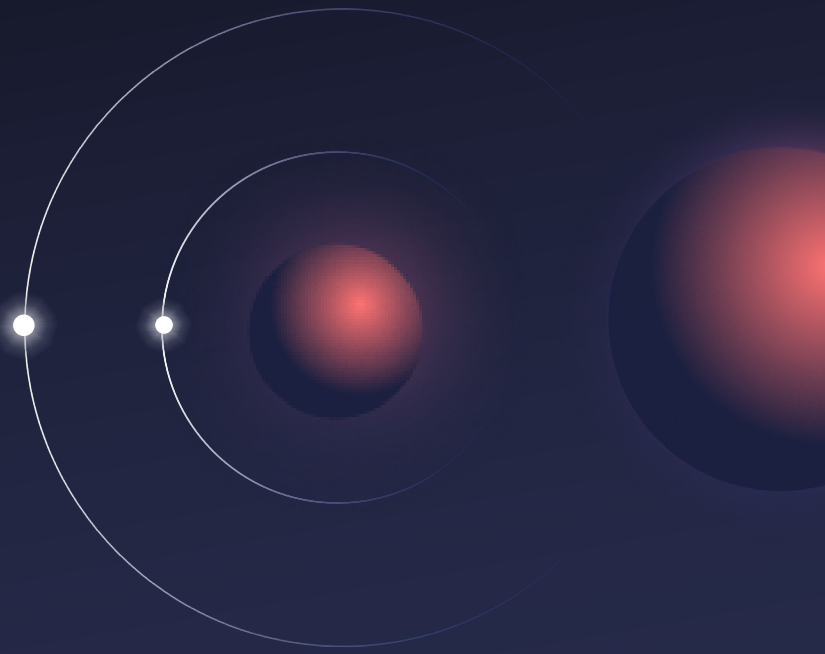

# PolicyEngine в Kubernetes – что, как, зачем?



Сергей Канибор
R&D/Container Security, Luntry

# whoami

- R&D/Container Security в Luntry
- Специализируюсь на безопасности контейнеров и Kubernetes
- Спикер PHDays, VolgaCTF, HackConf, CyberCamp, БЕКОН
- Редактор телеграм канала @k8security

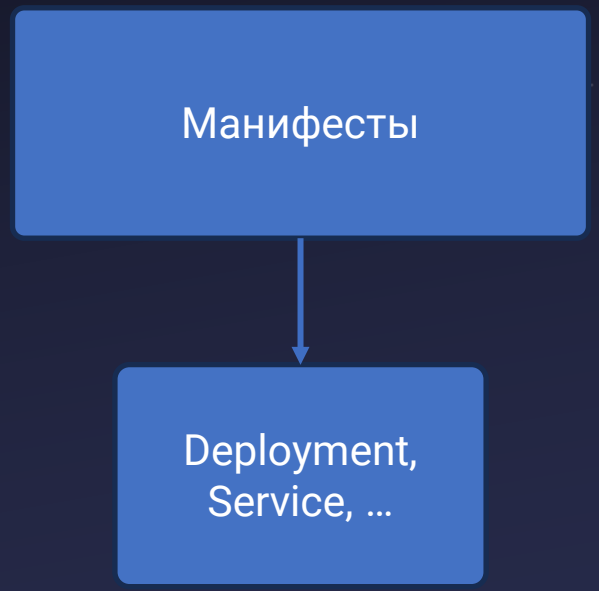
# Agenda

- Что происходит после kubectl apply
- Контроль ресурсов
- Policy Engines
- Demo

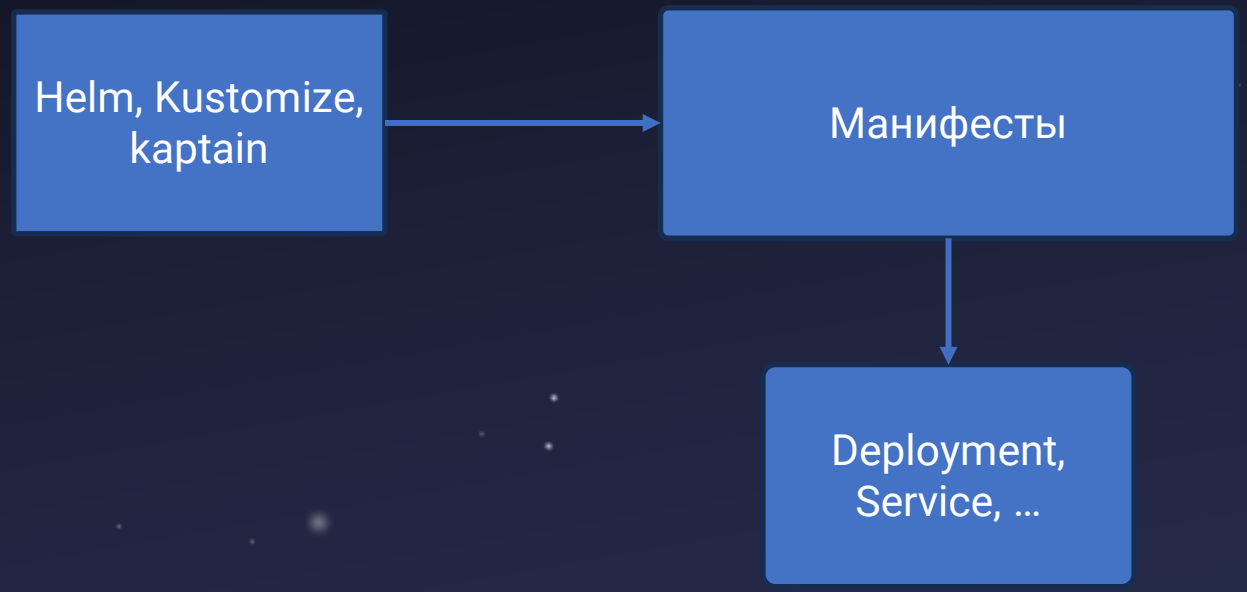
# Стадия деплоя в Kubernetes

Манифесты

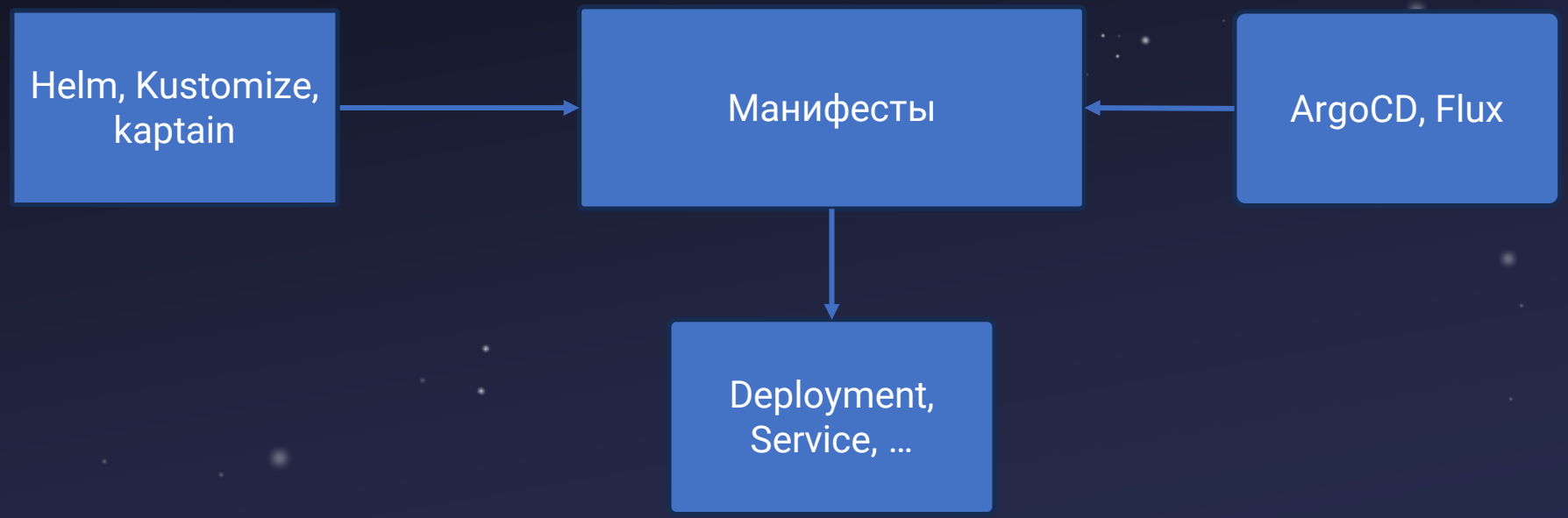
# Стадия деплоя в Kubernetes



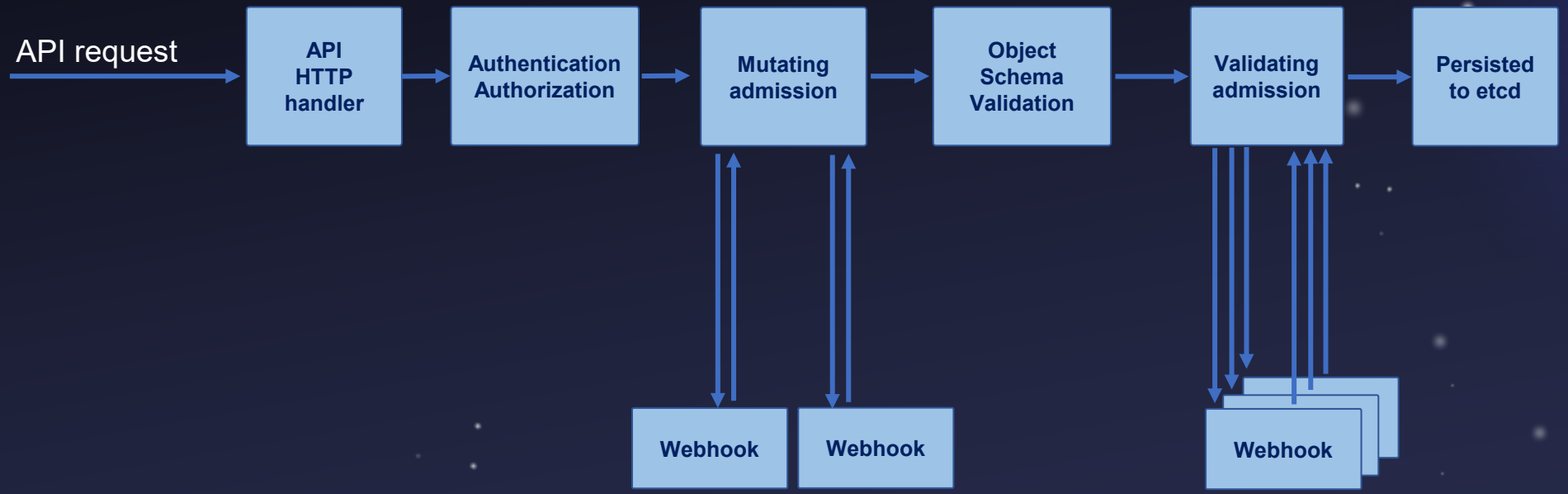
# Стадия деплоя в Kubernetes



# Стадия деплоя в Kubernetes



# Что происходит после



# Встроенные Admission Controllers

- LimitRanger
- PodSecurityPolicy
- ResourceQuota
- ImagePolicyWebhook
- MutatingAdmissionWebhook
- ValidatingAdmissionWebhook
- …
- Всего около 30 встроенных

```
kube-apiserver --enable-admission-plugins=LimitRanger,ResourceQuota ...
kube-apiserver --disable-admission-plugins=PodNodeSelector,AlwaysDeny ...
```

# Dynamic Admission Control

```yaml
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
...
webhooks:
- name: my-webhook.example.com
  objectSelector:
    matchLabels:
      foo: bar
  rules:
  - operations: ["CREATE"]
    apiGroups: ["*"]
    apiVersions: ["*"]
    resources: ["*"]
    scope: "*"
  ...
```

```json
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": true,
    "patchType": "JSONPatch",
    "patch": "W3sib3AiOiAiYWRkIiwgInBhdGgiOiAiL3NwZWMvcmVwbGljYXMiLCAidmFsdWUiOiAzfV0="
  }
}
```

# Dynamic Admission Control

- Нужен вебхук, который будет валидировать запросы
- По сути простой веб-сервер, который будет принимать и отправлять запросы
- В самом перехватчике описываем логику которую нужно валидировать/изменять
- Веб-перехватчики вызываются только через TLS/SSL, поэтому ваш веб-перехватчик должен иметь действительный подписанный сертификат

# Dyna dmission Control

- Нужен вебхук, к дет валидирова
- По сути простой ве оторый б имать и отправлять запросы
- В самом перехватчике опи которую нужно валидировать/изменять
- Веб-перехватчики вызы TLS/SSL, поэтому ваш веб-перехватчик долже йствите писанный сертификат

# Policy Engines

- Используем готовый движок
- Пишем только политики
- Сами политики хранятся в CRD

Open Policy Agent

# Зоопарк Policy Engines

- OPA
- Gatekeeper
- Kyverno
- MagTape
- K-rail
- Kubewarden
- JSPolicy

# OPA Gatekeeper VS Kyverno

| Признак сравнения | OPA Gatekeeper | Kyverno |
|---|---|---|
| Язык политик | Rego | Kubernetes-like |
| Реакция на недоступность webhook | Настройка на уровне конфига OPA | Можно настроить отдельно для критичных политик |
| Фоновое сканирование | Отсутствует | Background scan для каждой политики |
| Расширяемость кода и данных | Через CRD Provider | External Data Sources |
| Количество политик от сообщества | 44 политики | 283 политики |
| Возможность работы вне Kubernetes кластера (встраивание в CI/CD) | Conftest | Kyverno CLI |

# Пример: Политика OPA Gatekeeper

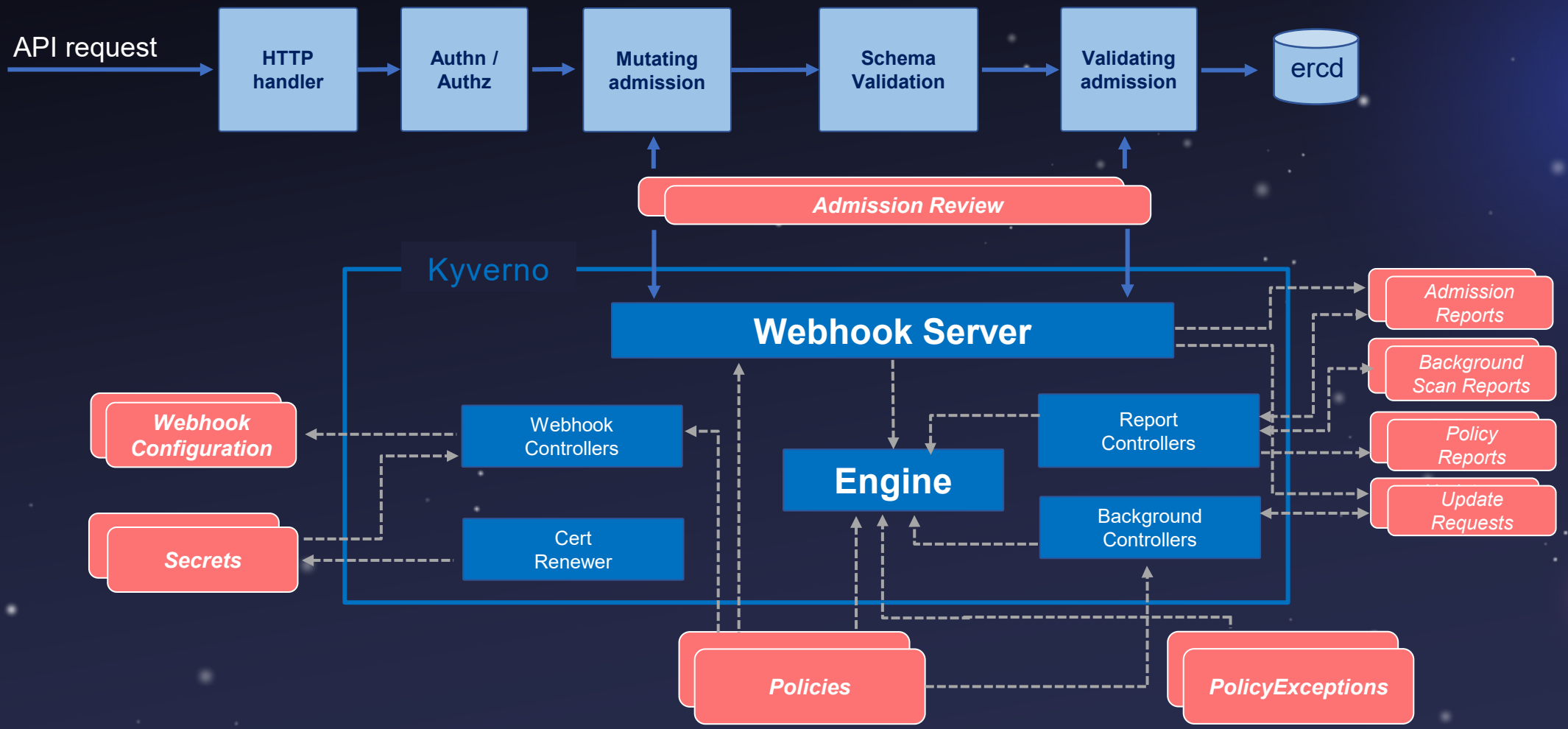
```yaml
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: k8sallowedrepos
spec:
  crd:
    spec:
      names:
        kind: K8sAllowedRepos
      validation:
        # Schema for the `parameters` field
        openAPIV3Schema:
          properties:
            repos:
              type: array
              items:
                type: string
  targets:
    - target: admission.k8s.gatekeeper.sh
      rego: |
        package k8sallowedrepos

        violation[{"msg": msg}] {
          container := input.review.object.spec.containers[_]
          satisfied := [good | repo = input.parameters.repos[_] ; good = contains(container.image, repo)]
          not any(satisfied)
          msg := sprintf("container <%v> has an invalid image repo <%v>, allowed repos are %v", [container.name, container.image, inpu
        }

        violation[{"msg": msg}] {
          container := input.review.object.spec.initContainers[_]
          satisfied := [good | repo = input.parameters.repos[_] ; good = contains(container.image, repo)]
          not any(satisfied)
          msg := sprintf("container <%v> has an invalid image repo <%v>, allowed repos are %v", [container.name, container.image, inpu
        }
```

```yaml
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sAllowedRepos
metadata:
  name: allow-only-private-registry
spec:
  match:
    kinds:
      - apiGroups: [""]
        kinds: ["Pod"]
  parameters:
    repos:
      - "private.example.com"
```

# Пример: Политика Kyverno

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: restrict-image-registries
  annotations:
    policies.kyverno.io/title: Restrict Image Registries
    policies.kyverno.io/category: Best Practices
    policies.kyverno.io/severity: medium
    policies.kyverno.io/minversion: 1.3.0
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Images from unknown, public registries can be of dubious quality and may not be
      scanned and secured, representing a high degree of risk. Requiring use of known, approved
      registries helps reduce threat exposure by ensuring image pulls only come from them. This
      sample validates that container images only originate from the registry `eu.foo.io` or
      `bar.io`.
spec:
  validationFailureAction: audit
  background: true
  rules:
  - name: validate-registries
    match:
      resources:
        kinds:
        - Pod
    validate:
      message: "Unknown image registry."
      pattern:
        spec:
          containers:
          - image: "eu.foo.io/* | bar.io/*"
```

# Архитектура Kyverno



# Возможности Kyverno

- Политики как Kubernetes Resources
- Validate, Mutate, Generate, Cleanup
- Проверка образов контейнеров для software supply chain security
- Сопоставление ресурсов через label selectors и wildcards
- Использование исключений в политиках
- Получение результатов о работе в отдельных CRD
- Встраивание в CI/CD через Kyverno CLI
- ...

# DEMO TIME

# Спасибо за внимание!

Telegram: @r0binak

E-mail: sk@luntry.ru


luntry.ru