



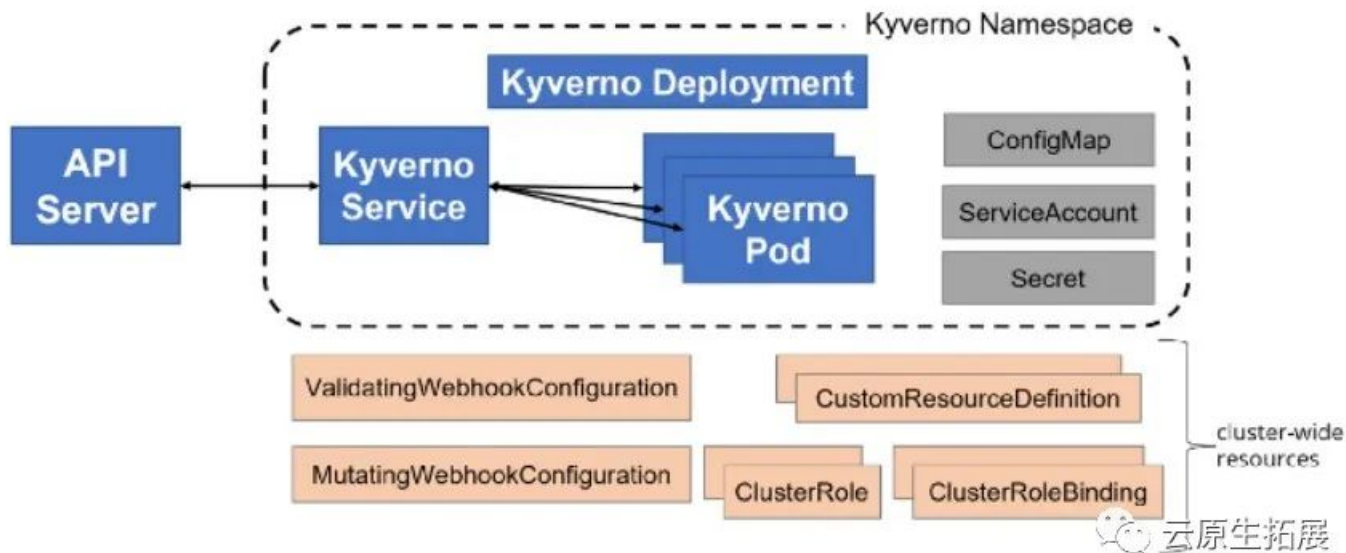
什么是 Kyverno ?

Kyverno是为Kubernetes设计的策略引擎。使用Kyverno，策略被作为Kubernetes资源来管理，不需要新的语言来编写策略。这允许使用熟悉的工具，如kubectl、git和kustomize来管理策略。

Kyverno 有哪些功能？

- 创建策略，作为 Kubernetes 资源 (没有新语言要学习!)
- 验证、变异或生成任何资源
- 为软件供应安全验证容器镜像

Kyverno 架构:



快速案例:

我们将在minikube集群中的命名空间上应用这三个策略::

- **禁止NodePort策略:** 验证任何新的服务不使用 'NodePort' 类型。
- **要求资源限制策略:** 验证所有容器都为内存和CPU请求和内存限制指定了一些东西。
- **禁止最新标签策略:** 验证镜像指定了一个标签, 并且它不是被称为 'latest' 。

环境设置**

```
Clone git branch to download the app demo :git clonehttps://github.com/vfarcic/kyverno-demodc kyverno-demo
```

Kyverno 安装:

```
kubectl create -filename
https://raw.githubusercontent.com/kyverno/kyverno/main/config/install.yaml
kubectl --namespace kyverno rollout status deployment kyverno
cp app/orig.yaml app/app.yaml
kubectl create namespace production
```

```
kyverno-demo git:(master) kubectl create \
--filename https://raw.githubusercontent.com/kyverno/kyverno/main/config/install.yaml
namespace/kyverno created
customresourcedefinition.apiextensions.k8s.io/admissionreports.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/backgroundscanreports.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/clusteradmissionreports.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/clusterbackgroundscanreports.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/clusterpolicies.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/clusterpolicyreports.wgpolicyk8s.io created
customresourcedefinition.apiextensions.k8s.io/generaterequests.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/policies.kyverno.io created
customresourcedefinition.apiextensions.k8s.io/policyreports.wgpolicyk8s.io created
customresourcedefinition.apiextensions.k8s.io/updaterequests.kyverno.io created
serviceaccount/kyverno-service-account created
role.rbac.authorization.k8s.io/kyverno:leaderelection created
clusterrole.rbac.authorization.k8s.io/kyverno created
clusterrole.rbac.authorization.k8s.io/kyverno:admin-generaterequest created
clusterrole.rbac.authorization.k8s.io/kyverno:admin-policies created
clusterrole.rbac.authorization.k8s.io/kyverno:admin-policyreport created
clusterrole.rbac.authorization.k8s.io/kyverno:admin-reports created
clusterrole.rbac.authorization.k8s.io/kyverno:admin-updaterequest created
clusterrole.rbac.authorization.k8s.io/kyverno:events created
clusterrole.rbac.authorization.k8s.io/kyverno:generate created
clusterrole.rbac.authorization.k8s.io/kyverno:policies created
clusterrole.rbac.authorization.k8s.io/kyverno:userinfo created
clusterrole.rbac.authorization.k8s.io/kyverno:view created
clusterrole.rbac.authorization.k8s.io/kyverno:webhook created
rolebinding.rbac.authorization.k8s.io/kyverno:leaderelection created
clusterrolebinding.rbac.authorization.k8s.io/kyverno created
configmap/kyverno created
configmap/kyverno-metrics created
service/kyverno-svc created
service/kyverno-svc-metrics created
deployment.apps/kyverno created
→ kyverno-demo git:(master) █
```

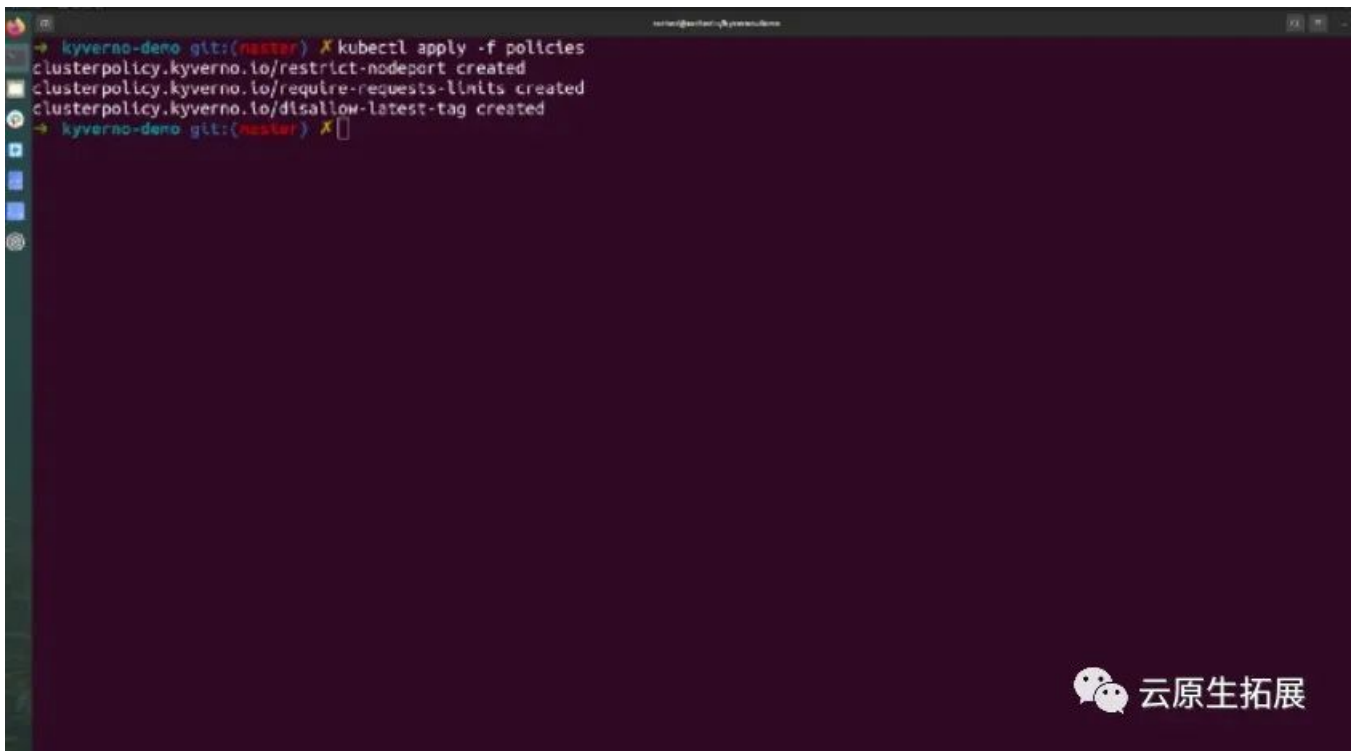
云原生拓展

```
kyverno-demo git:(master) kubectl --namespace kyverno \
rollout status \
deployment kyverno
Waiting for deployment "kyverno" rollout to finish: 0 of 1 updated replicas are available...
deployment "kyverno" successfully rolled out
→ kyverno-demo git:(master) █
```

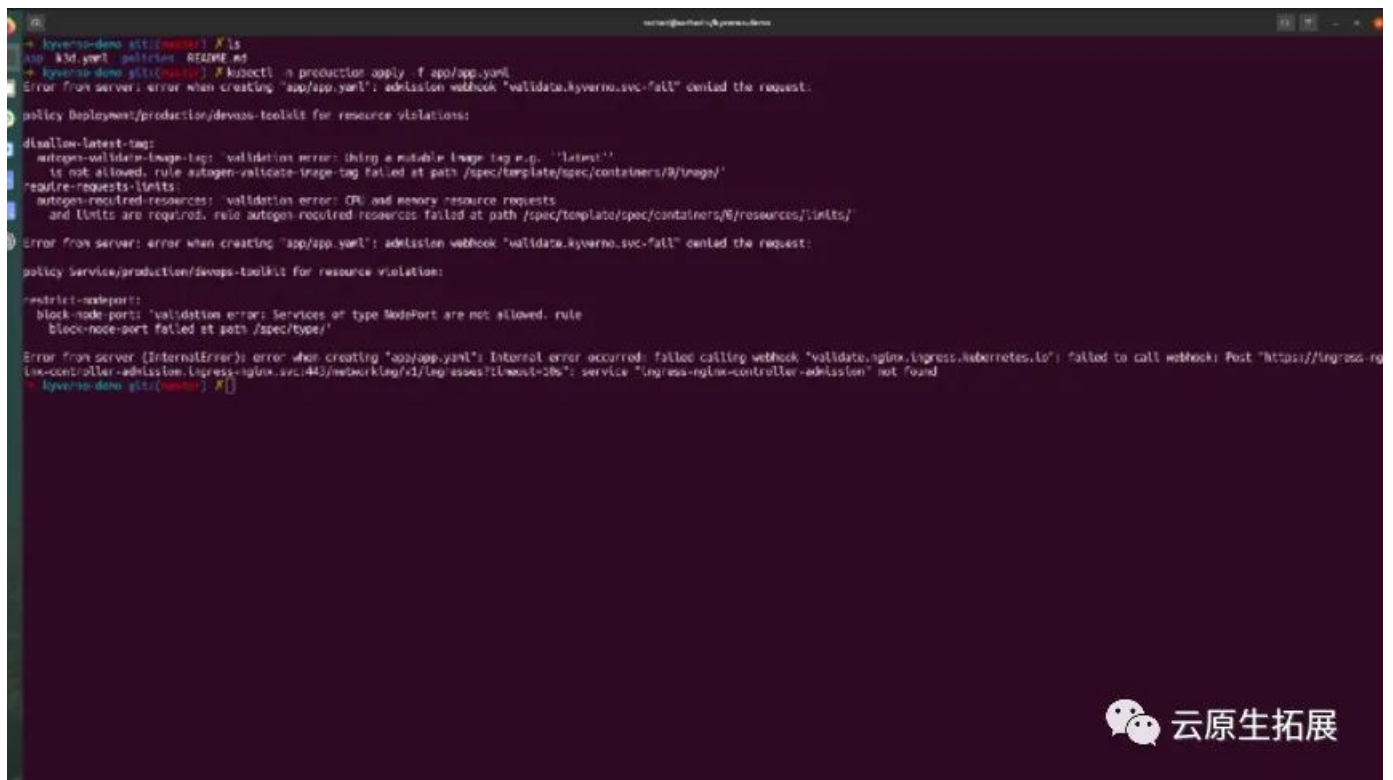
云原生拓展

现在，我们实行三大宏伟而美丽的政策：

```
kyverno-demo git:(master) kubectl apply --filename policies/
```



正如我们在这里看到的3个错误。这意味着我们的政策非常有效!



现在让我们打开应用程序的yaml文件并修复所有的错误!:: D

Error1: (修改 service 类型为 Cluster IP)

```
spec:
  type: ClusterIP
```

Error2: (将镜像 tag 从 Latest 修改为 3.0.0)

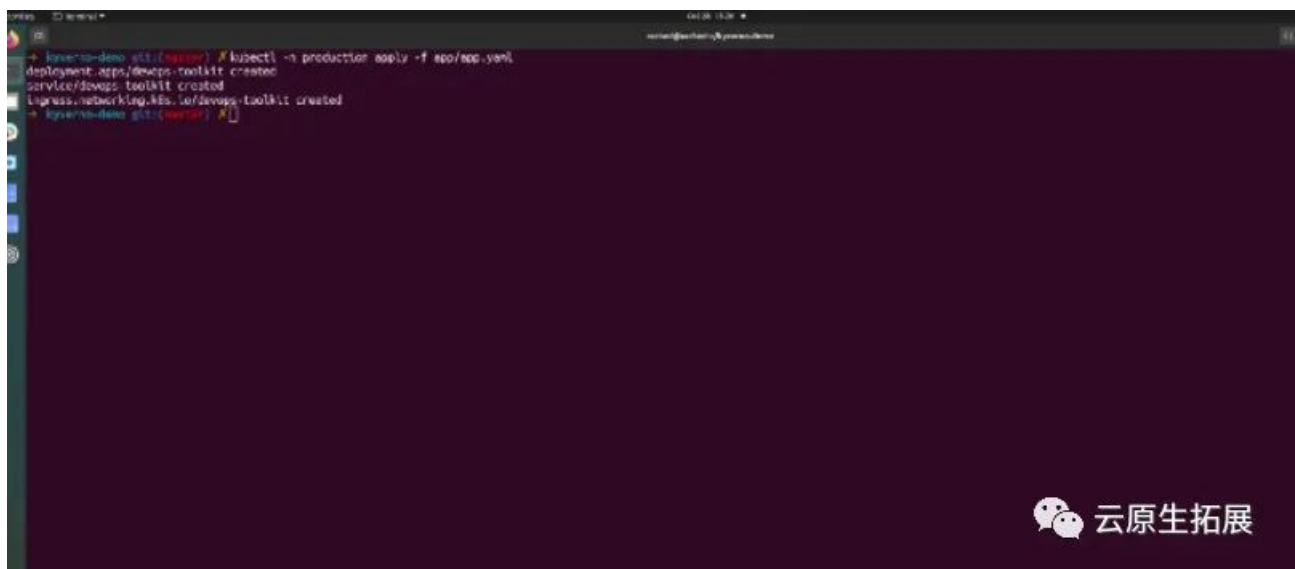
```
- name: devops-toolkit
  image: vfaric/devops-toolkit-series:3.0.0
```

Error3: (指定 resources 信息)

```
resources:
  requests:
    cpu: 250m
    memory: 250Mi
  limits:
    cpu: 500m
    memory: 500M
```

修改后, 重新验证.

然后 Voila !



PS: 您可以通过删除前面创建的名称空间“production”来销毁所有这些内容

```
kubectl delete namespace production
```

最后, 如果您喜欢Kyverno的工作方式, 并且想深入研究, 这里有一长串策略供您使用(<https://kyverno.io/policies/>)。

欢迎关注我的公众号“云原生拓展”, 原创技术文章第一时间推送。