讲师：李振良（阿良）

今天课题：《供应链安全》

学院官网：www.aliangedu.cn

阿良个人微信

DevOps技术栈公众号

第五章 供应链安全

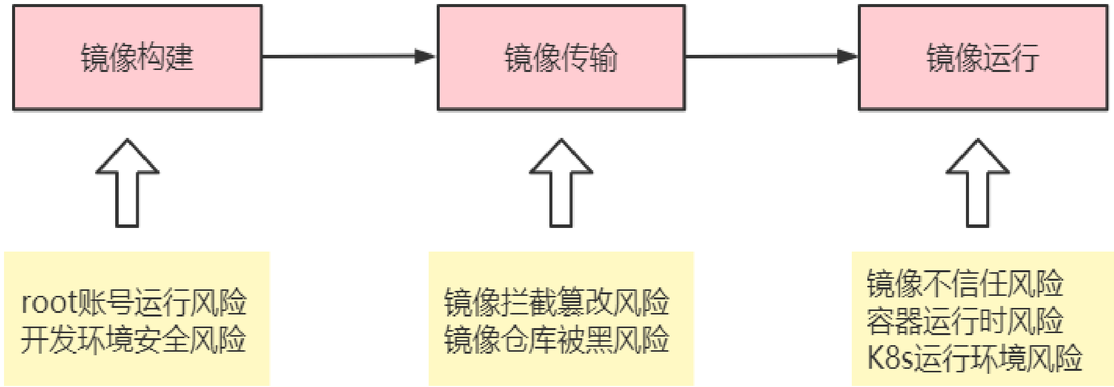
- 可信任软件供应链概述
- 构建镜像Dockerfile文件优化
- 镜像漏洞扫描工具：Trivy
- 检查YAML文件安全配置：kubesec
- 准入控制器： Admission Webhook
- 准入控制器： ImagePolicyWebhook

可信任软件供应链概述

可信任软件供应链：指在建设基础架构过程中，涉及的软件都是可信任的。

可信任软件供应链概述

在K8s领域可信软件供应链主要是指镜像，因为一些软件交付物都是镜像，部署的最小载体。



构建镜像Dockerfile文件优化

- **减少镜像层：**一次RUN指令形成新的一层，尽量Shell命令都写在一行，减少镜像层。
- **清理无用文件：**清理对应的残留数据，例如yum缓存。
- **清理无用的软件包：**基础镜像默认会带一些debug工具，可以删除掉，仅保留应用程序所需软件，防止黑客利用。
- **选择最小的基础镜像**：例如alpine
- **使用非root用户运行：**USER指令指定普通用户

构建镜像Dockerfile文件优化

示例：构建python web镜像

```
FROM python
RUN useradd python
RUN mkdir /data/www -p
COPY . /data/www
RUN chown -R python /data
RUN pip install flask -i https://mirrors.aliyun.com/pypi/simple/
WORKDIR /data/www
USER python
CMD python main.py
```

镜像漏洞扫描工具：Trivy

Trivy：是一种用于容器镜像、文件系统、Git仓库的漏洞扫描工具。发现目标软件存在的漏洞。Trivy易于使用，只需安装二进制文件即可进行扫描，方便集成CI系统。

项目地址：https://github.com/aquasecurity/trivy

镜像漏洞扫描工具：Trivy

示例：

```
# 容器镜像扫描
trivy image nginx
trivy image -i nginx.tar
# 打印指定（高危、严重）漏洞信息
trivy image -s HIGH nginx
trivy image -s HIGH, CRITICAL nginx
# JSON格式输出并保存到文件
trivy image nginx -f json -o /root/output.json
```

检查YAML文件安全配置：kubesec

kubesec：是一个针对K8s资源清单文件进行安全配置评估的工具，根据安全配置最佳实践来验证并给出建议。

官网：https://kubesec.io

项目地址：https://github.com/controlplaneio/kubesec

检查YAML文件安全配置：kubesec

示例：

kubesec scan deployment.yaml

或者使用容器环境执行检查

docker run -i kubesec/kubesec scan /dev/stdin < deployment.yaml

kubesec内置一个HTTP服务器，可以直接启用，远程调用。

- **二进制**

kubesec http 8080 &

- **Docker容器**

docker run -d -p 8080:8080 kubesec/kubesec http 8080

示例：

curl -sSX POST --data-binary @deployment.yaml http://192.168.31.71:8080/scan

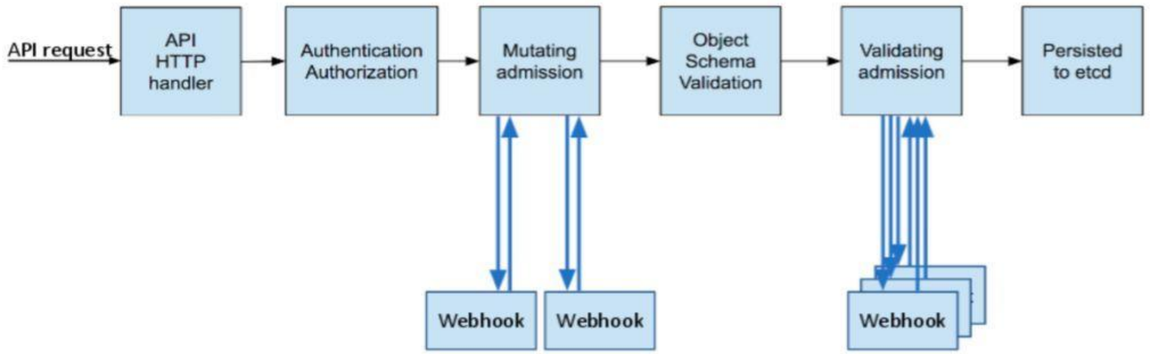
准入控制器： Admission Webhook

Admission Webhook：准入控制器Webhook是准入控制插件的一种，用于拦截所有向APISERVER发送的请求，并且可以修改请求或拒绝请求。

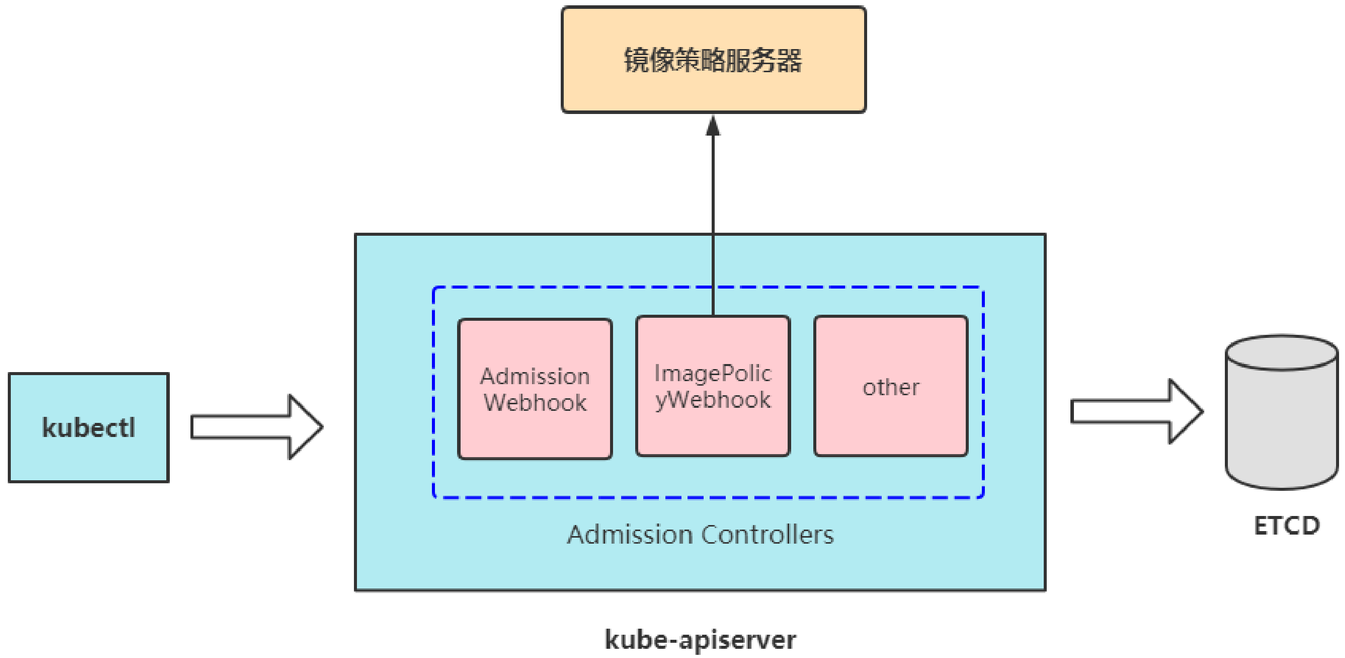
Admission webhook为开发者提供了非常灵活的插件模式，在kubernetes资源持久化之前，管理员通过程序可以对指定资源做校验、修改等操作。例如为资源自动打标签、pod设置默认SA，自动注入sidecar容器等。

相关Webhook准入控制器：

- MutatingAdmissionWebhook：修改资源，理论上可以监听并修改任何经过ApiServer处理的请求
- ValidatingAdmissionWebhook：验证资源
- ImagePolicyWebhook：镜像策略，主要验证镜像字段是否满足条件



准入控制器： ImagePolicyWebhook



工作流程图

准入控制器： ImagePolicyWebhook

1、启用准入控制插件

```
--enable-admission-plugins=NodeRestriction,ImagePolicyWebhook
--admission-control-config-file=/etc/kubernetes/image-policy/admission_configuration.yaml
并使用hostpath数据卷将宿主机/etc/kubernetes/image-policy目录挂载到容器中
```

2、准备配置文件

```
# /etc/kubernetes/image-policy/admission_configuration.yaml
apiVersion: apiserver.config.k8s.io/v1
kind: AdmissionConfiguration
plugins:
- name: ImagePolicyWebhook
  configuration:
    imagePolicy:
      kubeConfigFile: /etc/kubernetes/image-policy/connect_webhook.yaml  # 连接镜像策略服务器配置文件
      allowTTL: 50 # 控制批准请求的缓存时间，单位秒
      denyTTL: 50 # 控制批准请求的缓存时间，单位秒
      retryBackoff: 500 # 控制重试间隔，单位毫秒
      defaultAllow: true # 确定webhook后端失效时的行为
```

准入控制器： ImagePolicyWebhook

2、准备配置文件

```
apiVersion: v1
kind: Config
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/image-policy/webhook.pem # 数字证书，用于验证远程服务
    server: https://192.168.31.73:8080/image_policy # 镜像策略服务器地址，必须是https
  name: webhook
contexts:
- context:
    cluster: webhook
    user: apiserver
  name: webhook
current-context: webhook
preferences: {}
users:
- name: apiserver
  user:
    client-certificate: /etc/kubernetes/image-policy/apiserver-client.pem # webhook准入控制器使用的证书
client-key: /etc/kubernetes/image-policy/apiserver-client-key.pem # 对应私钥证书
```

注：涉及的证书文件，下一步生成，拷贝到该文件中对应路径

准入控制器： ImagePolicyWebhook

3、部署镜像服务器

自己用python开发一个简单的webhook端点服务器，作用是拒绝部署的镜像乜有指定标签（即latest）。

3.1 自签HTTPS证书

3.2 Docker容器启动镜像策略服务

```
docker run -d -u root --name=image-policy-webhook \
-v $PWD/webhook.pem:/data/www/webhook.pem \
-v $PWD/webhook-key.pem:/data/www/webhook-key.pem \
-e PYTHONUNBUFFERED=1 -p 8080:8080 \
lizhenliang/image-policy-webhook
```

4、测试

```
kubectl create deployment web1 --image=nginx:1.16
kubectl create deployment web2 --image=nginx
```

谢谢

阿良个人微信

DevOps技术栈公众号

阿良教育：www.aliangedu.cn