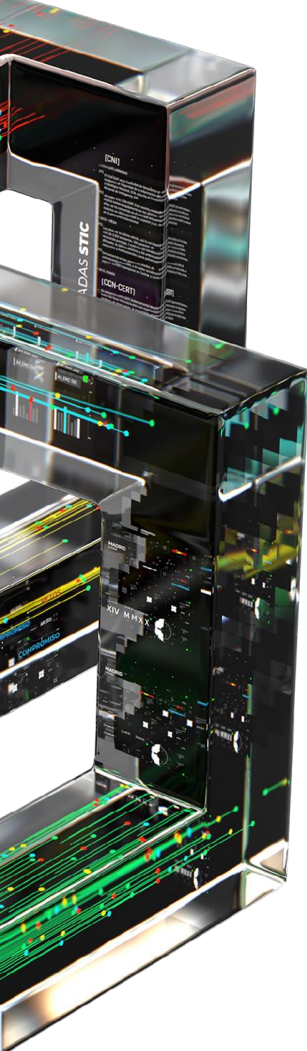
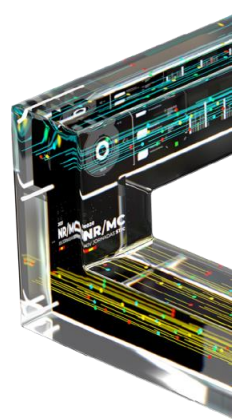
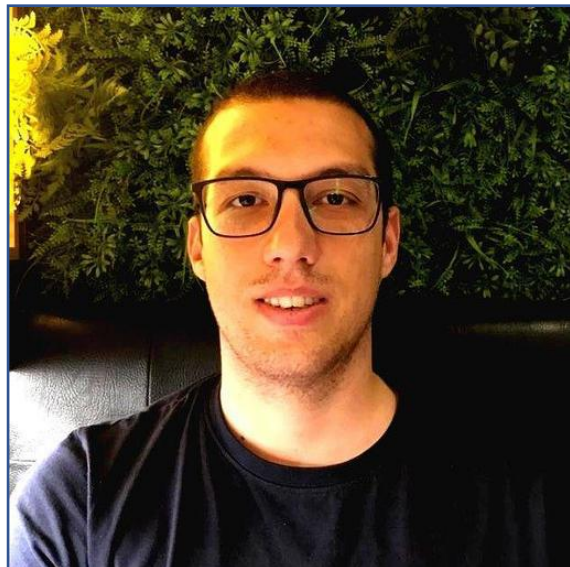




#XIVJORNADASCCNCERT

Bastionado de Kubernetes





**Ángel Barrera
Sánchez**

SIGHUP s.r.l

angel@sighup.io



Borja López Piñan

Aplazame S.L.

borja.lopez@aplazame.com



Jesús Marín García

Entelgy Innotec Security

jesus.marin@innotec.security

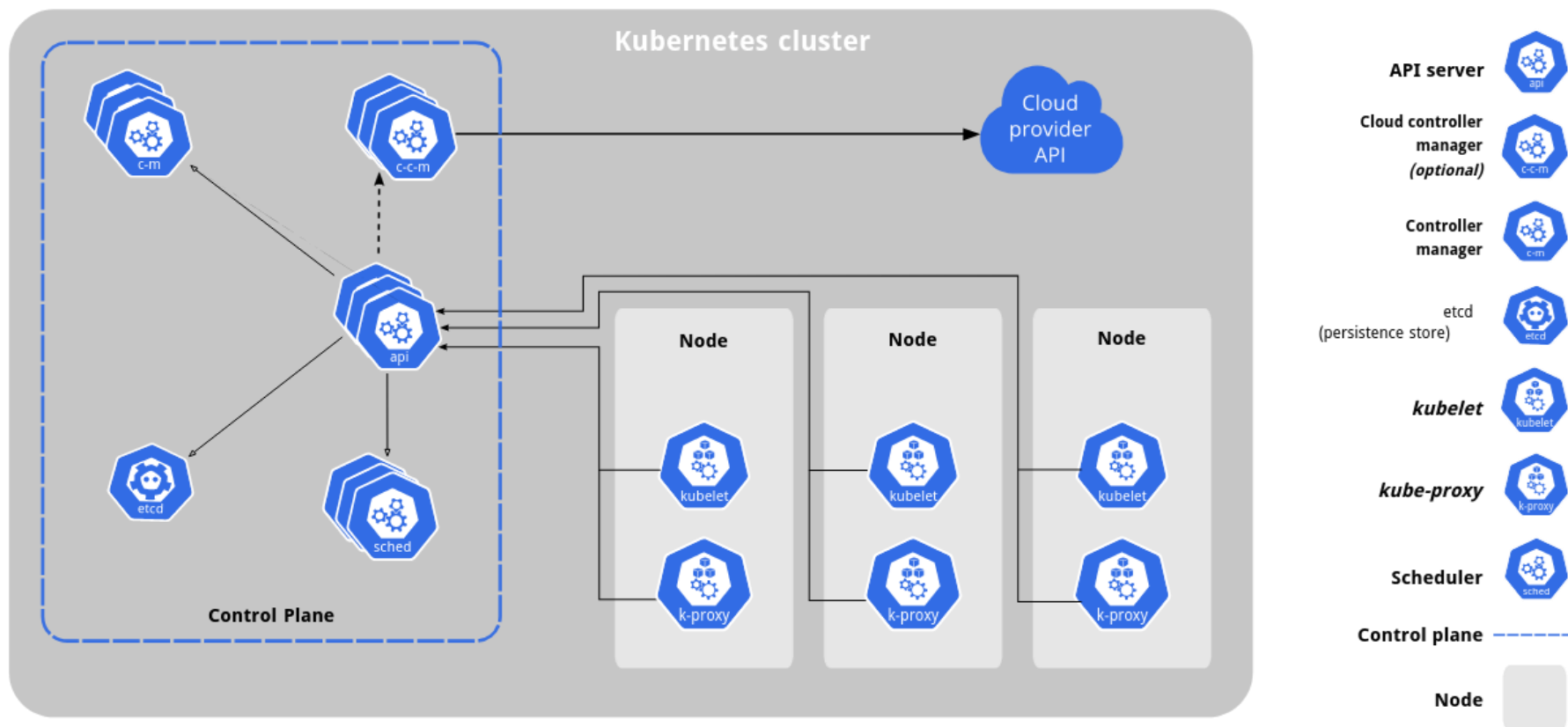
Bastionado de Kubernetes

Agenda

- **Kubernetes**
 - Componentes
 - Networking
 - Vectores de ataque
- **Cambio de paradigma**
 - Infraestructura
 - Seguridad
- **Ejemplos Prácticos**
 - Configuración de Kubelet
 - Configuración de etcd
 - Configuración de API Server
 - Tipos de autenticación
 - OIDC
 - ServiceAccounts
 - Tipos de autorización
 - RBAC
- Definición de imágenes de contenedores
 - Vulnerabilidades
 - Linting
 - Firmado (notary)
 - Comprobación de firma en Kubernetes (OPA Notary Connector)
- Políticas de seguridad en Kubernetes
 - Análisis y admisión. (Gatekeeper y Gatekeeper Policy Manager)
 - Escaner (kube-scan)
- Gestión de secretos
 - En repositorio (shhgit live, gitleaks)
 - Kubernetes Secrets
 - Cifrado etcd
- Network Policies
- Service Mesh

Kubernetes

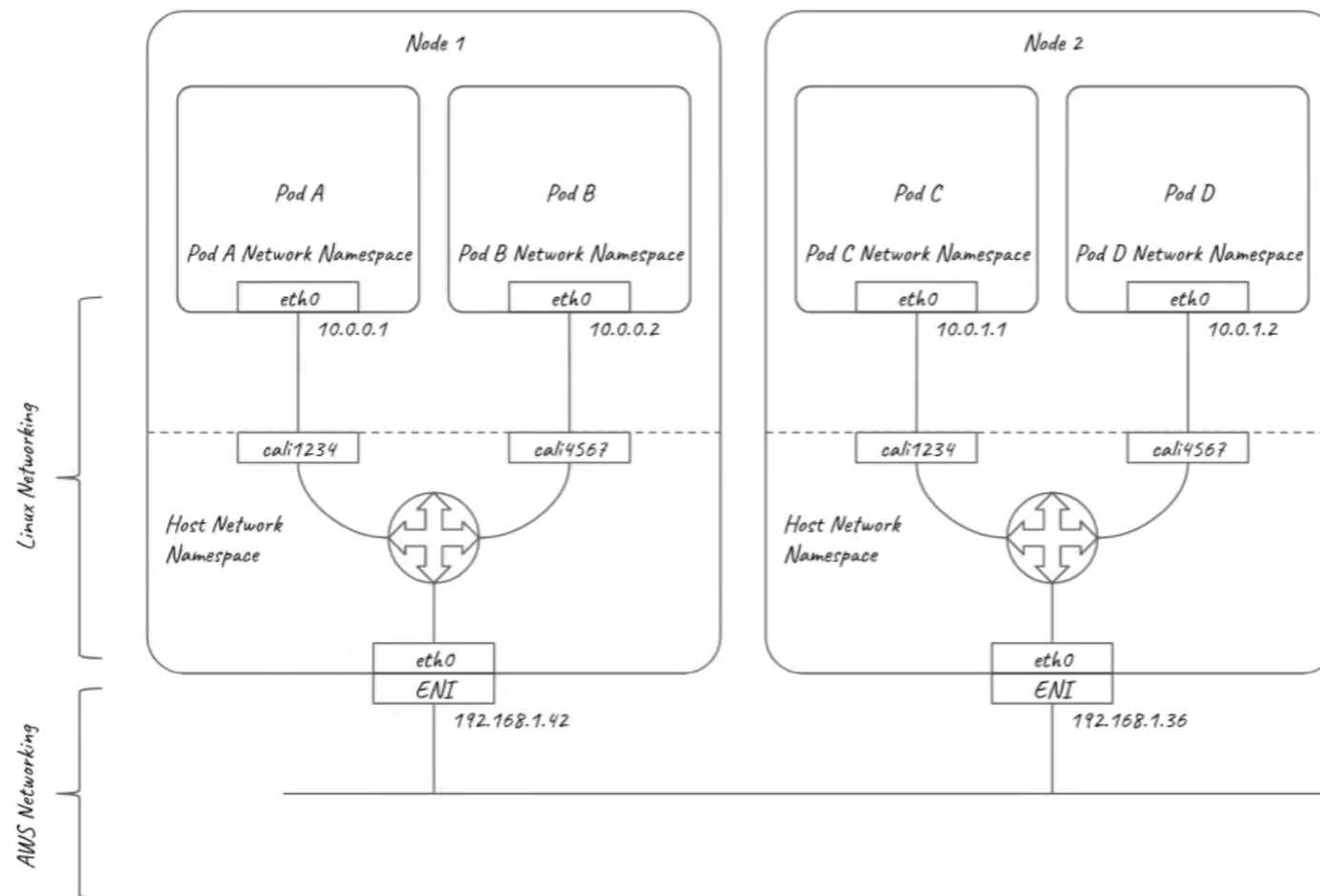
Componentes



Fuente: <https://kubernetes.io/docs/concepts/overview/components>

Kubernetes

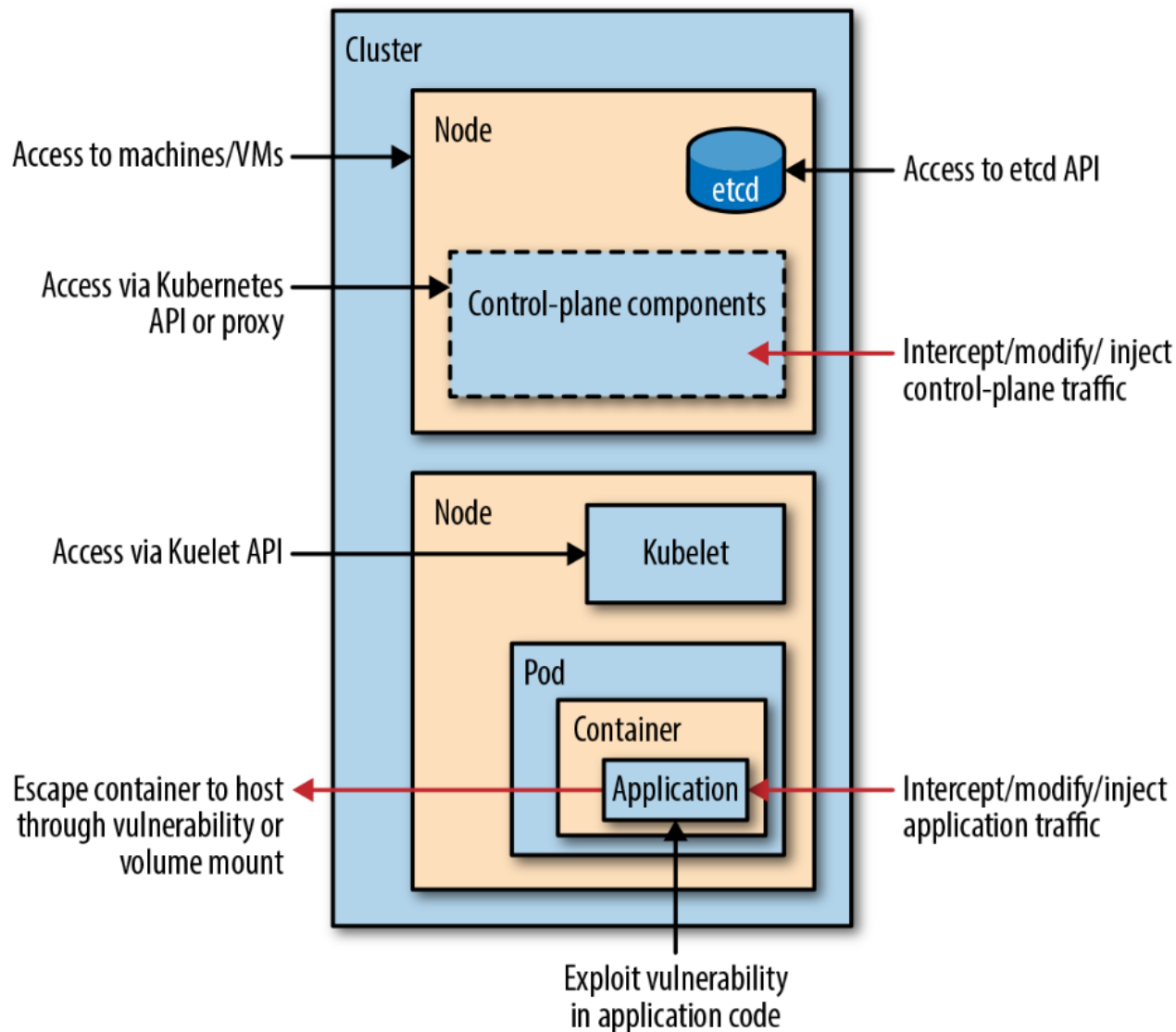
Networking



Fuente: <https://www.youtube.com/watch?v=J1VbZR7j4sl>

Kubernetes

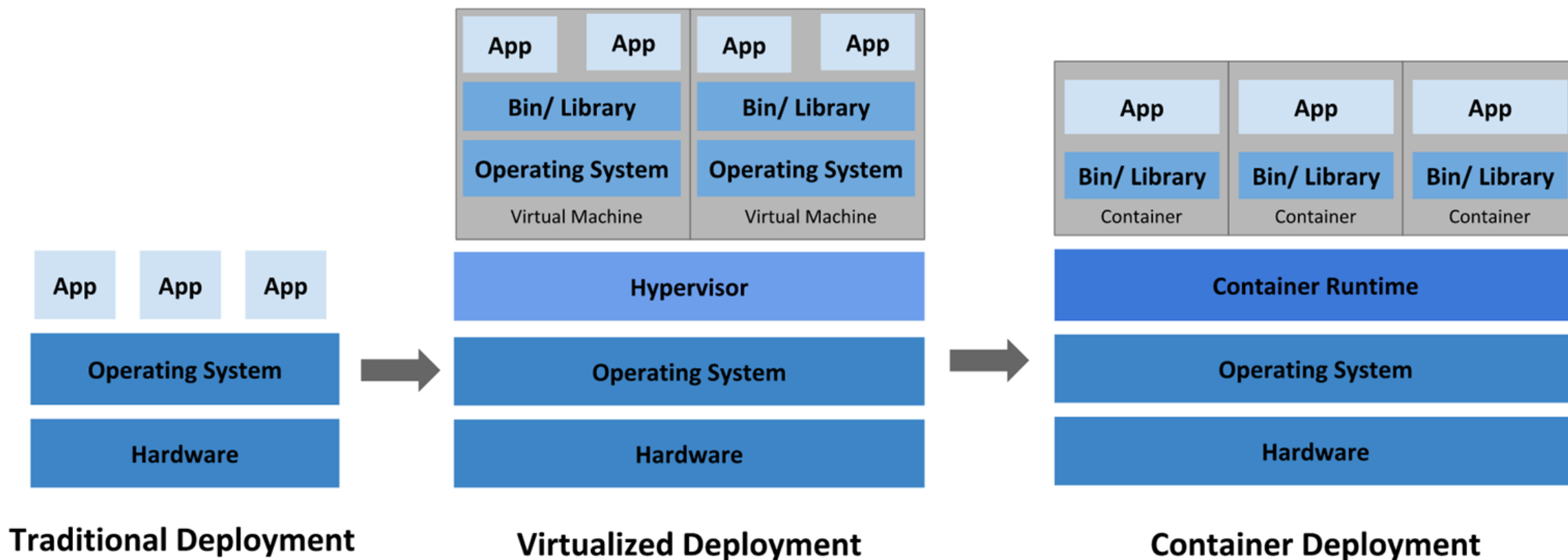
Vectores de ataque



Fuente: Kubernetes Security, Liz Rice and Michael Hausenblas, O'Reilly Media

Cambio de paradigma

Infraestructura



Fuente: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Cambio de paradigma

Infraestructura

Paradigma clásico	Kubernetes
- Infraestructura mutable - Tratada como una mascota - Aislamiento completo - Infraestructura estática	- Contenedores inmutables - Tratado como ganado - Aislamiento a nivel de namespace - Infraestructura dinámica

Ejemplos Prácticos

Kubelet

```
apiVersion: kubelet.config.k8s.io/v1beta1
address: 192.168.1.10
readOnlyPort: 0
authentication:
  anonymous:
    enabled: false
  webhook:
    cacheTTL: 0s
    enabled: true
  x509:
    clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionHard:
  imagefs.available: 0%
  nodefs.available: 0%
  nodefs.inodesFree: 0%
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageGCHighThresholdPercent: 100
imageMinimumGCAge: 0s
kind: KubeletConfiguration
logging: {}
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
# rotateCertificates: false
rotateCertificates: true
runtimeRequestTimeout: 0s
```

Ejemplos Prácticos

etcd

```

apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/etcd.advertise-client-urls: https://172.18.0.2:2379
  creationTimestamp: null
  labels:
    component: etcd
    tier: control-plane
  name: etcd
  namespace: kube-system
spec:
  containers:
    - command:
      - etcd
      - --advertise-client-urls=https://172.18.0.2:2379
      - --cert-file=/etc/kubernetes/pki/etcd/server.crt
      - --client-cert-auth=true
      - --cipher-suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      - --auto-tls=false
      - --peer-auto-tls=false
      - --data-dir=/var/lib/etcd
      - --initial-advertise-peer-urls=https://172.18.0.2:2380
      - --initial-cluster=kind-control-plane=https://172.18.0.2:2380
      - --key-file=/etc/kubernetes/pki/etcd/server.key
      - --listen-client-urls=https://127.0.0.1:2379,https://172.18.0.2:2379
      - --listen-metrics-urls=http://127.0.0.1:2381
      - --listen-peer-urls=https://172.18.0.2:2380
      - --name=kind-control-plane
      - --peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt
      - --peer-client-cert-auth=true
      - --peer-key-file=/etc/kubernetes/pki/etcd/peer.key
      - --peer-trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
      - --snapshot-count=10000
      - --trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt
    image: k8s.gcr.io/etcd:3.4.13-0
  
```

Ejemplos Prácticos

API Server

--enable-admission-control=...,NodeRestriction,SecurityContextDeny,PodSecurityPolicy,AlwaysPullImages

```
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=172.18.0.2
    - --authorization-mode=Node,RBAC #####
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-control=NamespaceLifecycle,LimitRanger,ServiceAccount,TaintNodesByCondition,Priority,DefaultTolerationSeconds,DefaultSt
    - --tls-cipher-suites=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 #
    - --enable-bootstrap-token-auth=true
    - --audit-log-path=/var/log/kubernetes/apiserver.log #####
    - --anonymous-auth=false #####
    - --allow-privileged=false #####
    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --insecure-port=0 #####
    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
    - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
    - --requestheader-allowed-names=front-proxy-client
    - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
    - --requestheader-extra-headers-prefix=X-Remote-Extra-
    - --requestheader-group-headers=X-Remote-Group
    - --requestheader-username-headers=X-Remote-User
    - --secure-port=6443
    - --service-account-key-file=/etc/kubernetes/pki/sa.pub
    - --service-cluster-ip-range=10.96.0.0/12
    - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
    - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
```

Ejemplos Prácticos

Tipos de Autenticación

X509 Client Certs	1. Activarlo usando: --client-ca-file=SOMEFILE en el API Server 2. Generas una private key y un CSR: <pre>\$ openssl genrsa -out user.key 4096 && openssl req -new -key user.key -out user.csr</pre> 1. Generas un CertificateSigningRequest con el csr anterior y lo apruebas: <pre>\$ kubectl get csr</pre> <pre>\$ kubectl certificate approve user</pre>
Static Token File	Activarlo usando el flag: --token-auth-file=SOMEFILE en el API Server.
Bootstrap Tokens	Activarlo usando el flag: --enable-bootstrap-token-auth en el API Server.
Webhook Token Authentication	--authentication-token-webhook-config-file , --authentication-token-webhook-cache-ttl , --authentication-token-webhook-version
Anonymous Authentication (¡¡¡CUIDADO!!!)	--anonymous-auth=true Desactivada por defecto.

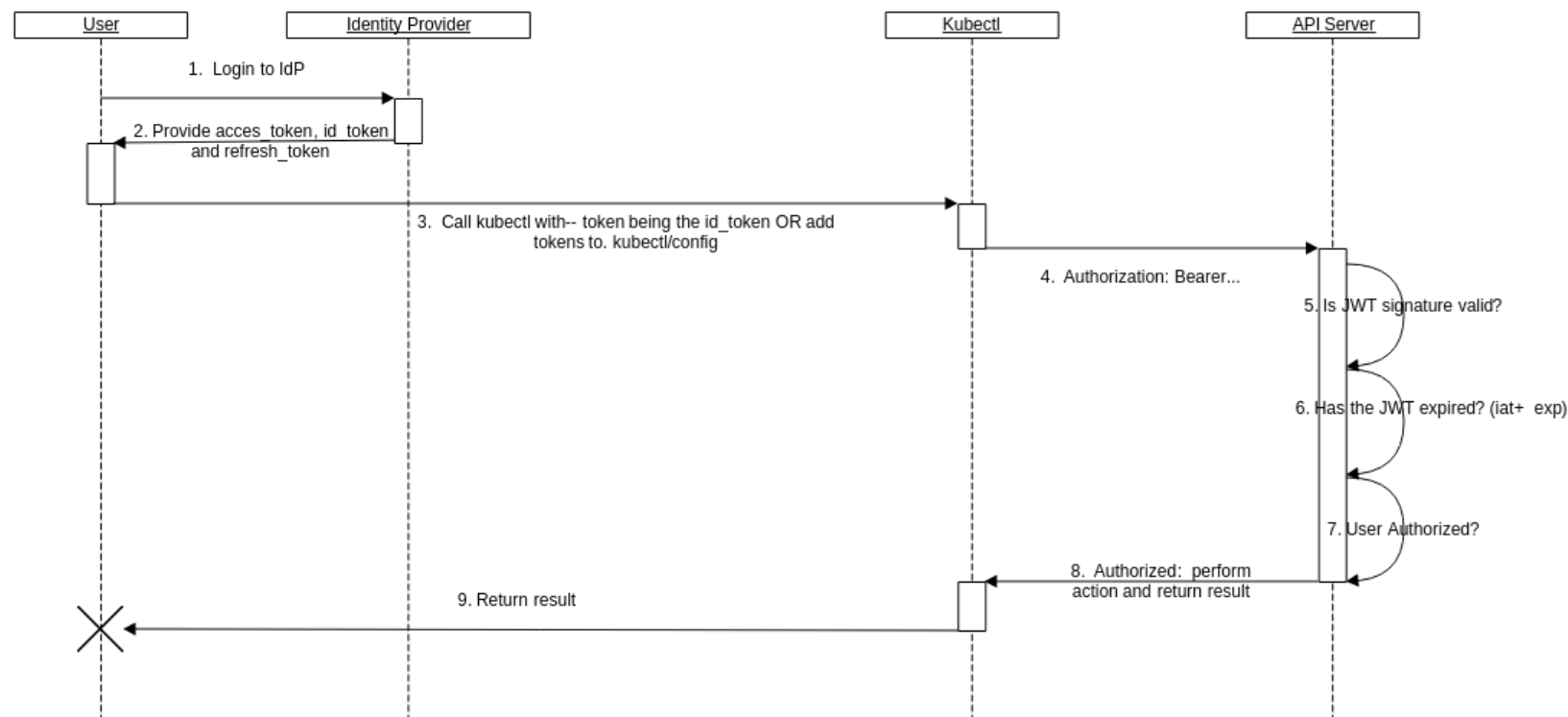
Ejemplos Prácticos

OIDC Autenticación

```
--oidc-issuer-url=https://dex.example.com:32000
--oidc-client-id=example-app
--oidc-ca-file=/etc/ssl/certs/openid-ca.pem
--oidc-username-claim=email
--oidc-groups-claim=groups
```



kubelogin



Fuente: <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>

Ejemplos Prácticos

Service Accounts

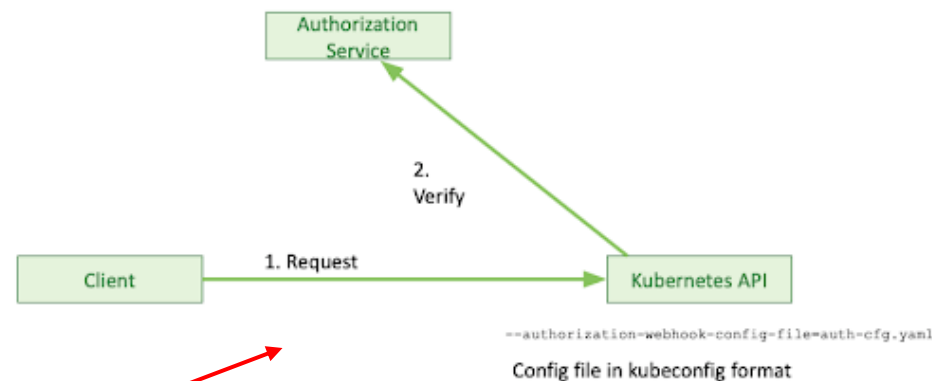
```

svc.yaml
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4   name: svc-test
5 ---
6 apiVersion: v1
7 kind: ServiceAccount
8 metadata:
9   name: svc-test-no-mount
10 automountServiceAccountToken: false
  
```

- Las cuentas de servicio como su nombre indica **son para los procesos**.
- Se debe tener en cuenta los permisos que se asigna a la cuenta de servicio.
- Por defecto **monta el token de servicio** en cada pod, si realmente no se necesita se puede usar el valor que se ve en el ejemplo.
- **Borrar el secret** de una service account para revocarlo.

Ejemplos Prácticos

Tipos de Autorización

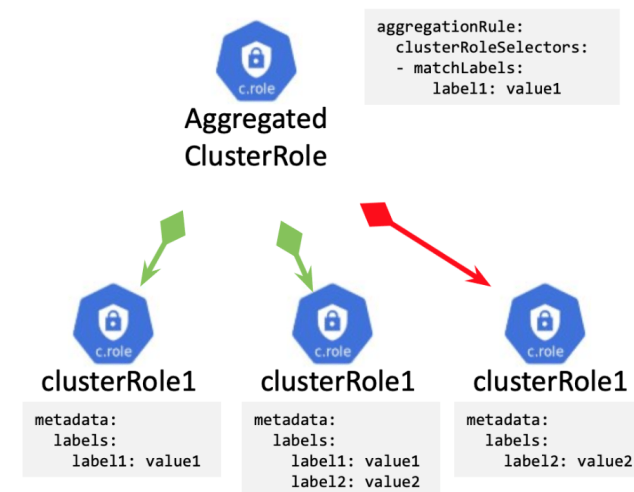
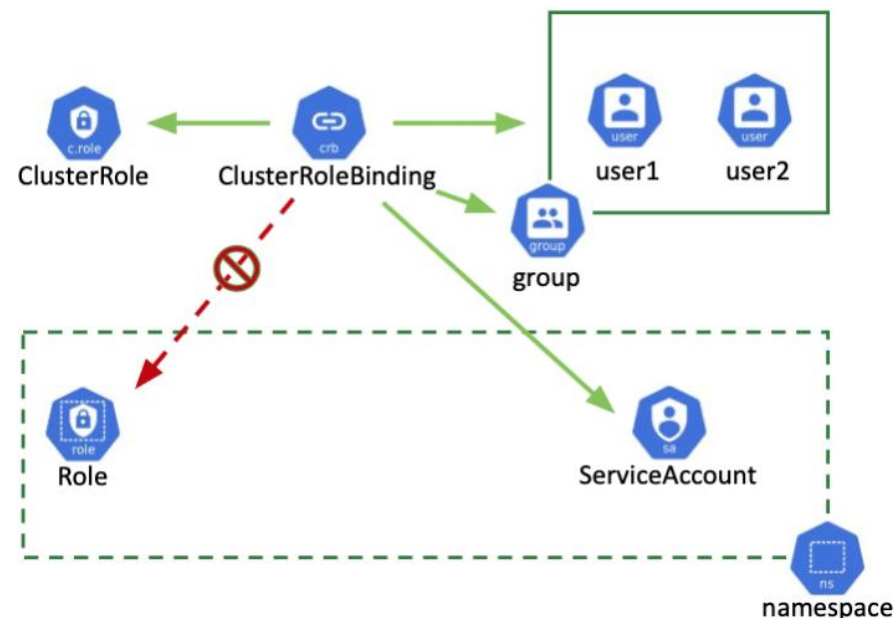
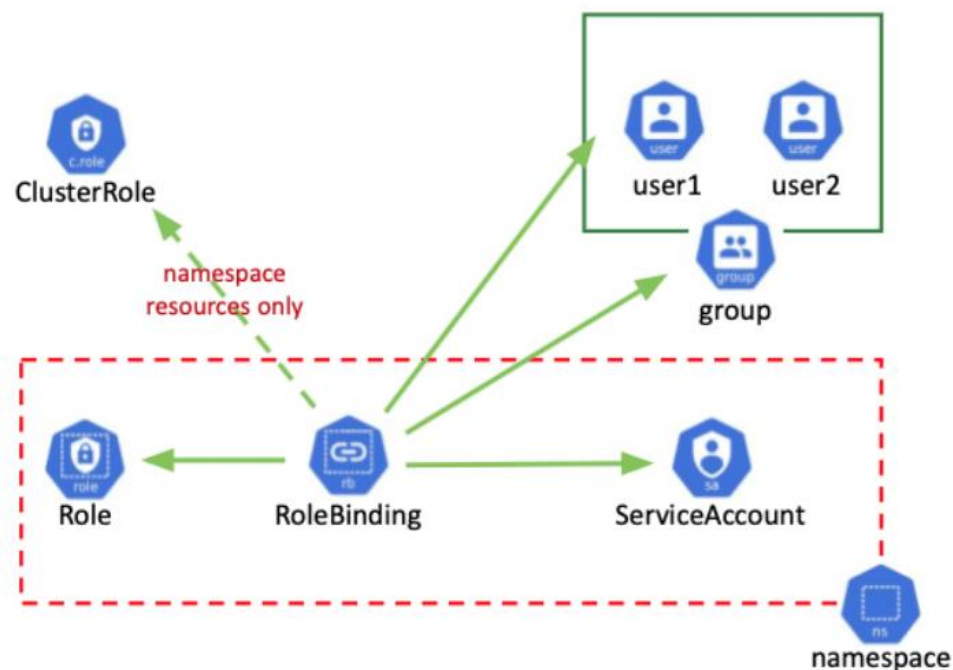


Node	kubelet authorization.
ABAC (Deprecated)	Autenticación basada en atributos.
RBAC	Autenticación basada en roles (verbos, recursos).
Webhook	Autenticación mediante llamada a un webhook externo.
AlwaysAllow(Default)/AlwaysDeny	Su propio nombre lo indica.

```
$ kubectl auth can-i create deployments --namespace dev
(yes|no)
```

Ejemplos Prácticos

Autorización RBAC



Fuente: <https://www.cncf.io/blog/2020/08/28/kubernetes-rbac-101-authorization/>

Ejemplos Prácticos

RBAC Ejemplos

ROLE

```
[node1 ~]$ kubectl create role pod-reader --verb=get --verb=list --verb=watch --resource=pods
role.rbac.authorization.k8s.io/pod-reader created
[node1 ~]$ █
```

CLUSTERROLE

```
[node1 ~]$ kubectl create clusterrole clusterrole-reader --verb=get,list,watch --resource=pods
clusterrole.rbac.authorization.k8s.io/clusterrole-reader created
[node1 ~]$ █
```

ROLEBINDING

```
[node1 ~]$ kubectl create rolebinding bob-admin-binding --clusterrole=admin --user=bob --namespace=default
rolebinding.rbac.authorization.k8s.io/bob-admin-binding created
[node1 ~]$ █
```

CLUSTERROLEBINDING

```
[node1 ~]$ kubectl create clusterrolebinding root-cluster-admin-binding --clusterrole=cluster-admin --user=root
clusterrolebinding.rbac.authorization.k8s.io/root-cluster-admin-binding created
[node1 ~]$ █
```

Ejemplos Prácticos

GESTIÓN RBAC

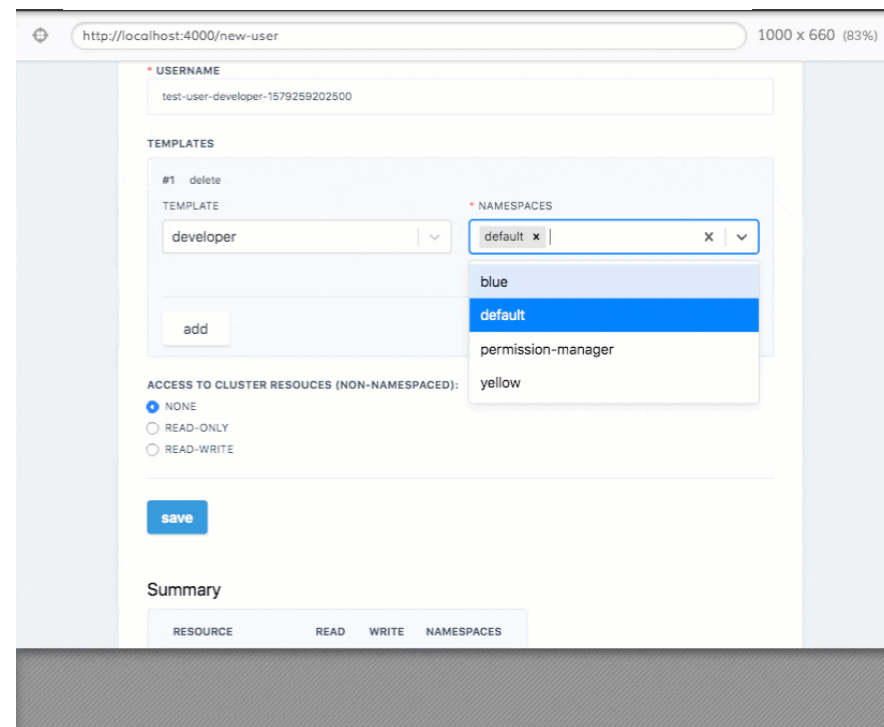
RBAC MANAGER

```
apiVersion: rbacmanager.reactiveops.io/v1beta1
kind: RBACDefinition
metadata:
  name: joe-access
rbacBindings:
  - name: joe
    subjects:
      - kind: User
        name: joe@example.com
    roleBindings:
      - namespace: api
        clusterRole: view
      - namespace: web
        clusterRole: edit
```



```
+-----+
|Risky Containers|
+-----+
| Priority | PodName | ContainerName | ServiceAccountNamespace | ServiceAccountName |
+-----+
| CRITICAL | risky-pod | nginx | default | risky-sa |
+-----+
```

Permission manager



The screenshot shows the 'new-user' page of the Permission manager. The URL is `http://localhost:4000/new-user`. The form includes a 'USERNAME' field with the value 'test-user-developer-1579259202500'. Below it is a 'TEMPLATES' section with a dropdown menu showing 'developer'. To the right is a 'NAMESPACES' dropdown menu with a list of namespaces: 'blue', 'default' (highlighted), 'permission-manager', and 'yellow'. Below these is a section for 'ACCESS TO CLUSTER RESOURCES (NON-NAMESPACED):' with radio buttons for 'NONE' (selected), 'READ-ONLY', and 'READ-WRITE'. At the bottom is a 'save' button and a 'Summary' section with tabs for 'RESOURCE', 'READ', 'WRITE', and 'NAMESPACES'.

Repositorios:

<https://github.com/FairwindsOps/rbac-manager>
<https://github.com/sighupio/permission-manager>
<https://github.com/cyberark/KubiScan>

Ejemplos Prácticos

Contenedores: Buenas prácticas

With great power comes great responsibility



podman



buildah



Kaniko

```
# Use a secure base image
FROM centos:centos7.6.1810

# Most generic layers first
LABEL maintainer "Bob Alice <bob.alice@example.org>"

# Input data
ARG SKOPEO_VERSION="0.1.37"

# Change to ROOT user to do maintenance/install tasks
USER root

# Install ONLY required packages
RUN yum -y --setopt=tsflags=nodocs install \
    git

# Install the application
RUN yum -y --setopt=tsflags=nodocs install \
    skopeo-${SKOPEO_VERSION}

# Run container as non-root user
RUN useradd --no-create-home --shell /bin/sh nroot
USER nroot
```

Repositorios:

<https://podman.io/>

<https://buildah.io/>

<https://github.com/GoogleContainerTools/kaniko>

Ejemplos Prácticos

Contenedores: Análisis de buenas prácticas



```
> cat Dockerfile
FROM debian
RUN export node_version="0.10" \
&& apt-get update && apt-get -y install nodejs="$node_verion"
COPY package.json usr/src/app
RUN cd /usr/src/app \
&& npm install node-static

EXPOSE 80000
CMD ["npm", "start"]%

> hadolint Dockerfile
Dockerfile:1 DL3006 Always tag the version of an image explicitly
Dockerfile:2 SC2154 node_verion is referenced but not assigned (did you mean 'node_version'?).
Dockerfile:2 DL3009 Delete the apt-get lists after installing something
Dockerfile:2 DL3015 Avoid additional packages by specifying `--no-install-recommends`
Dockerfile:5 DL3003 Use WORKDIR to switch to a directory
Dockerfile:5 DL3016 Pin versions in npm. Instead of `npm install <package>` use `npm install <package>@<version>`
Dockerfile:8 DL3011 Valid UNIX ports range from 0 to 65535
```

Ejemplos Prácticos

Contenedores: Análisis de vulnerabilidades



```
> trivy vulnerables/cve-2014-6271
2020-11-15T17:40:25.986+0100    WARN    You should avoid using the :latest tag as it is cached. You need to specify '--clear-cache' option when :latest image i
2020-11-15T17:40:28.095+0100    INFO    Detecting Debian vulnerabilities...
2020-11-15T17:40:28.097+0100    WARN    This OS version is no longer supported by the distribution: debian 7.11
2020-11-15T17:40:28.097+0100    WARN    The vulnerability detection may be insufficient because security updates are not provided

vulnerables/cve-2014-6271 (debian 7.11)
=====
Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 1, CRITICAL: 1)

+-----+-----+-----+-----+-----+-----+
| LIBRARY | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION | TITLE |
+-----+-----+-----+-----+-----+-----+
| bash    | CVE-2014-6271    | CRITICAL | 4.2+dfsg-0.1      | 4.2+dfsg-0.1+deb7u1 | bash: specially-crafted environment variables can be used to inject shell commands |
+-----+-----+-----+-----+-----+-----+
|          | CVE-2014-7169    | HIGH     |                   | 4.2+dfsg-0.1+deb7u3 | bash: code execution via specially-crafted environment (Incomplete fix for CVE-2014-6271) |
+-----+-----+-----+-----+-----+-----+
```

Ejemplos Prácticos

Firma de imágenes: Notary

```
$ notary -s https://notary.docker.io -d ~/.docker/trust list docker.io/library/alpine
```

NAME	DIGEST	SIZE (BYTES)	ROLE
2.6	e9cec9aec697d8b9d450edd32860ecd363f2f3174c8338beb5f809422d182c63	1374	targets
2.7	9f08005dff552038f0ad2f46b8e65ff3d25641747d3912e3ea8da6785046561a	1374	targets
3.1	e876b57b2444813cd474523b9c74aacacc238230b288a22bccece9caf2862197	1374	targets
3.2	4a8c62881c6237b4c1434125661cddf09434d37c6ef26bf26bfaef0b8c5e2f05	1374	targets
3.3	2d4f890b7eddb390285e3afea9be98a078c2acd2fb311da8c9048e3d1e4864d3	1374	targets
edge	878c1b1d668830f01c2b1622ebf1656e32ce830850775d26a387b2f11f541239	1374	targets
latest	24a36bbc059b1345b7e8be0df20f1b23caa3602e85d42fff7ecd9d0bd255de56	1377	targets

Ejemplos Prácticos

Firma de imágenes: OPA Notary Connector

What it does

It validate images' signatures against notary instances, rejecting untrusted images and setting the `sha256` to trusted images in order to pin the exact version. The enforcing is configurable per namespace and registry.

Repositorio:

<https://github.com/sighupio/opa-notary-connector>

Ejemplos Prácticos

Despliegue

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-readonlyrootfilesystem-allowed
  labels:
    app: nginx-readonlyrootfilesystem
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      readOnlyRootFilesystem: true
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-forbidden-sysctls-disallowed
  labels:
    app: nginx-forbidden-sysctls
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      sysctls:
      - name: kernel.msgmax
        value: "65536"
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-privileged-allowed
  labels:
    app: nginx-privileged
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      privileged: false
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-host-filesystem
  labels:
    app: nginx-host-filesystem-disallowed
spec:
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - mountPath: /cache
      name: cache-volume
      readOnly: true
  volumes:
  - name: cache-volume
    hostPath:
      path: /tmp # directory location on host
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-proc-mount-disallowed
  labels:
    app: nginx-proc-mount
spec:
  containers:
  - name: nginx
    image: nginx
    securityContext:
      procMount: Unmasked #Default
```

Fuente: <https://github.com/open-policy-agent/gatekeeper-library/tree/master/library/pod-security-policy>

Ejemplos Prácticos

OPA Gatekeeper & Gatekeeper Policy Manager



Repositorio:

<https://github.com/open-policy-agent/gatekeeper>

<https://github.com/sighupio/gatekeeper-policy-manager>

Ejemplos Prácticos

kube-scan



Repositorio:

<https://github.com/octarinesec/kube-scan>

#XIVJORNADASCCNCERT

Ejemplos Prácticos

Gestión de secretos: Repositorios



shhgit live! v0.4

101 matches 0 filters

- Log file 79
- Username and password in 7
- High entropy string 5
- SQLite3 database file 3
- Potential private key (.pfx) 2
- PHP configuration file 2
- Environment configuration file 1
- Facebook Client ID 1
- Potential Ruby On Rails data 1

☒ Interesting file extensions
 ☒ High entropy strings
 ☐ Notify on match


 @darkp0rt
  blog
 Connected

NEW! [shhgit-live](#) version 0.4 - blazing fast and 30 new signatures.

	Found	Signature Name	Matches	File	★
	03:14:50	Log file	—	./metadata/plugins/org.eclipse.m2e.logback.configuration/0.log	0
	03:14:50	Log file	—	./metadata/log	0
	03:14:49	NPM configuration file	—	./npmrc	0
	03:14:47	High entropy string	<code>SECRET_KEY = '%710m*zic)#0u((qugw#1@e^ty!c)9j04956v@ly(_86n\$rg)h'</code>	/wiki/settings.py	0
	03:14:47	Django configuration file	—	/wiki/settings.py	0
	03:14:47	SQLite3 database file	—	/db.sqlite3	0
	03:14:47	Log file	—	/log/development.log	0
	03:14:47	SQLite3 database file	—	/db/test.sqlite3	0
	03:14:47	SQLite3 database file	—	/db/development.sqlite3	0

Ejemplos Prácticos

Gestión de secretos: Repositorios



```

~ > gitLeaks --repo=https://github.com/jesmg/secretsRepo.git -v --pretty
INFO[2020-11-16T20:25:46+01:00] cloning... https://github.com/jesmg/secretsRepo.git
Enumerating objects: 28, done.
Total 28 (delta 0), reused 0 (delta 0), pack-reused 28
{
  "line": " - awskey: AKIA0123456789012345",
  "lineNumber": 2,
  "offender": "AKIA0123456789012345",
  "commit": "a244e3d5de9c5823775047eb1fe6c638cf3b6273",
  "repo": "secretsRepo.git",
  "rule": "AWS Manager ID",
  "commitMessage": "secrets is back\n",
  "author": "Jesús Marín",
  "email": "jesus.marin@innotec.security",
  "file": "secrets.yaml",
  "date": "2019-06-18T10:19:19+02:00",
  "tags": "key, AWS",
  "operation": "addition"
}
{
  "line": " - mws: amzn.mws.01234567-0123-0123-0123-012345678901",
  "lineNumber": 12,
  "offender": "amzn.mws.01234567-0123-0123-0123-012345678901",
  "commit": "a244e3d5de9c5823775047eb1fe6c638cf3b6273",
  "repo": "secretsRepo.git",
  "rule": "AWS MWS key",
  "commitMessage": "secrets is back\n",

```

```

  "line": " mailchimp: mc.sadf:\01234567890123456789012345678901-us01",
  "lineNumber": 24,
  "offender": "mailchimp: mc.sadf:\01234567890123456789012345678901-us01",
  "commit": "e466b5fb3de8043656fd06a7ac68bace30a02fa0",
  "repo": "secretsRepo.git",
  "rule": "MailChimp API key",
  "commitMessage": "Update secrets.yaml",
  "author": "Jesús",
  "email": "jesmg@users.noreply.github.com",
  "file": "secrets.yaml",
  "date": "2019-06-18T10:09:21+02:00",
  "tags": "key, Mailchimp",
  "operation": "addition"
}
{
  "line": " - notsecret: ${-----BEGIN PRIVATE KEY-----}",
  "lineNumber": 33,
  "offender": "-----BEGIN PRIVATE KEY-----",
  "commit": "e466b5fb3de8043656fd06a7ac68bace30a02fa0",
  "repo": "secretsRepo.git",
  "rule": "Asymmetric Private Key",
  "commitMessage": "Update secrets.yaml",
  "author": "Jesús",
  "email": "jesmg@users.noreply.github.com",
  "file": "secrets.yaml",
  "date": "2019-06-18T10:09:21+02:00",
  "tags": "key, AsymmetricPrivateKey",
  "operation": "addition"
}

```

WARN[2020-11-16T20:25:47+01:00] 94 leaks detected. 13 commits scanned in 22 milliseconds 333 microseconds

Ejemplos Prácticos

Gestión de secretos: Kubernetes secrets

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-basic-auth
type: kubernetes.io/basic-auth
stringData:
  username: admin
  password: t0p-Secret
```

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
    - name: mypod
      image: redis
      volumeMounts:
        - name: foo
          mountPath: "/etc/foo"
          readOnly: true
      volumes:
        - name: foo
          secret:
            secretName: mysecret
```

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
spec:
  containers:
    - name: mycontainer
      image: redis
      env:
        - name: SECRET_USERNAME
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: username
        - name: SECRET_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
      restartPolicy: Never
```

Ejemplos Prácticos

Gestión de secretos: Cifrado en etcd

```
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
    - secrets
  providers:
    - aescbc:
      keys:
        - name: key1
          secret: <BASE 64 ENCODED SECRET>
      identity: {}
```

```
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
    - secrets
  providers:
    - kms:
      name: myKmsPlugin
      endpoint: unix:///tmp/socketfile.sock
      cachesize: 100
      timeout: 3s
      identity: {}
```

Name	Encryption	Strength	Speed	Key Length
identity	None	N/A	N/A	N/A
aescbc	AES-CBC with PKCS#7 padding	Strongest	Fast	32-byte
secretbox	XSalsa20 and Poly1305	Strong	Faster	32-byte
aesgcm	AES-GCM with random nonce	Must be rotated every 200k writes	Fastest	16, 24, or 32-byte
kms	Uses envelope encryption scheme: Data is encrypted by data encryption keys (DEKs) using AES-CBC with PKCS#7 padding, DEKs are encrypted by key encryption keys (KEKs) according to configuration in Key Management Service (KMS)	Strongest	Fast	32-bytes

Fuente: <https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/>

Ejemplos Prácticos

Gestión de secretos: ¡¡CUIDADO!!!

Un usuario con **permisos para crear pods** en un namespace, puede **leer los secretos** del namespace.

Ya que:

- Puede crear pods que corren bajo cualquier service account con mayores permisos
- Puede crear pods capaces de acceder a secretos que el usuario no puede leer

Fuente: <https://kubernetes.io/docs/concepts/configuration/secret/>

Ejemplos Prácticos

Network Policies

```

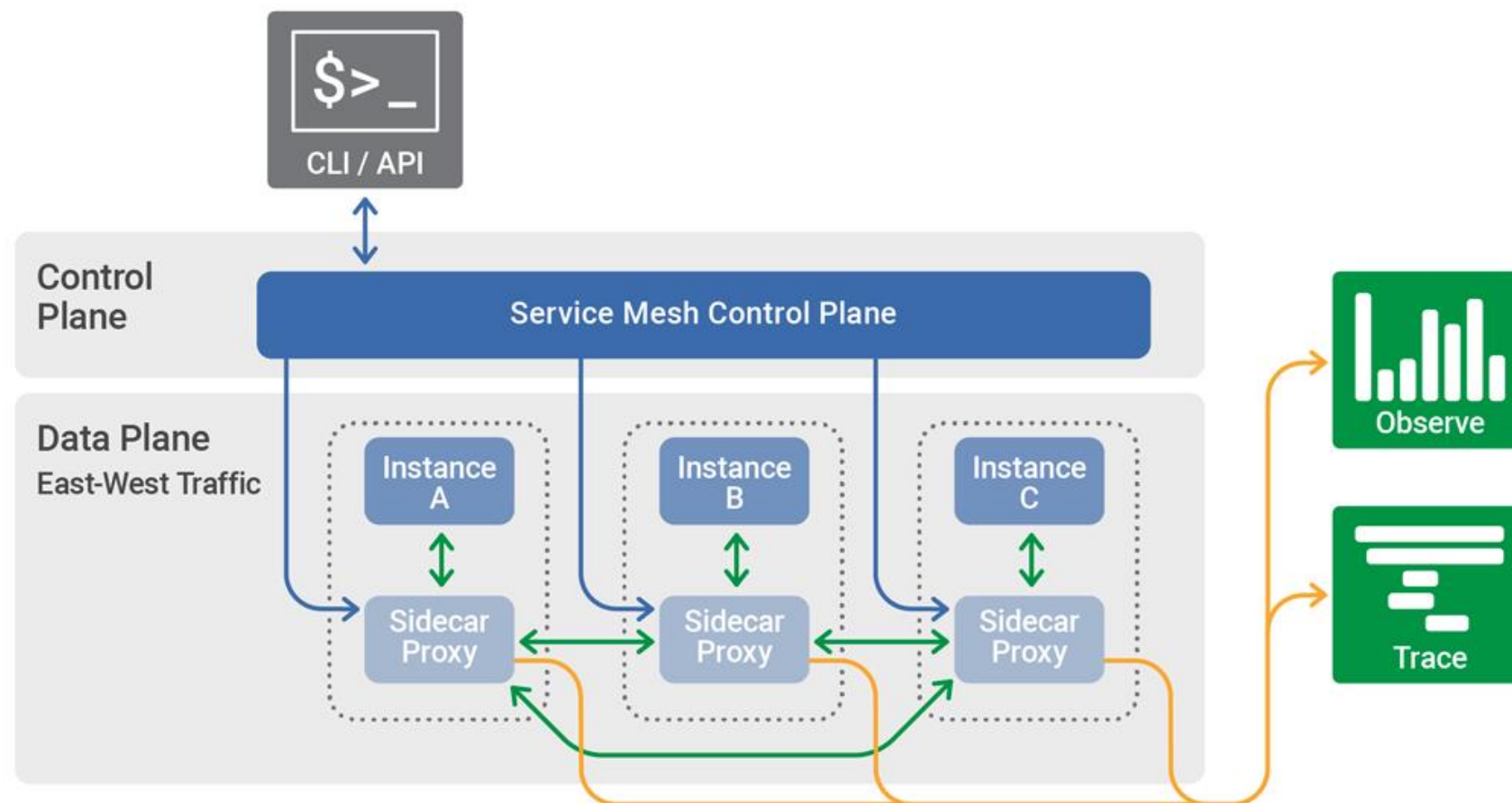
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny-all
spec:
  podSelector: {}
  policyTypes:
  - Ingress
  - Egress
  
```

```

1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: test-network-policy
5    namespace: default
6  spec:
7    podSelector:
8      matchLabels:
9        role: db
10   policyTypes:
11     - Ingress
12     - Egress
13   ingress:
14     - from:
15       - ipBlock:
16         cidr: 172.17.0.0/16
17         except:
18         - 172.17.1.0/24
19       - namespaceSelector:
20         matchLabels:
21           project: myproject
22     - podSelector:
23       matchLabels:
24         role: frontend
25   ports:
26     - protocol: TCP
27       port: 6379
28   egress:
29     - to:
30       - ipBlock:
31         cidr: 10.0.0.0/24
32     ports:
33       - protocol: TCP
34         port: 5978
  
```

Ejemplos Prácticos

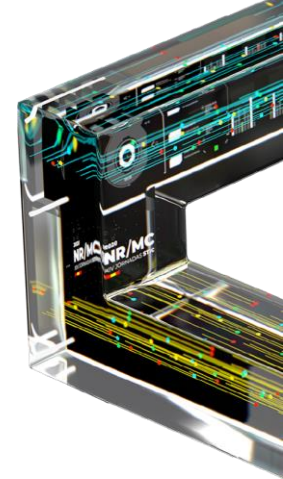
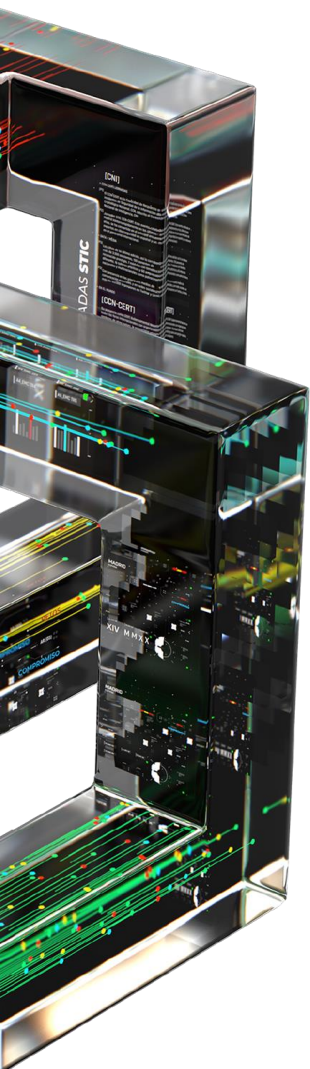
Service Mesh



Fuente: <https://www.nginx.com/blog/what-is-a-service-mesh/>

REFERENCIAS:

- <https://kubernetes.io/docs/concepts/overview/components>
- <https://www.youtube.com/watch?v=J1VbZR7j4sl>
- *Kubernetes Security, Liz Rice and Michael Hausenblas, O'Reilly Media*
- <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>
- <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>
- <https://kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/>
- <https://kubernetes.io/docs/reference/access-authn-authz/authorization/>
- <https://www.cncf.io/blog/2020/08/28/kubernetes-rbac-101-authorization/>
- <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>
- <https://github.com/FairwindsOps/rbac-manager>
- <https://github.com/sighupio/permission-manager>
- <https://github.com/cyberark/KubiScan>
- <https://podman.io/>
- <https://buildah.io/>
- <https://github.com/GoogleContainerTools/kaniko>
- <https://github.com/hadolint/hadolint>
- <https://github.com/aquasecurity/trivy>
- https://docs.docker.com/notary/getting_started/
- <https://github.com/sighupio/opa-notary-connector>
- <https://github.com/open-policy-agent/gatekeeper-library/tree/master/library/pod-security-policy>
- <https://github.com/open-policy-agent/gatekeeper>
- <https://github.com/sighupio/gatekeeper-policy-manager>
- <https://github.com/octarinesec/kube-scan>
- <https://www.shhggit.com/>
- <https://github.com/zricethezav/gitleaks>
- <https://kubernetes.io/docs/concepts/configuration/secret/>
- <https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/>
- <https://kubernetes.io/docs/concepts/configuration/secret/>
- <https://kubernetes.io/docs/concepts/services-networking/network-policies/>
- <https://www.nginx.com/blog/what-is-a-service-mesh/>



MUCHAS GRACIAS

#XIVJORNADASCCNCERT