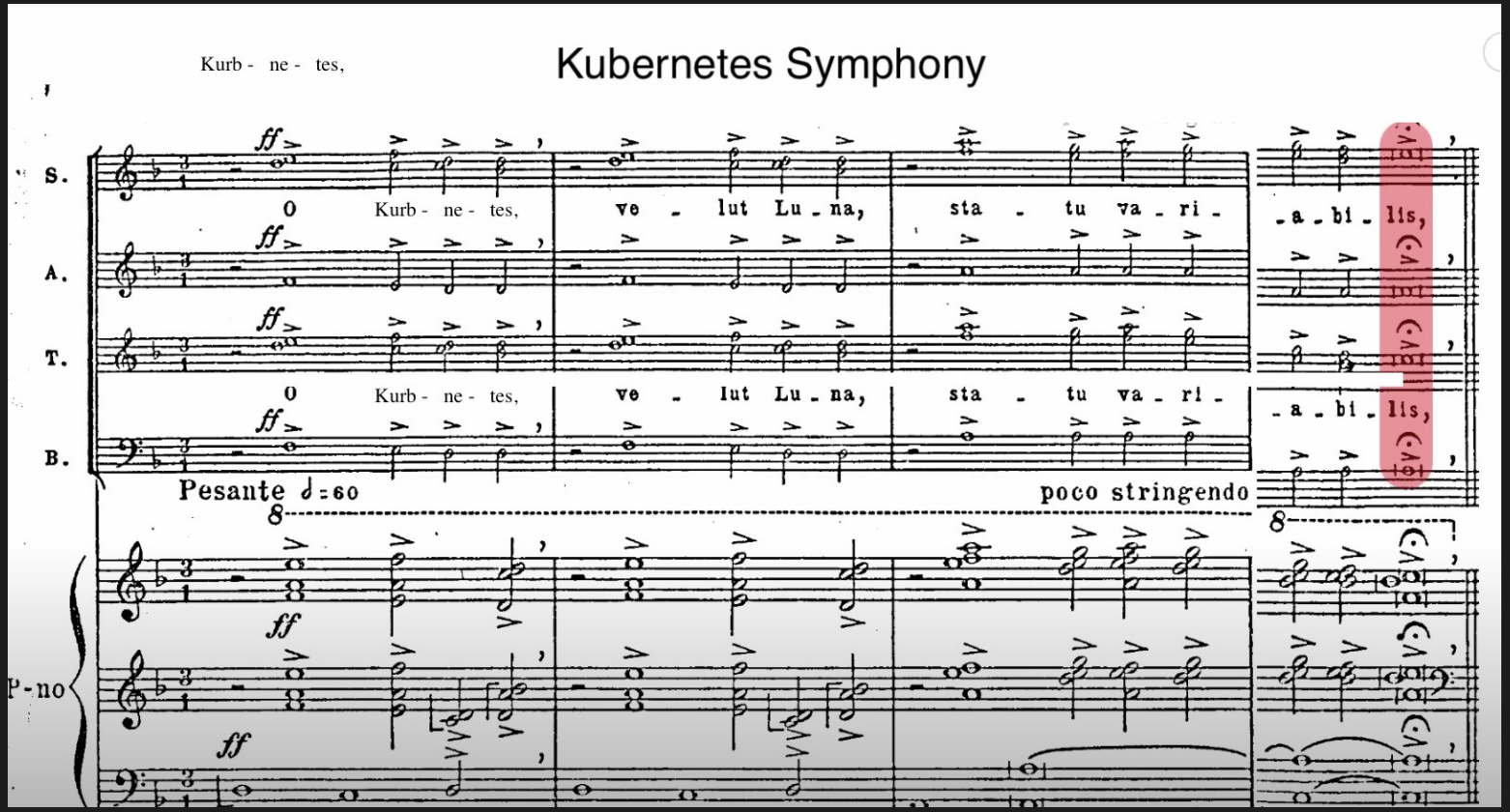
& C

D V P

Engineering

Kubernetes symphony



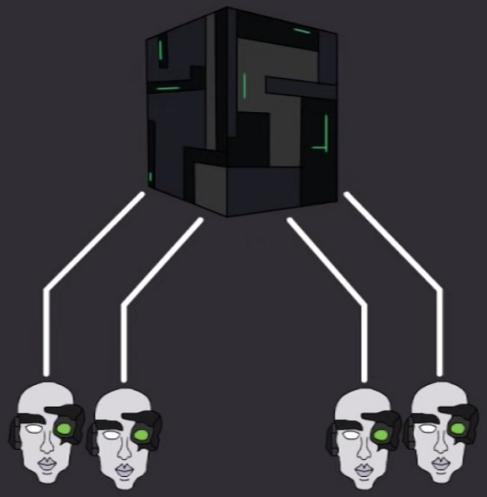
O κυβε - ρνή - της , Ve - lut Lu – na , sta – tu va – ri - a - bi - lis ,
O Kuber – ne - tes , Like the moon, You are changing unpredictably

Kubernetes securities from built-in features to open source tools

By .c, .e, .s

Contents

About k8s
k8s security
Demos

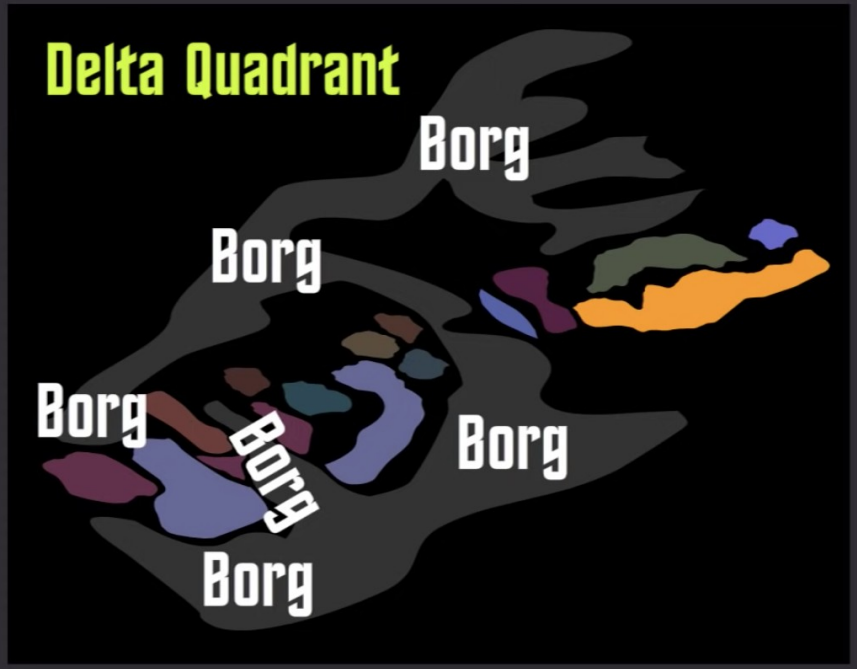


Collective Consciousness

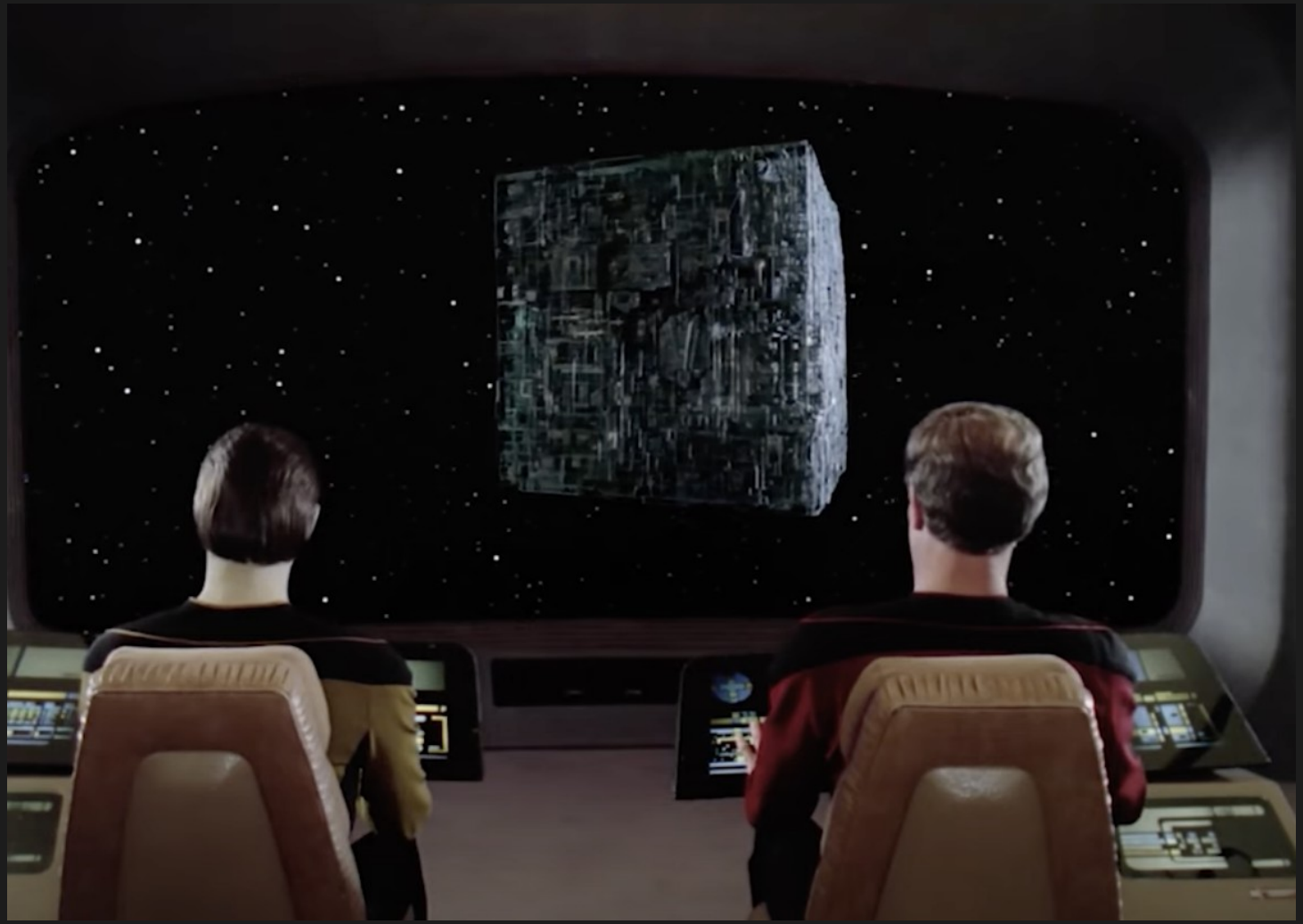
All Borg Minds Linked Together

Shared Thoughts and Memories

Acted as Unified Entity



borg



borg

Borg-Voyager Alliance
Voyager's Doctor Created Modified Nanoprobes to Fight Species 8472
Janeway Agreed to Work with the Borg in Exchange for Passage through their Space
Seven of Nine Represented the Borg
Borg Won the War
Voyager Escaped Borg's Betrayal
Seven of Nine
Tertiary Adjunct of Unimatrix Zero One
Borg-Species 8472 War
2373-2374

Product Team Enterprise Explore Marketplace Pricing Search / Sign in Sign up

/ kubernetes-first-commit Public

Notifications Fork 0 Star 1

Code Pull requests Actions Security Insights

first-commit 1 branch 0 tags Go to file Code

jbeda First commit 2c4b3a5 on Jun 7, 2014 1 commit

api	First commit	8 years ago
cmd	First commit	8 years ago
examples/guestbook	First commit	8 years ago
hooks	First commit	8 years ago
pkg	First commit	8 years ago
src	First commit	8 years ago
third_party	First commit	8 years ago
.gitignore	First commit	8 years ago
LICENSE	First commit	8 years ago
README.md	First commit	8 years ago

README.md

Kubernetes

Kubernetes is an open source reference implementation of container cluster management.

Getting started on Google Compute Engine

Prerequisites

1. You need a Google Cloud Platform account with billing enabled. Visit http://cloud.google.com/console for more details
2. You must have Go installed: www.golang.org

About

The first commit of kubernetes/kubernetes. Borrowed from github.com/kubernetes/kubernetes.

Readme
Apache-2.0 license
1 star
1 watching
0 forks

Releases

No releases published

Packages

No packages published

Languages

Go 57.5% HTML 25.0%
Shell 12.0% SaltStack 2.6%
RAML 1.6% Python 1.3%

6.7 millions loc

```
bash-5.1$ git grep ^ | wc -l
 6717725
bash-5.1$ pwd;
/Users/user/Documents/k8s/kubernetes
bash-5.1$
```

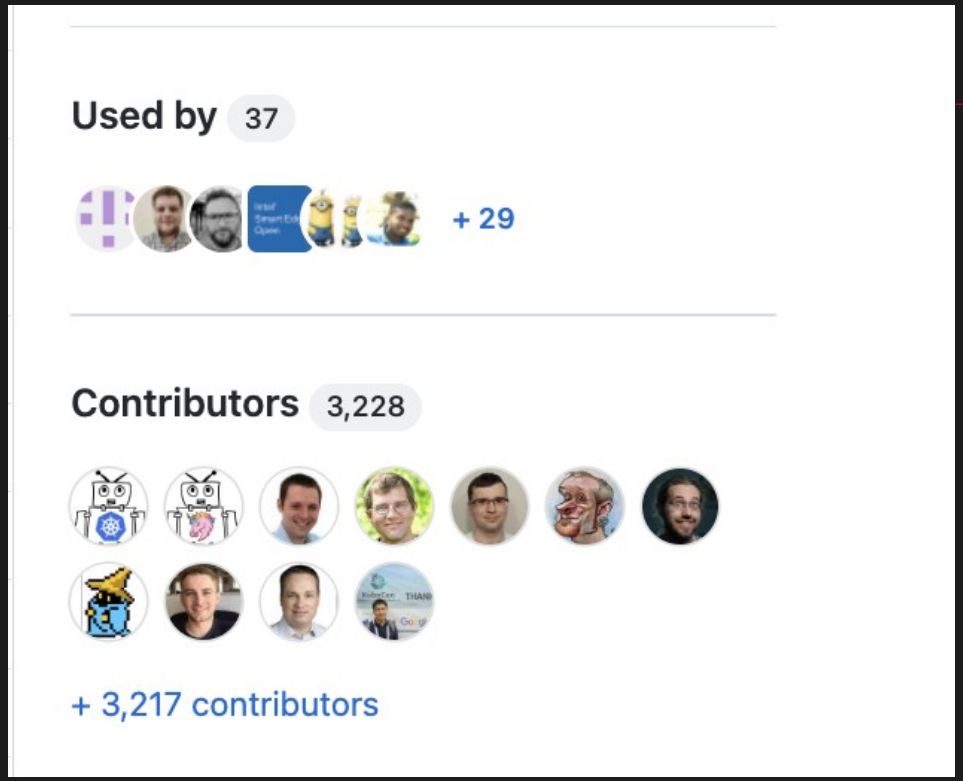
Releases 550

Kubernetes v1.25.0 Latest

3 days ago

+ 549 releases

k8s first commit on June 7 2014



k8s first commit on June 7 2014

Languages

- Go 96.8%
- Shell 2.8%
- PowerShell 0.2%
- Makefile 0.1%
- Dockerfile 0.1%
- Python 0.0%

About

Production-Grade Container Scheduling and Management

kubernetes.io

go kubernetes containers cncf

Readme

Apache-2.0 license

Code of conduct

91.5k stars

3.3k watching

33.5k forks

k8s source build

```
kubernetes
user@ebpf-dev:~/go/src/k8s.io$ cd kubernetes/
user@ebpf-dev:~/go/src/k8s.io/kubernetes$ ls
api           cluster             docs    LICENSE   Makefile.generated_files  plugin             SUPPORT.m
build         cmd                 go.mod  LICENSES  OWNERS                    README.md          test
CHANGELOG     code-of-conduct.md  go.sum  logo      OWNERS_ALIASES            SECURITY_CONTACTS  third_par
CHANGELOG.md  CONTRIBUTING.md     hack    Makefile  pkg                       staging            vendor
user@ebpf-dev:~/go/src/k8s.io/kubernetes$ time make quick-release
+++ [0804 21:21:34] Verifying Prerequisites....
+++ [0804 21:21:34] Building Docker image kube-build:build-b13697006d-5-v1.25.0-go1.19-bullseye.0
+++ [0804 21:25:30] Creating data container kube-build-data-b13697006d-5-v1.25.0-go1.19-bullseye.0
+++ [0804 21:25:35] Syncing sources to container
+++ [0804 21:25:43] Running build command...
+++ [0804 21:25:45] Building go targets for linux/amd64
    k8s.io/kubernetes/hack/make-rules/helpers/go2make (non-static)
+++ [0804 21:26:01] Building go targets for linux/amd64
    k8s.io/code-generator/cmd/prerelease-lifecycle-gen (non-static)
```

```
+++ [0804 23:24:18] Building tarball: node linux-amd64
+++ [0804 23:24:19] Building images: linux-amd64
+++ [0804 23:24:19] Starting docker build for image: kube-apiserver-amd64
+++ [0804 23:24:19] Starting docker build for image: kube-controller-manager-amd64
+++ [0804 23:24:19] Starting docker build for image: kube-scheduler-amd64
+++ [0804 23:24:19] Starting docker build for image: kube-proxy-amd64
+++ [0804 23:24:25] Deleting docker image k8s.gcr.io/kube-scheduler-amd64:v1.24.4-rc.0.12_18faf58c9721c5
+++ [0804 23:24:28] Deleting docker image k8s.gcr.io/kube-controller-manager-amd64:v1.24.4-rc.0.12_18faf58
+++ [0804 23:24:31] Deleting docker image k8s.gcr.io/kube-proxy-amd64:v1.24.4-rc.0.12_18faf58c9721c5
+++ [0804 23:24:34] Deleting docker image k8s.gcr.io/kube-apiserver-amd64:v1.24.4-rc.0.12_18faf58c9721c5
+++ [0804 23:24:34] Docker builds done
+++ [0804 23:24:34] Building tarball: server linux-amd64
+++ [0804 23:25:35] Building tarball: final
+++ [0804 23:25:35] Starting tarball: test linux-amd64
+++ [0804 23:25:35] Waiting on test tarballs
+++ [0804 23:26:27] Building tarball: test portable

real    19m20.823s
user    2m32.221s
sys     0m18.780s
```

10 years borg

Large-scale cluster management at Google with Borg

Abhishek Verma† Luis Pedrosa‡ Madhukar Korupolu
David Oppenheimer Eric Tune John Wilkes

Google Inc.

Abstract

Google's Borg system is a cluster manager that runs hundreds of thousands of jobs, from many thousands of different applications, across a number of clusters each with up to tens of thousands of machines.

It achieves high utilization by combining admission control, efficient task-packing, over-commitment, and machine sharing with process-level performance isolation. It supports high-availability applications with runtime features that minimize fault-recovery time, and scheduling policies that reduce the probability of correlated failures. Borg simplifies life for its users by offering a declarative job specification language, name service integration, real-time job monitoring, and tools to analyze and simulate system behavior.

We present a summary of the Borg system architecture and features, important design decisions, a quantitative analysis of some of its policy decisions, and a qualitative examination of lessons learned from a decade of operational experience with it.

1. Introduction

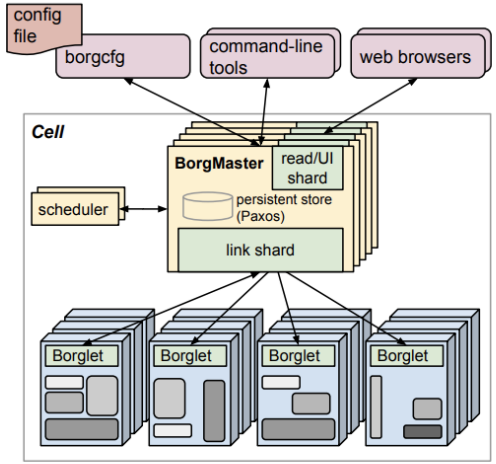


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

cluding with a set of qualitative observations we have made from operating Borg in production for more than a decade.

10 years borg

User view

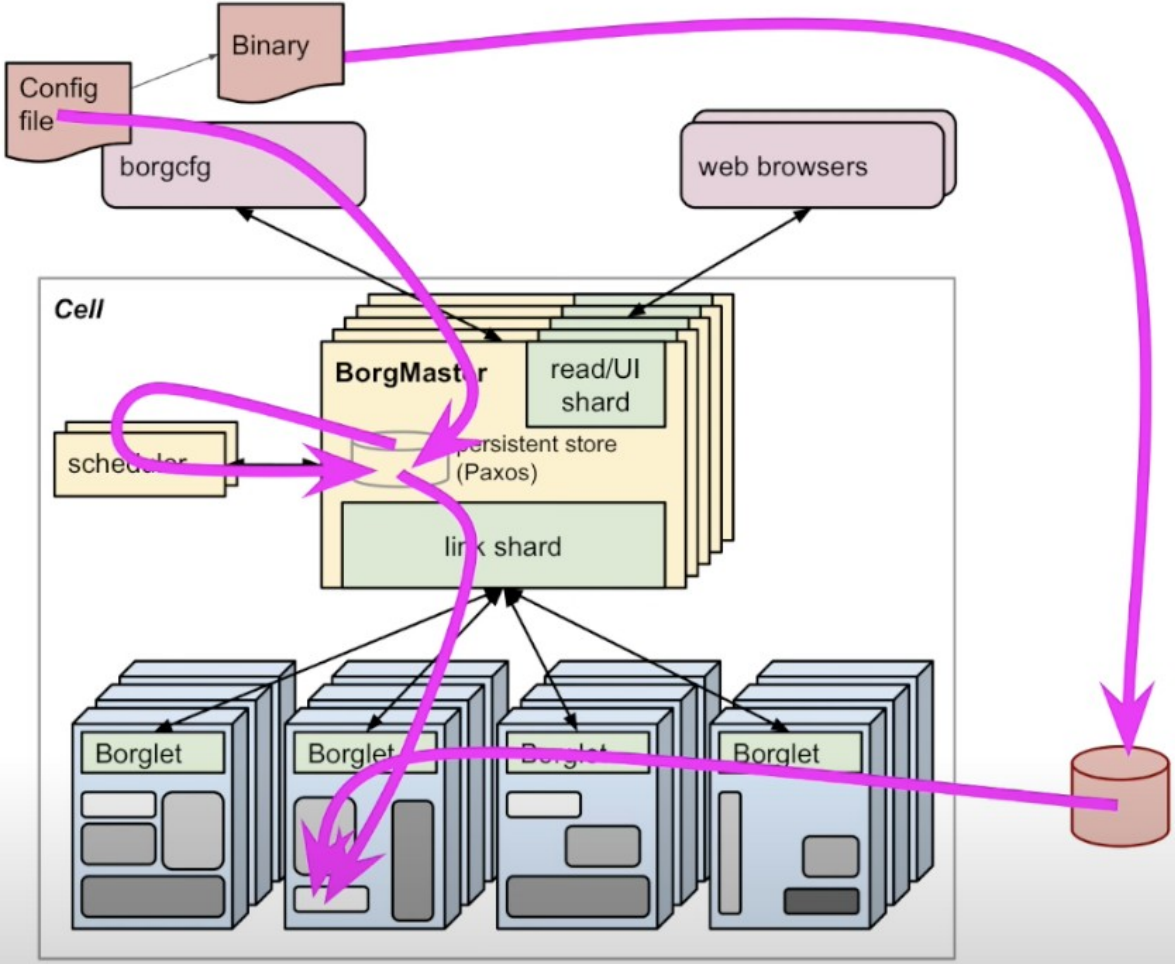
```
job hello_world = {
   runtime = { cell = 'ic' }                // Cell (cluster) to run in
   binary = '.../hello_world_webserver'     // Program to run
   args = { port = '%port%' }               // Command line parameters
   requirements = {      // Resource requirements (optional)
     ram = 100M
     disk = 100M
     cpu = 0.1
   }
   replicas = 10000     // Number of tasks
}
```

Google Cloud Platform

10 years borg

User view

What just happened?



kubernetes

Home Getting Started Community News Events Blog

Kubernetes

κυβερνήτης: *pilot or helmsman of a ship*

kubernetes by Google

http://kubernetes.io

- Top 0.01% of all Github projects
- 800+ unique contributors
- 15000+ people signed up for k8s meetups

10 years b

kubernetes

Home Getting Started Community News Events Blog

Kubernetes

Direct Borg analogues:

- containers
- pods
- Kubelet
- persistent, declarative specs
- reconciliation loops

One computer

“We must treat the datacenter itself as one massive warehouse-scale computer.”

Luiz Andre Barroso, Urs Hoelzle
The Datacenter as a Computer

One computer

“We must treat the ~~datacenter~~ world itself as one massive global-scale computer.”

Kelsey Hightower

Just now at the doGot conference in Paris.

One computer

CLOUD NATIVE COMPUTING FOUNDATION

About Projects Training Community Blog & News Join

CNCF projects are the foundation of cloud native computing

As part of the **Linux Foundation**, we provide support, oversight and direction for fast-growing, **cloud native** projects, including Kubernetes, Envoy, and Prometheus.

18 GRADUATED PROJECTS

37 INCUBATING PROJECTS

77 SANDBOX PROJECTS

ALL PROJECTS

-toto argo Backstage Buildpacks.io Chaos Mesh cilium cloudevents CNI container-d

Open Policy Agent notary NATS LONGHORN Litmus LINKERD Kyverno KubeVirt kubernetes

pmi

pmi.org/learning/library/art-war-project-management-principles-8165

The Art of War and Project Management.

Although it is not clear (Wikipedia, 2006) if Sun Tzu – Exhibit 1 - was one person or a group of philosophers, his pronouncements are extremely clear and definitely practical, no matter which Age is considering them.

Exhibit 1 – Sun Tzu?

Reading the Art of War I discovered strong collocations to Project Management.

The basis for all thoughts presented by Sun Tzu are five elements

described by Robert Friedler (2006) as: path, heaven, earth, leader and law (Exhibit 2). And should be the key things for any Project Manager to remember when managing a project.

	Direct translation	Meaning in the Art of War	Examples
	path	mutual philosophy & purpose of the group	patriotism, team sprit, shared values
	heaven	environmental factors	time, seasons, light, darkness
	earth	situation	distance from opponent, even or uneven ground
	leader	leadership	intelligence, credibility, compassion, courage, discipline
	law	art, as in "Art of War"	the ability to perceive and implement the strategic concept

Exhibit 2 - Sun Tzu's Five Elements

5 elements

	Direct translation	Meaning in the Art of War	Examples
	path	mutual philosophy & purpose of the group	patriotism, team sprit, shared values
	heaven	environmental factors	time, seasons, light, darkness
	earth	situation	distance from opponent, even or uneven ground
	leader	leadership	intelligence, credibility, compassion, courage, discipline
	law	art, as in "Art of War"	the ability to perceive and implement the strategic concept

Crazy delicious

You can enjoy doing your job

Contents

k8s security

4c

k8s is not a security system

Sec wasn’t built in natively

k8s sec improving (1.22+)

K8s sec

OWASP

PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org

OWASP Kubernetes Top Ten

Main

About the Kubernetes Top 10

When adopting Kubernetes, we introduce new risks to our applications and infrastructure. The The *OWASP Kubernetes Top 10* is aimed at helping security practitioners, system administrators, and software developers prioritize risks around the Kubernetes ecosystem. The Top Ten is a prioritized list of these risks backed by data collected from organizations varying in maturity and complexity.

- K00:2022 Welcome to the Kubernetes Security Top Ten
- K01:2022 Insecure Workload Configurations
- K02:2022 Supply Chain Vulnerabilities
- K03:2022 Overly Permissive RBAC Configurations
- K04:2022 Lack of Centralized Policy Enforcement
- K05:2022 Inadequate Logging and Monitoring
- K06:2022 Broken Authentication Mechanisms
- K07:2022 Missing Network Segmentation Controls
- K08:2022 Secrets Management Failures
- K09:2022 Misconfigured Cluster Components
- K10:2022 Outdated and Vulnerable Kubernetes Components
- Other Risks to Consider

K8s se

cloud native wiki
by aqua

- DevSecOps
- Docker Container
- Cloud Native Applications
- Kubernetes in Production
- Supply Chain Security
- Containerized Architecture
- Cloud Security
- Serverless Architecture
- Container Platforms
- Kubernetes
- Vulnerability Management

How Can You Best Secure Your Kubernetes (K8s) Deployment?

Kubernetes is a complex platform and requires extensive configuration and management. To keep Kubernetes workloads safe, especially in a production environment, you need to address key architectural vulnerabilities and platform dependencies, by implementing security best practices.

In this article, you will learn about the following Kubernetes security best practices:

- Enable Role-Based Access Control (RBAC)
- Use Third-Party Authentication for API Server
- Protect ETCD with TLS and Firewall
- Isolate Kubernetes Nodes
- Monitor Network Traffic to Limit Communications
- Use Process Whitelisting
- Turn on Audit Logging
- Keep Kubernetes Version Up to Date
- Lock Down Kubelet
- Secure Kubernetes with Aqua

K8s sec

This guide describes the security challenges associated with setting up and securing a Kubernetes cluster. It includes strategies for system administrators and developers of National Security Systems, helping them avoid common misconfigurations and implement recommended hardening measures and mitigations when deploying Kubernetes. This guide details the following mitigations:

- Scan containers and Pods for vulnerabilities or misconfigurations.
- Run containers and Pods with the least privileges possible.
- Use network separation to control the amount of damage a compromise can cause.
- Use firewalls to limit unneeded network connectivity and use encryption to protect confidentiality.
- Use strong authentication and authorization to limit user and administrator access as well as to limit the attack surface.
- Capture and monitor audit logs so that administrators can be alerted to potential malicious activity.
- Periodically review all Kubernetes settings and use vulnerability scans to ensure risks are appropriately accounted for and security patches are applied.

K

Ship of Fools: Shoring Up Kubernetes Security - SANS Secure DevOps Summit 2018

DIY Pen-Testing for Your

K8s se

juju.is/cloud-native-kubernetes-usage-report-2022

Juju Charmhub About Blog Learn Contribute

Introduction
Key takeaways
About the survey

Who is using Kubernetes and cloud native technologies?
cloud native: goals, benefits, and estate size
Container and Kubernetes usage
Configuration management
CNCF landscape
Kubernetes operators
Kubernetes advanced usage
Edge computing
Industry expert bios
Acknowledgements

Key takeaways

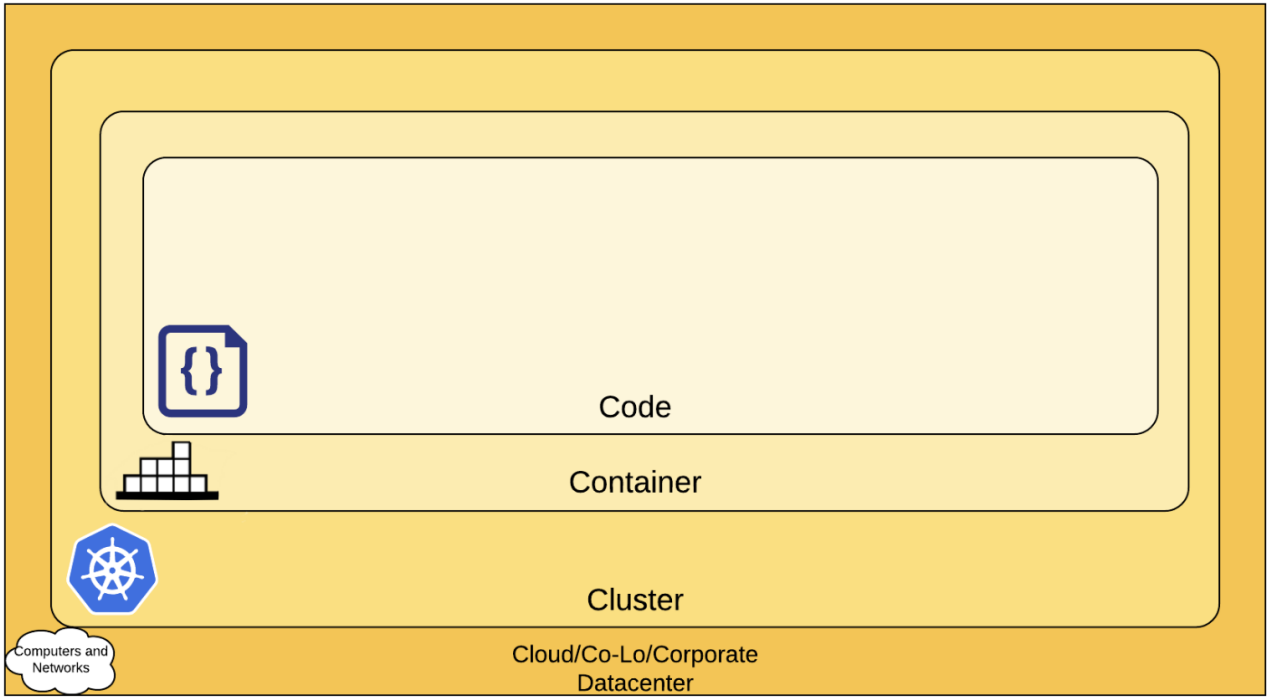
- **Hybrid vs multi-cloud: the reality behind the adoption**
  More than 83% of respondents told us they're using either hybrid or multi-cloud. In the last year alone, the percentage of respondents who did not use hybrid or multi-cloud dropped from 22.4% to 16.4%. This time, Tim Hockin discusses the reality behind that adoption: "People often build a straw man of hybrid or multi-cloud, with the idea of one giant mesh that spans the world and all the clouds, applications running wherever capacity is cheap and available. But in reality, that's not at all what people are doing with it. What they're actually doing is using each environment for just the things they have to use it for."
- **Kubernetes on bare metal**
  The question of where to run applications is an interesting one. 14% of respondents said that they run everything on Kubernetes, over 20% said on bare metal and VMs, and over 29% said a combination of bare metal, VMs, and Kubernetes. As highlighted by Ihor Dvoretskyi, this distribution shows how the flexibility of Kubernetes allows organisations to run the same type of workloads everywhere. Looking back at last year's highlight, where Kelsey Hightower stated that bare metal was a better choice for compute and resource-heavy use cases such as interactive machine learning jobs, it seems that the tune is changing. Actually, as running Kubernetes is becoming more accessible, Alexis Richardson speculates that organisations would further adopt Kubernetes on bare metal if they knew it was possible.
- **Security is still everyone's concern**
  This comes as no surprise: 38% of respondents suggest that security is the most important consideration whether when operating Kubernetes, building container images or defining an edge strategy. Keeping clusters up-to-date is a definitive best practice to solve security issues. However, according to Jose Miguel Parrella, it is not as embedded within the strategic-thinking IT infrastructure group as one could expect. Today, it is more of a Day-30 discussion that only occurs within the small team of Kubernetes maintainers of every organisation. Combined with the fact that only 13.5% of people reported that they've "mastered" security in the cloud native space, it is clear that organisations have some room to grow when it comes to properly adopting and managing Kubernetes in production.

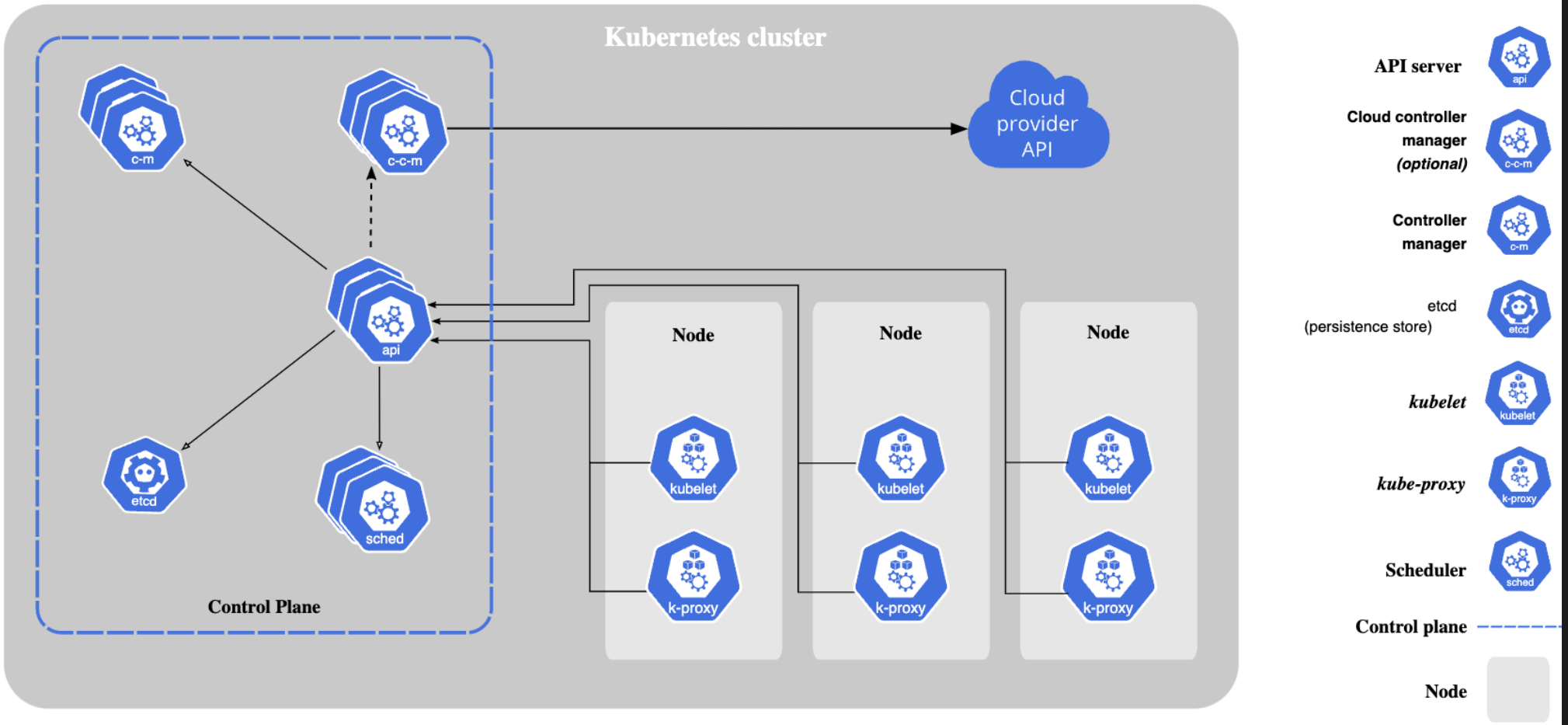
4c

The 4C's of Cloud Native security

You can think about security in layers. The 4C's of Cloud Native security are Cloud, Clusters, Containers, and Code.

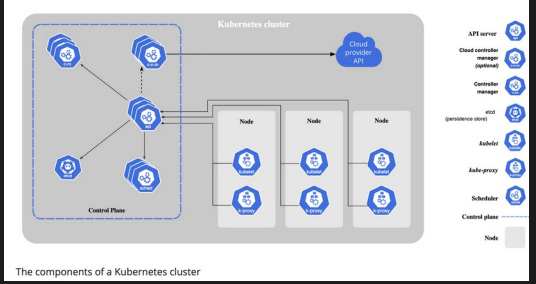
> **Note:** This layered approach augments the defense in depth computing approach to security, which is widely regarded as a best practice for securing software systems.





The components of a Kubernetes cluster

k8s control plane sec



The components of a Kubernetes cluster

Install / upgrade / downgrade / cordon / taints affinity / labels
CNI install
Apiserver
Kubelet
Schedulercontroler

Enable RBAC
Enable admission control (image pull policy, pod security standard)
Enable TLS
Remove anonymous

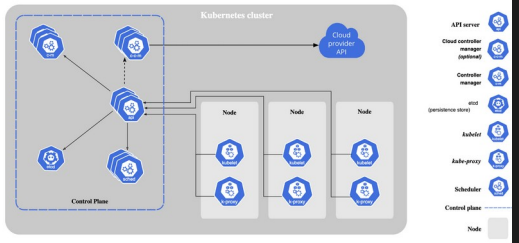
Network Policy
Secrets / ConfigMaps

etcd tls
etcd encryption
etcd backup / restore

k8s node sec

Kubelet auth
Kubelet TLS
Kubelet config remove anonymous

Labels
Taints
Affinity
Resource Quotas

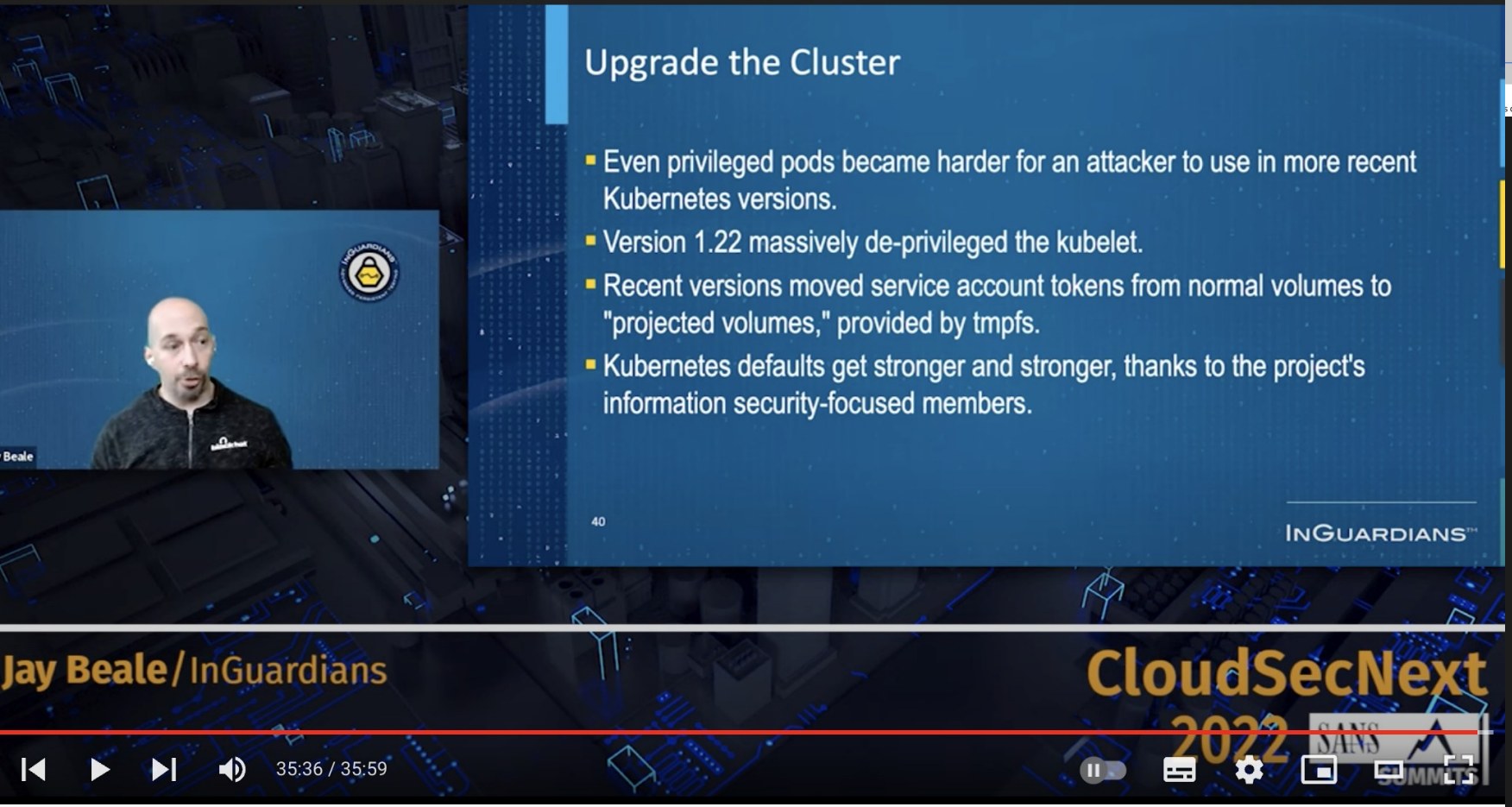


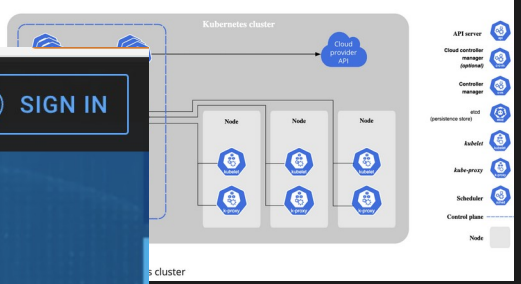
The components of a Kubernetes cluster

YouTube GB

sans cloud security

SIGN IN





35:36 / 35:59

Kubernetes Attack and Defense: Break Out and Escalate!

k8s cluster sec

kubernetes.io/blog/2021/08/04/kubernetes-1-22-release-announcement/

kubernetes

Documentation Kubernetes Blog Training Partners Community Case Studies Versions English

Search

2022

2021

Kubernetes-in-Kubernetes and the WEDOS PXE bootable server farm

More secure control plane with kubeadm

A new alpha feature allows running the `kubeadm` control plane components as non-root users. This is a long requested security measure in `kubeadm`. To try it you must enable the `kubeadm` specific RootlessControlPlane feature gate. When you deploy a cluster using this alpha feature, your control plane runs with lower privileges.

For `kubeadm`, Kubernetes 1.22 also brings a new v1beta3 configuration API. This iteration adds some long requested features and deprecates some existing ones. The v1beta3 version is now the preferred API version; the v1beta2 API also remains available and is not yet deprecated.

k8s cluster sec

kubernetes.io/blog/2021/12/07/kubernetes-1-23-release-announcement/

kubernetes

Documentation Kubernetes Blog Training Partners Community Case Studies Versions English

Search

2022

2021

Kubernetes-in-Kubernetes and

Software Supply Chain SLSA Level 1 Compliance in the Kubernetes Release Process

Kubernetes releases now generate provenance attestation files describing the staging and release phases of the release process. Artifacts are now verified as they are handed over from one phase to the next. This final piece completes the work needed to comply with Level 1 of the SLSA security framework (Supply-chain Levels for Software Artifacts).

k8s cluster sec

kubernetes.io/blog/2022/05/03/kubernetes-1-24-release-announcement/

kubernetes

Documentation Kubernetes Blog Training Partners Community Case Studies Versions English

Search

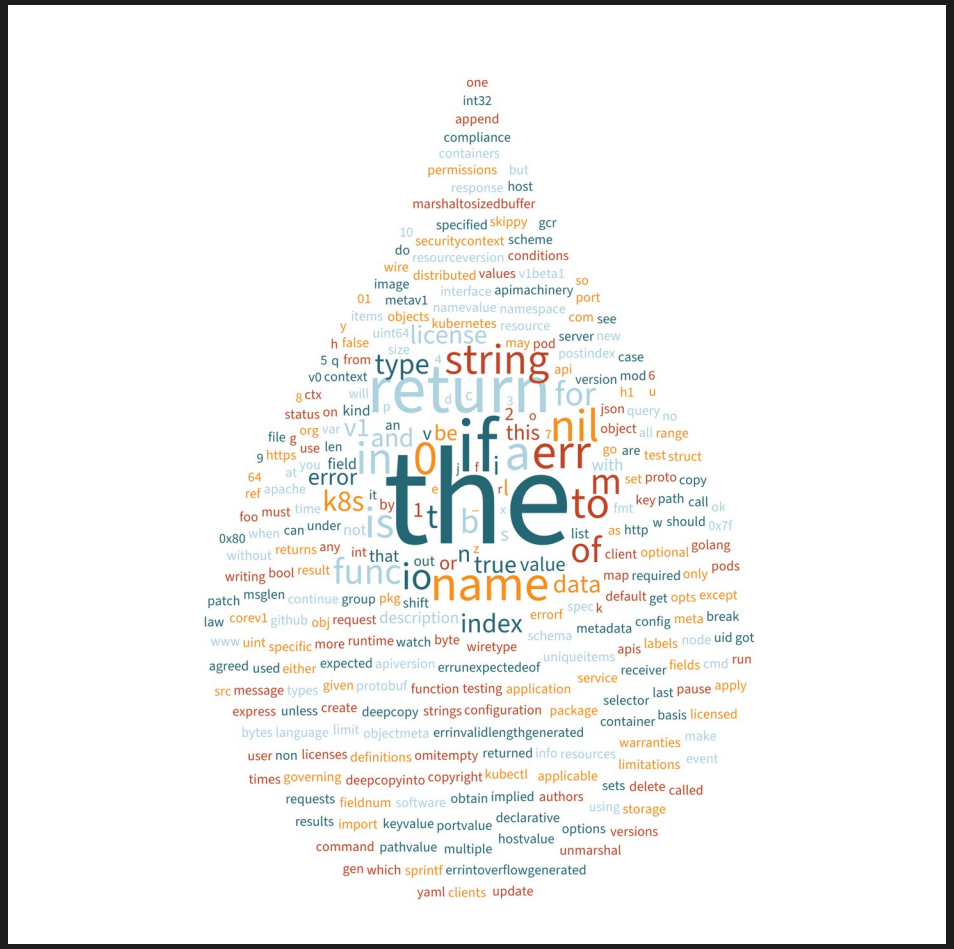
2022

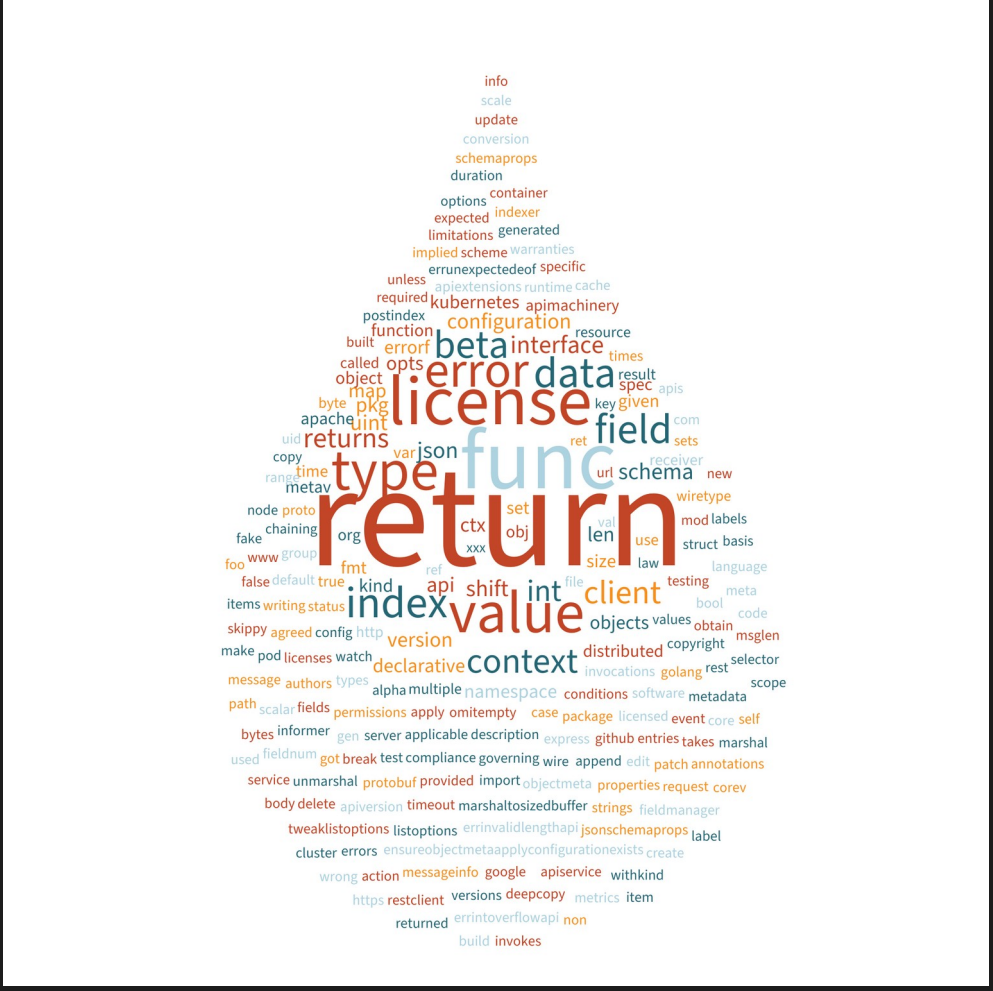
Kubernetes 1.25: CSI Inline Volumes have graduated to

Signing Release Artifacts

Release artifacts are signed using cosign signatures, and there is experimental support for verifying image signatures. Signing and verification of release artifacts is part of increasing software supply chain security for the Kubernetes release process.

k8s src


one
int32
append
compliance
containers
permissions
but
response
host
marshaltosizedbuffer
specified
skippy
gcr
securitycontext
scheme
resourceversion
conditions
distributed
values
v1beta1
image
interface
apimachinery
metav1
namevalue
namespace
items
objects
kubernetes
resource
com
see
license
uint64
server
new
may
pod
postindex
case
size
type
string
return
version
mod
context
api
h1
status
kind
nil
this
object
range
file
use
len
field
error
k8s
is
the
if
and
in
0
a
err
to
m
set
proto
copy
key
path
call
ok
foo
must
time
not
b
t
1
fmt
w
should
0x7f
0x80
when
can
under
list
as
http
without
returns
any
int
that
out
or
n
true
value
of
client
optional
golang
pods
writing
bool
result
func
io
name
data
map
required
only
patch
msglen
continue
group
pkg
shift
default
get
opts
except
law
corev1
github
obj
request
description
index
errorf
spec
k
metadata
config
meta
break
www
uint
specific
more
runtime
watch
byte
wiretype
schema
labels
node
uid
got
agreed
used
either
expected
apiversion
errunexpectedeof
uniqueitems
apis
receiver
fields
cmd
run
src
message
types
given
protobuf
function
testing
application
service
last
pause
apply
express
unless
create
deepcopy
strings
configuration
package
selector
basis
licensed
bytes
language
limit
objectmeta
errinvalidlengthgenerated
container
make
user
non
licenses
definitions
omitempty
warranties
limitations
event
times
governing
deepcopyinto
copyright
kubectl
applicable
sets
delete
called
requests
fieldnum
software
obtain
implied
authors
using
storage
results
import
keyvalue
portvalue
declarative
options
versions
command
pathvalue
multiple
hostvalue
unmarshal
gen
which
sprintf
errintoverflowgenerated
yaml
clients
update


info
scale
update
conversion
schemaprops
duration
options
container
expected
indexer
limitations
generated
implied
scheme
warranties
errunexpectedeof
specific
unless
apiextensions
runtime
cache
required
kubernetes
apimachinery
postindex
configuration
function
resource
built
errorf
beta
interface
times
called
opts
error
data
result
object
spec
apis
map
byte
pkg
license
given
key
apache
uint
field
com
uid
returns
ret
sets
copy
var
json
func
receiver
time
type
schema
new
range
metav
wiretype
node
proto
set
return
mod
labels
fake
chaining
org
ctx
obj
val
len
use
struct
basis
foo
www
group
xxx
size
law
language
fmt
ref
false
default
true
kind
api
shift
int
client
testing
meta
file
items
writing
status
index
value
objects
values
obtain
bool
code
skippy
agreed
config
http
version
copyright
msglen
make
pod
licenses
watch
declarative
context
distributed
invocations
golang
rest
selector
message
authors
types
alpha
multiple
namespace
conditions
software
metadata
scope
path
scalar
fields
permissions
apply
omitempty
case
package
licensed
event
core
self
bytes
informer
gen
server
applicable
description
express
github
entries
takes
marshal
used
fieldnum
got
break
test
compliance
governing
wire
append
edit
patch
annotations
service
unmarshal
protobuf
provided
import
objectmeta
properties
request
corev
body
delete
apiversion
timeout
marshaltosizedbuffer
strings
fieldmanager
tweaklistoptions
listoptions
errinvalidlengthapi
jsonschemaprops
label
cluster
errors
ensureobjectmetaapplyconfigurationexists
create
wrong
action
messageinfo
google
apiservice
withkind
https
restclient
versions
deepcopy
metrics
item
returned
errintoverflowapi
non
build
invokes

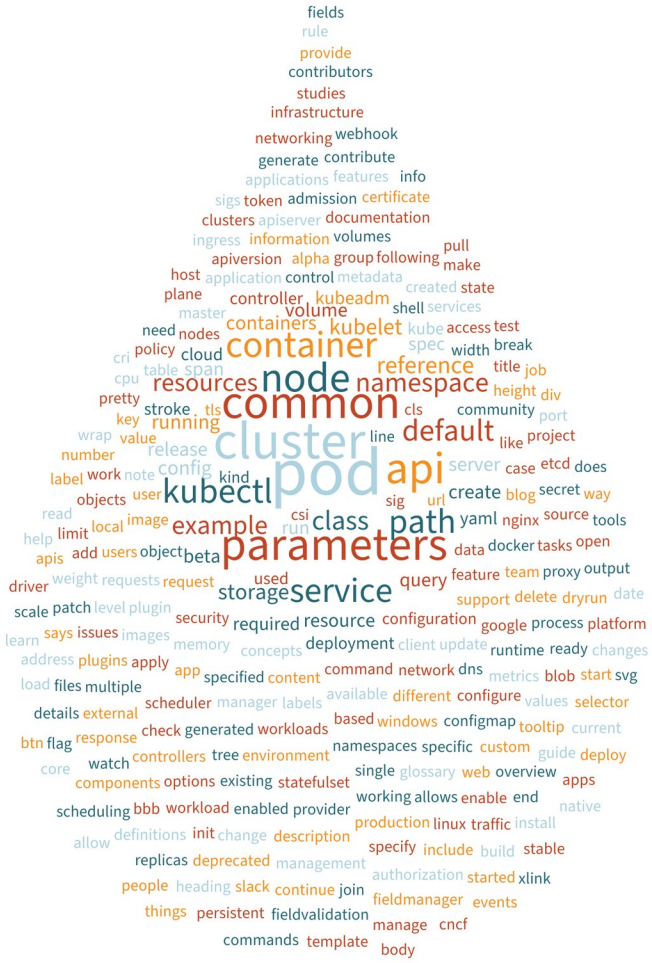
k8s src

value
context
pod
key
ctx
client
request
resource
errorf
result
metav
kubernetes
description
interface
framework
test
header
response
spec
node
case
api
types
path
namespace
container
data
testing
fields
errors
volume
requests
urlparams
runtime
package
append
service
omitempty
conditions
gensupport
kind
proto
config
returns
req
defer
obj
autorest
klog
operation
create
message
object
expected
googleapi
distributed
objectmeta
apimachinery
copyright
permissions
deployment
forcesendfields
governing
statusnotmodified
timestamp
marshaljson
elementdescription
internal
protoimpl
descriptor
instances
remove
original
parameter
dorequest
readonly
scopes
listoptions
queryparameters
services
unique
wire
counterid
represents
restclient
poduid
trace
objs
gets
persistentvolume
deprecated
sprintf
check
function
offset
image
sync
claim
matches
rule
pattern
pagetoken
driver
parse
kube
podname
expand
discovery
operator
note
statusok
groupversion
rights
invocations
batch
waiting
certificate
table
resourcelist
regardless
unrecognized
ifnonematch
requestid
declarative
annotations
grpc
tracing
validation
multiple
unmarshal
testcases
selector
reference
optional
limitations
reserved
creating
controller
handler
include
software
protoreflect
based
global
deepequal
send
exec
infos
volumes
template
apiversion
item
nodes
xfe
xac
xbc
compiler
apply
deleted
podspec
wiretype
prefix
xab
xad
source
job
close
blas
line
ptr
opt
const
region
licensed
scheme
phase
xfa
apps
device
specify
xcb
xdf
openapiv
format
insert
syscall
stats
metric
scope
wait
gce
update
mode
etcd
xdd
language
instance
strings
block
ipv
pvc
bar
uid
ctrl
null
yaml
cloud
start
label
event
tag
secret
handle
omitted
projectid
float
specific
link
contains
merge
xea
actual
local
xeb
filepath
location
arg
min
deepequal
reset
parent
core
max
xbd
sent
failure
exists
exist
base
route
plugin
mock
incx
disk
values
compute
bytes
corev
failed
options
obtain
switch
zero
nullfields
retry
body
agreed
unix
rerr
fake
read
chan
express
xcf
auth
root
logf
usage
parser
number
required
urls
dst
tls
buf
xbf
cmd
xda
xec
xdb
xba
level
tcp
action
equal
watch
aws
input
ports
xae
write
azure
run
user
xcc
host
patch
raw
sets
lock
xfb
break
build
content
implied
xca
xce
xff
src
lda
net
basis
output
match
fatalf
meta
policy
entry
expr
keys
shift
xee
sort
slice
unless
golang
address
init
xbb
apis
results
ginkgo
assert
token
fail
cfg
xaa
labels
law
args
port
xed
reader
math
single
msg
testcase
alpha
called
licenses
flag
xaa
ready
continue
second
protocol
existing
zones
order
allows
caller
manager
events
containers
mount
condition
diff
xdc
partial
unlock
provided
expect
uuid
providers
rules
target
state
cpu
future
post
logger
accept
applicable
current
utf
summary
authors
uintptr
capacity
length
metadata
cancel
updates
pending
added
period
plugins
dev
directory
store
endpoint
encode
unexpected
projects
running
conn
proxy
process
addr
feature
nomethod
subscriptionid
protobuf
compliance
statuscode
expectnoerror
warranties
resourcegroupname
objects
headers
reqheaders
network
method
kubelet
resources
xbe
virtual
instead
limit
expression
created
duration
protowire
load
xfd
chain
queue
flags
iter
stop
missing
gomock
nodename
sum
cert
apiserver
metrics
schema
tmp
reason
end
clientset
xde
replicas
buffer
specified
groups
fatal
page
pointer
success
count
invalid
xef
unsafe
storage
xee
unless
golang
address
init
xbb
apis
results
ginkgo
expr
entry
policy
meta
keys
shift
match
fatalf
xca
output
xfb
break
build
content
implied
xce
xda
xec
query
writing
cache
xdb
desc
xba
level
tcp
action
equal
reflect
filter
aws
watch
input
ports
xae
write
azure
import
copy
run
user
xcc
host
patch
raw
sets
lock
typ
rest
cluster
scale
join
names
xeb
local
actual
returned
proto
arg
filepath
location
tag
event
handle
secret
operation
klog
matrix
omitted
projectid
float
xea
merge
link
contains
specific
podspec
deleted
apply
compiler
min
wiretype
prefix
xab
xad
source
job
close
blas
line
deepequal
global
based
protoreflect
software
send
exec
infos
volumes
template
apiversion
item
nodes
xfe
xac
xbc
scopes
readonly
dorequest
parameter
deployment
gets
persistentvolume
deprecated
objs
listoptions
queryparameters
services
trace
unique
wire
counterid
xfe
represents
restclient
poduid
descriptor
internal
protoimpl
elementdescription
requestid
ignore
unknown
properties
ifnonematch
unrecognized
resourcelist
regardless
rights
invocations
batch
waiting
certificate
table
multiple
unmarshal
testcases
selector
reference
optional
limitations
reserved
creating
controller
handler
include
software
protoreflect
based
global
deepequal

```
most common local variable names
 38882 err
 12390 c
  9106 t
  8300 ctx
  8113 m
  7329 i
  7115 s
  6331 ok
  5416 b
  5306 v
  4233 client
  3998 f
  3846 e1
  3768 p
  3686 result
  3652 req
  3588 r
  3507 n
  3443 name
  3434 x
  2612 opts
  2459 a
  2327 resp
  2319 key
  2302 k
  2272 value
  2094 pod
  2062 e
  2043 obj
  1990 path
```

```
most common global variable names
    36 errorList
    36 signalList
    10 alwaysReady
     7 parseCPUInfo
     7 logger
     6 codecs
     6 scheme
     6 now
     5 ccc
     5 zero
     5 marshalerType
     5 nfkcIndex
     5 nfkcValues
     5 recompMap
     5 nfkcSparseOffset
     5 debug
     5 recompMapOnce
     5 nfcSparseOffset
     5 port
     5 nfcIndex
     5 nfcValues
     5 modkernel32
     5 All
     5 nfkcSparseValues
     5 decomps
     5 nfcSparseValues
     5 sysctlMib
     5 ErrNotSupported
     5 controllerKind
     5 errUnsupported
```

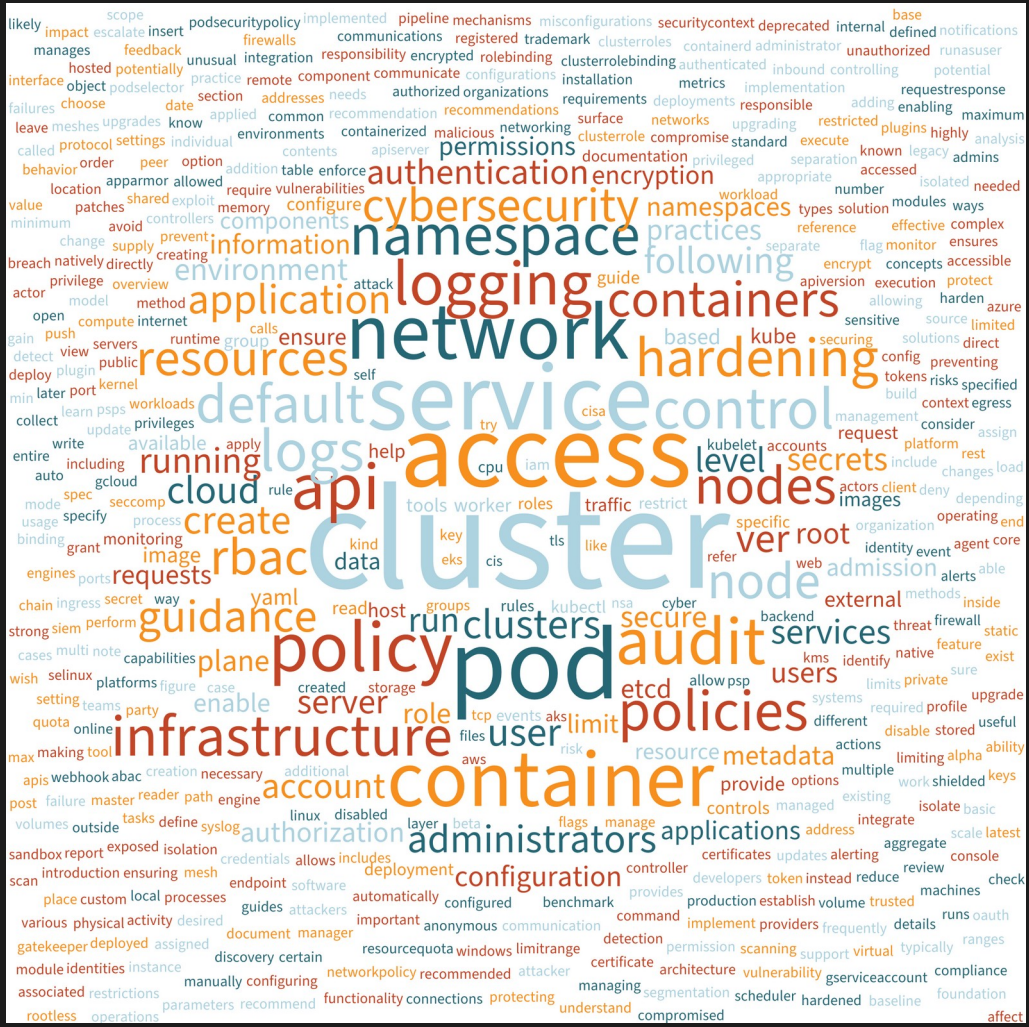
K8s documentation



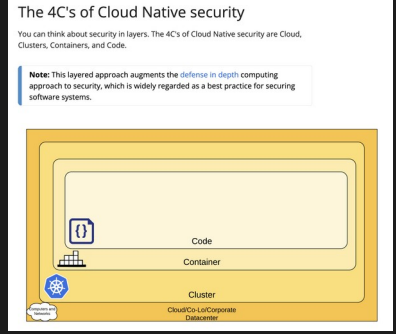
!k8s

https://kubernetes.io/blog/2016/08/security-best-practices-kubernetes-deployment/
https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/
https://sysdig.com/learn-cloud-native/kubernetes-security/kubernetes-security-101/
https://owasp.org/www-project-kubernetes-top-ten/
https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENIN
https://www.aquasec.com/cloud-native-academy/kubernetes-in-production/kubernetes-security
https://docs.cilium.io/en/latest/security/policy/
https://docs.aws.amazon.com/eks/latest/userguide/security-best-practices.html
https://cloud.google.com/kubernetes-engine/docs/how-to/hardening-your-cluster
https://docs.microsoft.com/en-us/azure/aks/best-practices

k8s ten bp


cluster
access
service
pod
container
network
namespace
infrastructure
policy
policies
audit
logging
cybersecurity
authentication
permissions
encryption
containers
hardening
control
nodes
secrets
resources
default
logs
api
rbac
guidance
administrators
configuration
authorization
account
applications
metadata
services
users
clusters
run
secure
create
cloud
running
application
environment
information
practices
following
namespaces
level
ver
root
node
admission
external
server
enable
plane
etcd
limit
user
role
resource
provide
controls
deployment
requests
data
image
monitoring
secret
yaml
host
read
kubectl
rules
backend
threat
firewall
kms
identify
native
feature
exist
private
limits
required
profile
upgrade
different
disable
stored
useful
actions
limiting
alpha
ability
multiple
work
shielded
keys
isolate
basic
integrate
address
scale
latest
aggregate
console
alerting
updates
certificates
review
check
machines
runs
oauth
details
frequently
providers
implement
production
establish
volume
trusted
reduce
instead
token
developers
controller
provides
benchmark
configured
automatically
important
anonymous
communication
command
detection
permission
scanning
support
virtual
typically
ranges
compliance
gserviceaccount
vulnerability
architecture
certificate
managing
segmentation
scheduler
hardened
baseline
foundation
affect
compromised
understand
protecting
connections
functionality
recommended
networkpolicy
attacker
limitrange
resourcequota
windows
discovery
certain
manually
configuring
instance
identities
module
gatekeeper
deployed
assigned
document
manager
desired
activity
physical
various
place
custom
local
processes
guides
attackers
endpoint
software
introduction
ensuring
mesh
scan
sandbox
report
exposed
isolation
credentials
allows
includes
volumes
outside
tasks
define
syslog
linux
disabled
layer
beta
flags
manage
quota
online
max
making
tool
wish
selinux
platforms
figure
case
created
storage
setting
teams
party
apis
webhook
abac
creation
necessary
additional
post
failure
master
reader
path
engine
tcp
events
aks
files
aws
risk
chain
ingress
way
cyber
nsa
strong
siem
perform
cases
multi
note
capabilities
kubelet
accounts
request
platform
rest
changes
load
deny
depending
operating
end
agent
core
events
identity
refer
web
methods
inside
able
alerts
usage
specify
process
binding
grant
kind
key
tls
like
eks
cis
mode
spec
seccomp
tools
worker
roles
traffic
restrict
actors
client
images
specific
organization
min
later
port
kernel
deploy
plugin
view
public
servers
runtime
group
calls
ensure
self
cisa
build
context
egress
tokens
risks
specified
config
preventing
solutions
direct
sensitive
source
limited
allowing
harden
azure
apiversion
execution
protect
encrypt
concepts
accessible
flag
monitor
ensures
reference
effective
complex
workload
number
isolated
needed
modules
ways
types
solution
separate
appropriate
accessed
known
legacy
admins
privileged
standard
execute
compromise
clusterrole
networks
upgrading
restricted
plugins
highly
analysis
responsible
adding
enabling
maximum
requestresponse
implementation
metrics
deployments
requirements
authorized
organizations
recommendations
surface
malicious
networking
containerized
environments
recommendation
common
applied
contents
apiserver
addition
table
enforce
documentation
behavior
order
peer
option
apparmor
allowed
require
vulnerabilities
location
patches
shared
exploit
memory
configure
value
minimum
avoid
supply
prevent
creating
controllers
components
change
breach
natively
directly
privilege
overview
actor
model
method
attack
guide
open
compute
internet
push
gain
detect
collect
learn
psps
workloads
privileges
update
write
entire
available
apply
help
cpu
iam
auto
including
gcloud
rule
tools
worker
roles
traffic
restrict
specify
process
binding
grant
monitoring
image
engines
ports
requests
yaml
chain
ingress
secret
way
strong
siem
perform
cases
multi
note
capabilities
plane
wish
selinux
platforms
figure
case
setting
teams
party
quota
online
max
making
tool
apis
webhook
abac
creation
necessary
post
failure
master
reader
path
engine
volumes
outside
tasks
define
syslog
linux
disabled
sandbox
report
exposed
isolation
credentials
allows
includes
scan
introduction
ensuring
mesh
endpoint
software
place
custom
local
processes
guides
attackers
various
physical
activity
desired
gatekeeper
deployed
assigned
document
manager
module
identities
instance
discovery
certain
manually
configuring
associated
restrictions
parameters
recommend
rootless
operations
likely
impact
escalate
insert
podsecuritypolicy
implemented
pipeline
mechanisms
misconfigurations
securitycontext
deprecated
internal
base
defined
notifications
manages
feedback
firewalls
communications
registered
trademark
clusterroles
containerd
administrator
unauthorized
runasuser
hosted
potentially
unusual
integration
responsibility
encrypted
rolebinding
clusterrolebinding
authenticated
inbound
controlling
potential
interface
object
podselector
practice
remote
component
communicate
configurations
installation
failures
choose
date
section
addresses
needs
leave
meshes
upgrades
know
called
protocol
settings
individual
behavior
order
peer
option
location
apparmor
allowed
value
patches
shared
exploit
minimum
avoid
supply
prevent
creating
breach
natively
directly
privilege
overview
actor
model
method
open
compute
internet
push
gain
detect
deploy
plugin
min
later
port
kernel
collect
learn
psps
workloads
privileges
update
write
entire
available
apply
including
auto
gcloud
mode
spec
seccomp
usage
specify
process
binding
grant
monitoring
image
engines
ports
requests
chain
ingress
secret
way
strong
siem
perform
cases
multi
note
capabilities
plane
wish
selinux
platforms
figure
case
setting
teams
party
quota
online
max
making
tool
apis
webhook
abac
creation
necessary
post
failure
master
reader
path
engine
volumes
outside
tasks
define
syslog
linux
disabled
sandbox
report
exposed
isolation
credentials
allows
includes
scan
introduction
ensuring
mesh
endpoint
software
place
custom
local
processes
guides
attackers
various
physical
activity
desired
gatekeeper
deployed
assigned
document
manager
module
identities
instance
discovery
certain
manually
configuring
associated
restrictions
parameters
recommend
rootless
operations

!k8s



code: languages (c++, java, python, js, rust, how about go?)
ci/cd, repo management, build / deploy / patch time
readability, compatibility -> attack surface

containers: base image, library, multi-stage, user, fs / v
image repo / scan, supply chain, deep dependency, othogonality

data center: access control, server patch, configuration
redundancy, network

open source tools:
code: go simple, readability, promise backward compbality, static, compiled, grpc, no repo

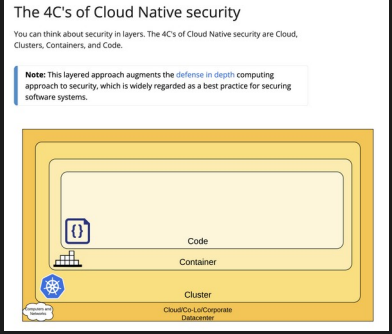
containers: Dockerfile, build tools, image repo, image scan trivy, kubebench, image pull polic

data center: server patch/upgrrade, apparmor/selinux, sysdig/falco

philosophical thinking

1 computer
1 network
1 tool

EBPF
XDP



ebpf

Eye of Sauron

syscalls

```
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$ grep "asmlinkage long sys" syscalls.h  | wc
    441    3020   27803
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$ uname -r
5.15.0-46-generic
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$ cat /etc/os-release | grep VERSION
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$
```

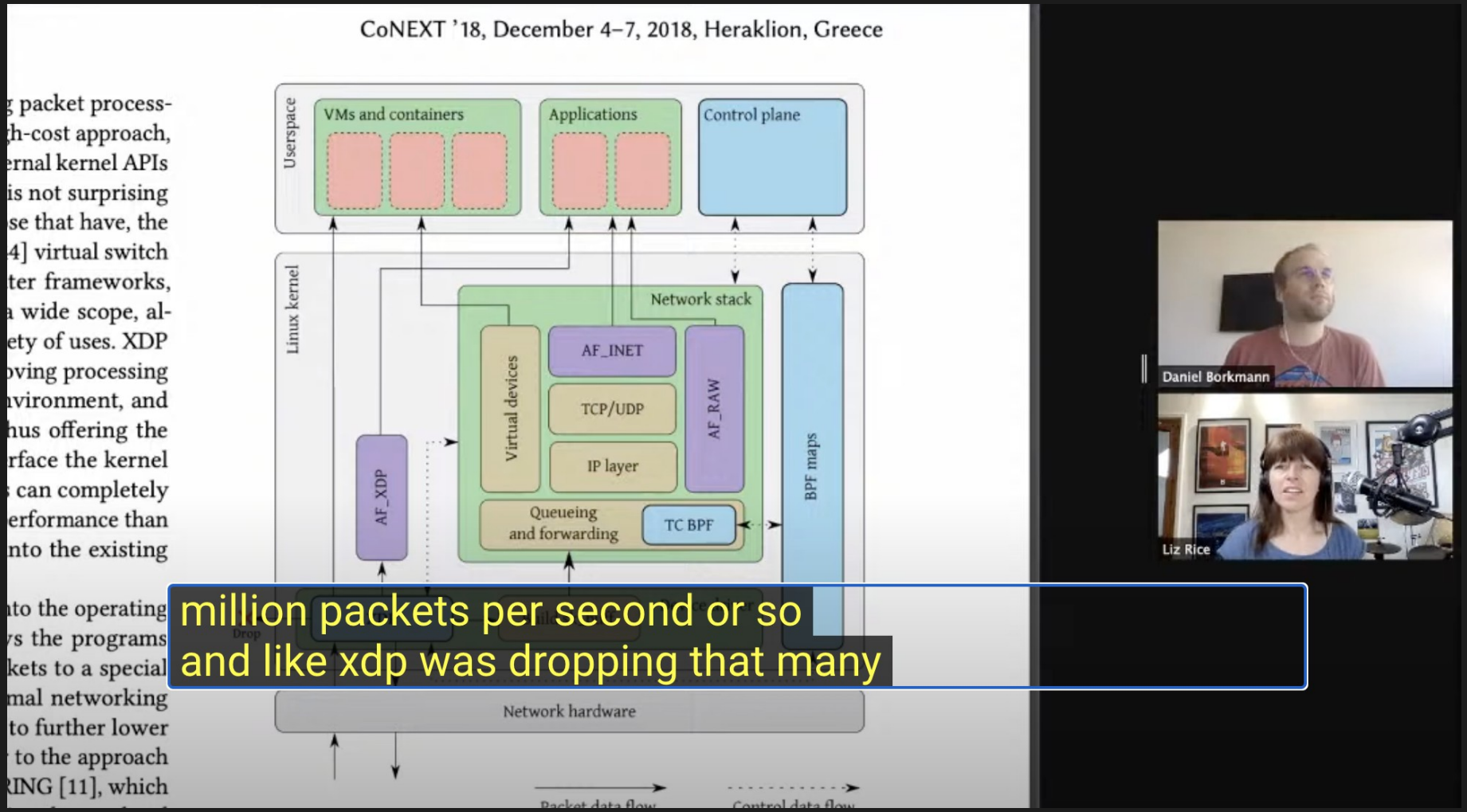
pkg.go.dev/syscall

GO syscall package standard library

Constants

```
    SYS_INOTIFY_INIT1          = 294
    SYS_PREADV                 = 295
    SYS_PWRITEV                = 296
    SYS_RT_TGSIGQUEUEINFO      = 297
    SYS_PERF_EVENT_OPEN        = 298
    SYS_RECVMMSG               = 299
    SYS_FANOTIFY_INIT          = 300
    SYS_FANOTIFY_MARK          = 301
    SYS_PRLIMIT64              = 302
)
```

xdp ddos drop pkts



xdp ddos drop pkts

Droplet: DDoS Protection Framework

Programmability: abstract away interactions with user space

- Droplet handler: handles the dirty work
  - Runtime compilation
  - Kernel load/hook
- Different types of handlers
  - GenericHandler
  - IPHandler
  - PrefixHandler ...
- The user only needs to write BPF code in C

Contents

Demo
RBAC (ubuntu 3 nodes)
Network Policy (same)
eBPF/BPF (fedora / docker)
Cilium (minikube)

Demo
RBAC (ubuntu 3 nodes)

rbac

```
user@cp01:~$
user@cp01:~$ kn test
Context "kubernetes-admin@kubernetes" modified.
user@cp01:~$
user@cp01:~$ k create sa myaccount -n test
serviceaccount/myaccount created
user@cp01:~$
user@cp01:~$ k create role testadmin -n test --verb=get --resource=pod
role.rbac.authorization.k8s.io/testadmin created
user@cp01:~$
user@cp01:~$ k create rolebinding testadminbinding --role testadmin --serviceaccount=test:myaccount
rolebinding.rbac.authorization.k8s.io/testadminbinding created
user@cp01:~$
user@cp01:~$ k auth can-i get pods -n test --as=system:serviceaccount:test:myaccount
yes
user@cp01:~$
user@cp01:~$ k auth can-i create pods -n test --as=system:serviceaccount:test:myaccount
no
user@cp01:~$
user@cp01:~$
```

Demo
Network Policy (ubuntu 3 nodes)

network policy

```
user@cp01:~$ kubectl run web --image=nginx --labels="app=web" --expose --port=80
service/web created
pod/web created
user@cp01:~$ kubectl run --rm --labels="role=frontend-na"  -it --image=alpine test-$RANDOM  -- sh
If you don't see a command prompt, try pressing enter.
/ # wget -qO- http://web
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

```
user@cp01:~$ cat netpol-allow-labels.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-deny-all
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
    - from:
      - podSelector:
          matchLabels:
            role: frontend-na
user@cp01:~$
```

```
user@cp01:~$ kubectl run --rm --labels="role=frontend-xx"  -it --image=alpine test-$RANDOM  -- sh
If you don't see a command prompt, try pressing enter.
/ # wget -qO- --timeout=3 http://web
wget: download timed out
/ #
```

```
user@cp01:~$ cat netpol-allow-labels.yaml
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-deny-all
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
    - from:
      - podSelector:
          matchLabels:
            role: frontend-na
user@cp01:~$
```

Demo

eBPF/BPF (fedora / docker)

ebpf docker

```
pwd
/home/fedora/code/bpf_with_go

less Dockerfile

docker ps -a
docker rm 76b1
docker image ls
docker image rm 573

make
sudo docker run -it --privileged -v /sys/kernel/debug:/sys/kernel/debug "exec_scrape"
open another term, ssh, run cmds

#update cod
make
docker stop
sudo docker run -it --privileged -v /sys/kernel/debug:/sys/kernel/debug "exec_scrape"
```

ebpf docker

```
[fedora@fedora bpf_with_go]$ docker image ls
REPOSITORY                    TAG          IMAGE ID       CREATED         SIZE
exec_scrape                   latest       40eae72b44d1   9 seconds ago   3.52MB
<none>                        <none>       e0b0315859de   9 seconds ago   1.43GB
<none>                        <none>       cec2ba1929ea   3 minutes ago   1.43GB
<none>                        <none>       d47189e7c1f3   8 minutes ago   3.52MB
<none>                        <none>       69f22e335878   4 days ago      1.43GB
<none>                        <none>       010a5555d7f6   4 days ago      7.5GB
<none>                        <none>       e89228f2c12a   5 days ago      3.52MB
almalinux                     8.5          2e19f62a21cc   3 months ago    198MB
quay.io/cilium/ebpf-builder   1648566014   e10d558109c6   5 months ago    1.89GB
k8s.gcr.io/coredns/coredns    v1.8.6       a4ca41631cc7   10 months ago   46.8MB
k8s.gcr.io/pause              3.6          6270bb605e12   12 months ago   683kB
[fedora@fedora bpf_with_go]$ sudo docker run -it --privileged -v /sys/kernel/debug:/sys/kernel/debug "exec_scrap
On ***from devup cpu 03 containerd-shim ran : 3523 /usr/bin/runc
On ***from devup cpu 01 containerd-shim ran : 3529 /usr/bin/runc
On ***from devup cpu 02 bash ran : 3535 /usr/bin/clear
On ***from devup cpu 03 (sd-worker) ran : 3536 /usr/lib/systemd/systemd-userwo
On ***from devup cpu 03 (sd-worker) ran : 3537 /usr/lib/systemd/systemd-userwo
On ***from devup cpu 03 bash ran : 3539 /usr/bin/curl
On ***from devup cpu 03 bash ran : 3541 /usr/bin/git
On ***from devup cpu 01 bash ran : 3542 /usr/bin/su
On ***from devup cpu 03 bash ran : 3545 /usr/bin/su
```

ebpf docker

```
package main

//go:generate go run github.com/cilium/ebpf/cmd/bpf2go -target bpfel -cc clang gen_execve ./bpf/execve.bpf.c -- -I/usr/include/bpf -I.

import (
        "bytes"
        "encoding/binary"
        "fmt"
        "github.com/cilium/ebpf/link"
        "github.com/cilium/ebpf/perf"
        "golang.org/x/sys/unix"
        "log"
        "os"
)

type exec_data_t struct {
        Pid    uint32
        F_name [32]byte
"app/src/main.go" 67L, 1378B
```

ebpf c

```go
func main() {
	setlimit()

	objs := gen_execveObjects{}

	loadGen_execveObjects(&objs, nil)
	link.Tracepoint("syscalls", "sys_enter_execve", objs.EnterExecve, nil)

	rd, err := perf.NewReader(objs.Events, os.Getpagesize())
	if err != nil {
		log.Fatalf("reader err")
	}

	for {
		ev, err := rd.Read()
		if err != nil {
			log.Fatalf("Read fail")
		}

		if ev.LostSamples != 0 {
			log.Printf("perf event ring buffer full, dropped %d samples", ev.LostSamples)
			continue
		}

		b_arr := bytes.NewBuffer(ev.RawSample)

		var data exec_data_t
		if err := binary.Read(b_arr, binary.LittleEndian, &data); err != nil {
			log.Printf("parsing perf event: %s", err)
			continue
		}

		fmt.Printf("On ***from devup cpu %02d %s ran : %d %s\n",
			ev.CPU, data.Comm, data.Pid, data.F_name)
```

Contents

Demo

Cilium (minikube)

cilium

C

```
minikube version
# start
minikube start --network-plugin=cni --cni=fals --memory=4096
# install cilium
CILIUM_CLI_VERSION=$(curl -s https://raw.githubusercontent.com/cilium/cilium-cli/master/stable.txt)
CLI_ARCH=amd64
if [ "$(uname -m)" = "arm64" ]; then CLI_ARCH=arm64; fi
curl -L --fail --remote-name-all https://github.com/cilium/cilium-cli/releases/download/${CILIUM_CLI_VERSION}/cilium-darwin-${CLI_ARCH}.tar.gz{,.
shasum -a 256 -c cilium-darwin-${CLI_ARCH}.tar.gz.sha256sum
sudo tar xzvfC cilium-darwin-${CLI_ARCH}.tar.gz /usr/local/bin

# install
cilium install
cilium status
cilium hubble enable
cilium status

#install hubble client
export HUBBLE_VERSION=$(curl -s https://raw.githubusercontent.com/cilium/hubble/master/stable.txt)
HUBBLE_ARCH=amd64
if [ "$(uname -m)" = "arm64" ]; then HUBBLE_ARCH=arm64; fi
curl -L --fail --remote-name-all https://github.com/cilium/hubble/releases/download/$HUBBLE_VERSION/hubble-darwin-${HUBBLE_ARCH}.tar.gz{,.sha256s
shasum -a 256 -c hubble-darwin-${HUBBLE_ARCH}.tar.gz.sha256sum
sudo tar xzvfC hubble-darwin-${HUBBLE_ARCH}.tar.gz /usr/local/bin
#validate hubble api access
cilium hubble port-forward&
#
hubble status
hubble observe
# enable hubble ui
cilium hubble enable -ui
cilium hubble ui
```

cilium

cil

localhost

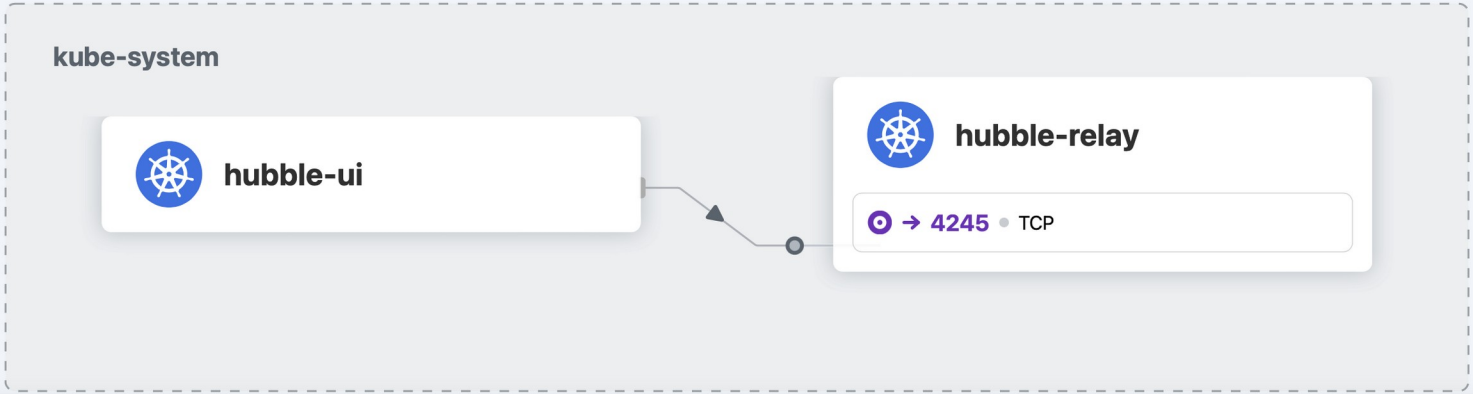
kube-system

Filter by: label key=val, ip=1.1.1.1, dns=google.com, identity=42, pod=frontend

Any verdict

Visual

6.2 flows/s • 1/1 nodes



Columns

Source Service	Destination Service	Destination Port	L7 info	Verdict	Timestamp
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 5 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 10 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds
hubble-ui *kube-system*	hubble-relay *kube-system*	4245	—	forwarded	less than 20 seconds

techbeacon.com/security/tesla-drives-cryptojack-gangs-aws-cloud-down-kubernetes-avenue

App Dev & Testing | Enterprise IT | Security | GUIDES | CONFERENCES | SUBSCRIBE

Home / Security / Information Security

Security Blogwatch

Tesla drives cryptojack gang's AWS cloud down Kubernetes avenue

Richi Jennings
Your humble blogwatcher, dba RJA

A Tesla-owned AWS account was hacked to mine Monero. Who knows when; who knows for how long. Shocking.

The hackers drove straight in using an "unsecured" Kubernetes admin console (i.e., it had no password). They ran the *WannaMine* fileless APT on auto-pilot—and might also have stolen some sensitive data from Tesla drivers.

It's an electrifying story. In this week's *Security Blogwatch*, we're charged with plugging in the route of the tale, and bringing you up to speed.

Accessible URL

Admin Roles

cuckoo’s game

Hardback copies of my first book, *The Cuckoo's Egg*. Hardback. New old stock, unread. 323 pages printed on the finest quality 55 pound Glatfelter paper. *ALL GONE* *The last one left here on July 13, 2014 -- and my wife is delighted to reclaim our closet!*

This book is straight from 1989, back before my hair turned gray. It's the true story about how I caught a spy over the early Internet.. Some of the technology feels antique (2400 baud modems), and the author feels a quarter century older. So get a copy while I'm still able to sign it..

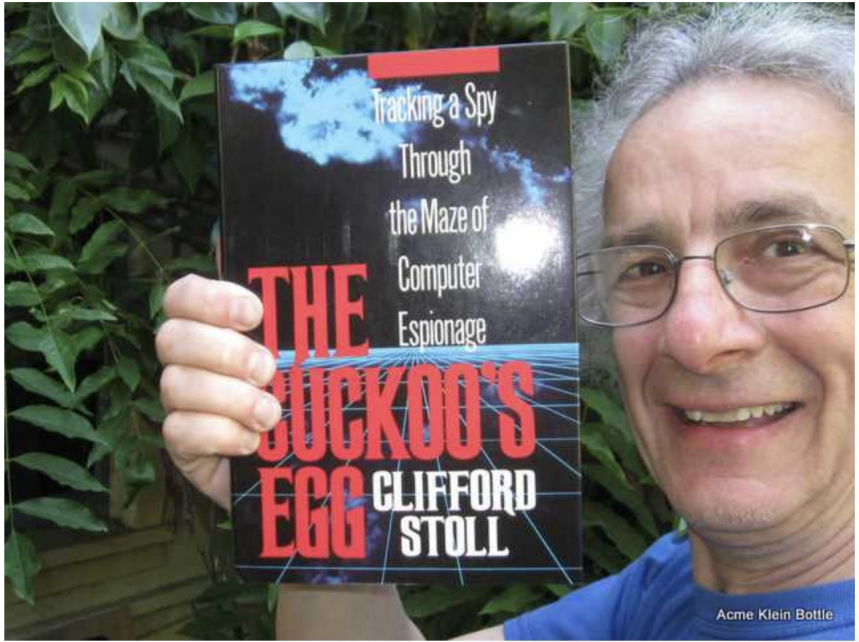
By default, I will autograph this book to you. Alternatively, I'm happy to sign it to a friend of yours, or leave it unsigned. (tell me what to do in the comments section of the order form)

Send me email if you want a copy of Cuckoo's Egg in another language - I have a few in Czech and Dutch; possibly one or two in Spanish and Chinese, and maybe a couple other languages.

Each of these books comes complete with book jacket, 323 pages, lots of nouns, verbs, prepositions, and even a few participles (whatever thosea are). I guarantee that I wrote the whole book (six months of pecking on a MacPlus during 1988)

If you want 10 copies, oh, do I have a deal for you! (yep, I have about two dozen of these books, still in their original boxes) *SOLD - GONE - NO MORE*

Buy a copy of The Cuckoo's Egg and make my wife really happy...



```
user@cp01: ~
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.4 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
user@cp01:/var/lib$ k -n kube-system get pods
NAME                              READY   STATUS    RESTARTS       AGE
etcd-cp01                         1/1     Running   19 (29m ago)   24m
kube-apiserver-cp01               1/1     Running   18 (27m ago)   24m
kube-controller-manager-cp01      1/1     Running   1 (32m ago)    24m
kube-scheduler-cp01               1/1     Running   1 (32m ago)    24m
user@cp01:/var/lib$
```

Minikube
Vagrant
VMWare/VirtualBox

Others Cloud / Labs
Online free first taste

```
KIND:     Pod
VERSION:  v1

RESOURCE: spec <Object>

DESCRIPTION:
     Specification of the desired behavior of the pod. More info:
     https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

     PodSpec is a description of a pod.

FIELDS:
   activeDeadlineSeconds        <integer>
     Optional duration in seconds the pod may be active on the node relative to
     StartTime before the system will actively try to mark it failed and kill
:
```

```
Every 2.0s: kubectl -n2 -n kube-system get pods,deploy,svc --show-labels -o wide                    cp01: Tue May 10 22:41:22 2022

NAME                                 READY   STATUS    RESTARTS       AGE   IP             NODE   NOMINATED NODE   READINESS GATES   LABELS
pod/etcd-cp01                        1/1     Running   19 (29m ago)   25m   192.168.0.38   cp01   <none>           <none>            component=etcd,tier=control-plane
pod/kube-apiserver-cp01              1/1     Running   18 (27m ago)   25m   192.168.0.38   cp01   <none>           <none>            component=kube-apiserver,tier=control-pl
ane
pod/kube-controller-manager-cp01     1/1     Running   1 (32m ago)    25m   192.168.0.38   cp01   <none>           <none>            component=kube-controller-manager,tier=c
ontrol-plane
pod/kube-scheduler-cp01              1/1     Running   1 (32m ago)    25m   192.168.0.38   cp01   <none>           <none>            component=kube-scheduler,tier=control-pl
ane
```

```
Every 2.0s: kubectl config get-con...  cp01: Tue May 10 22:41:23 2022

CURRENT   NAME                          CLUSTER      AUTHINFO
   NAMESPACE
*         kubernetes-admin@kubernetes   kubernetes   kubernetes-admin
```

```
  1  [|||                    9.0%]   Tasks: 58, 152 thr; 1 running
  2  [||                     6.2%]   Load average: 0.59 0.42 0.37
  Mem[||||||||||||851M/3.81G]   Uptime: 00:31:33
  Swp[                    0K/0K]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+
   5022 root       20   0 1084M  310M 74488 S  4.2  7.9  1:18.62
   1701 root       20   0  805M   97M 62888 S  2.1  2.5  0:32.02
   4666 root       20   0 10.7G 48664 21504 S  2.1  1.2  0:29.77
   4870 root       20   0 1915M   99M 67592 S  2.1  2.6  0:35.24
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +
```

```
[1] 0:bash*                                                                                  "cp01" 22:41 10-May-22
```

Art of wars

Know sylf know thee, hund battles hund wins.
Know sylf not thee, half win.
Know not sylf nor thee, lose all.

Winning k8s sec without fighting

Nature
Opportunity
Environment
Leadership
Tools

Public domains
Open source codes/tools

```
func DoNext(b ...labs, t ...tools, i ...ideas) {
    defer SeeUsAfter()
    go ThankDevUPMgrGTL(){ <-BetterJobs }()
}
```

end

$$S = f(K, S, T)^{p}$$

Success k8s sec = f (Knowledge, Skills, Tools) power of people