

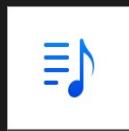
& C

D V P

Engineering

**What are benefits 1/2/3
how to build/implement/upgrade k8s 1/2/3
Everyone should use it
If not used you behind the technology**

Kubernetes symphony



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

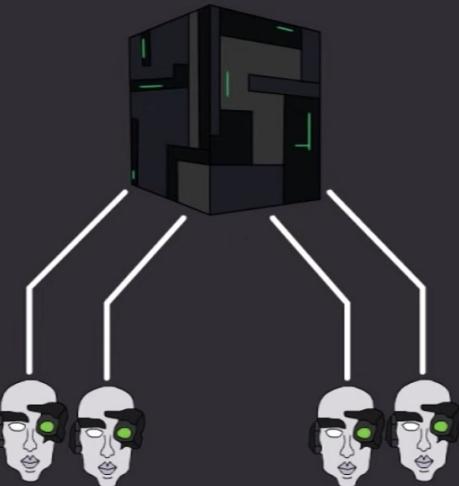
intro

Kubernetes securities from built-in features to open source tools

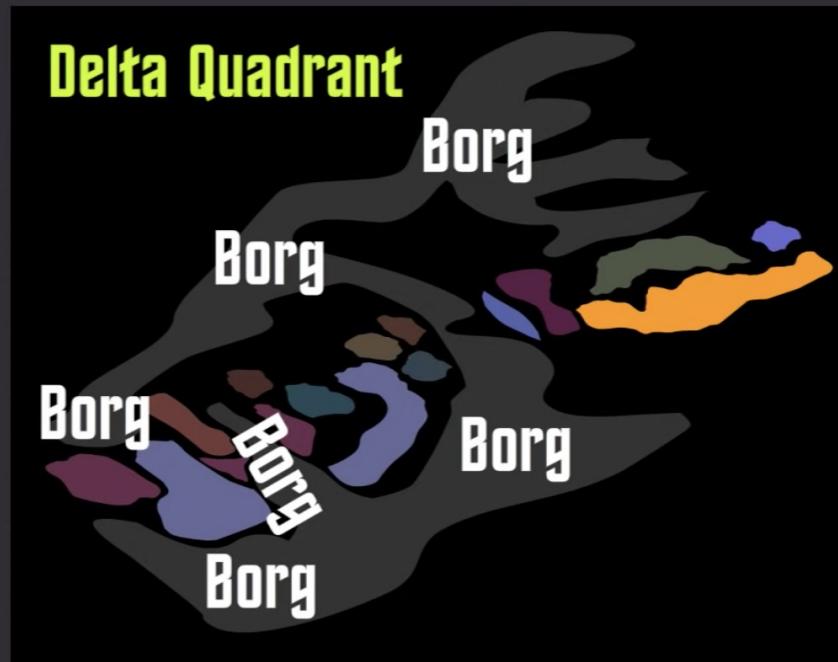
By a,b,c

Contents

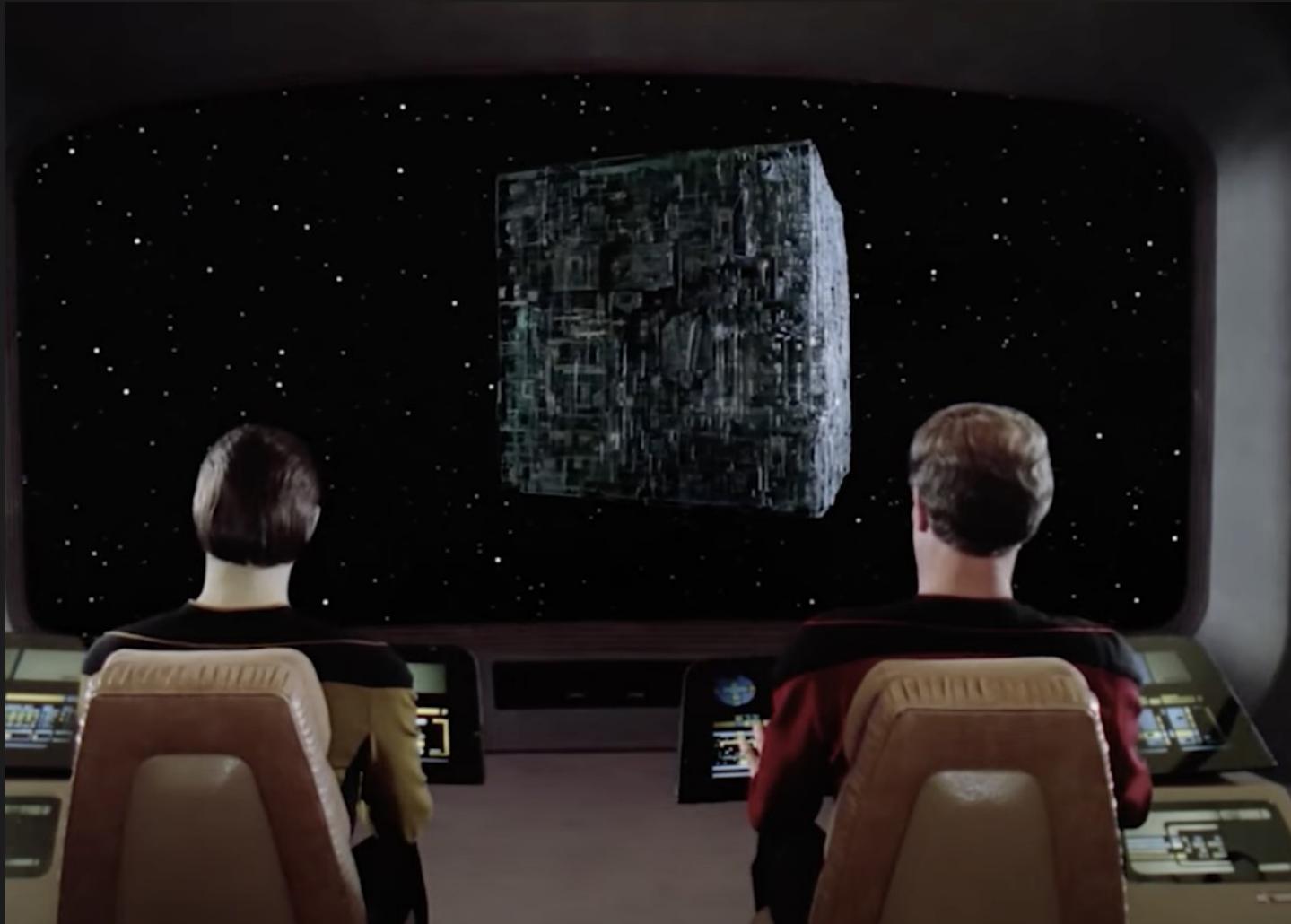
**About k8s
k8s security
Demos**



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

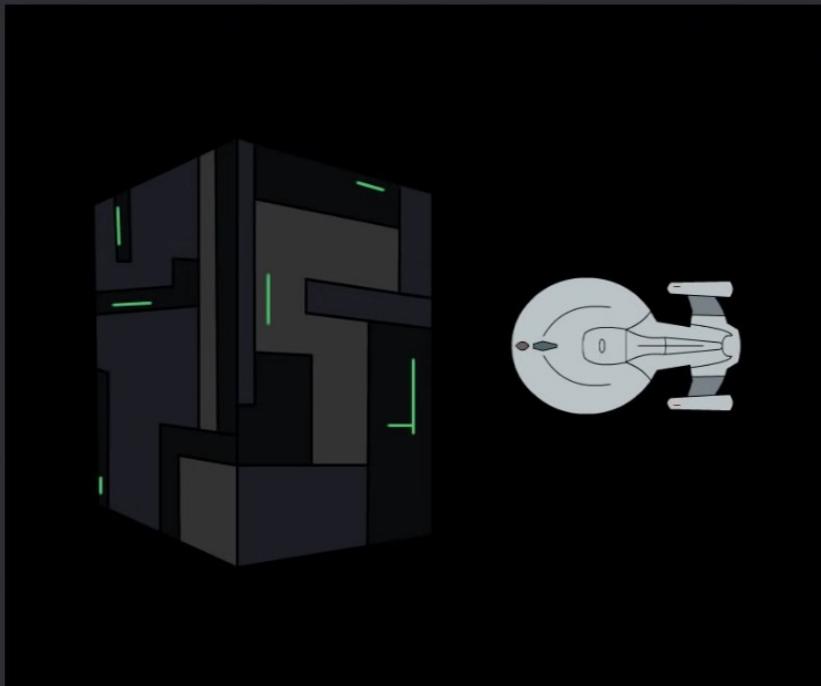
Borg Won the War
Voyager Escaped
Borg's Betrayal



Seven of Nine
Tertiary Adjunct of
Unimatrix Zero One

Borg-Species 8472 War

2373-2374





first-commit 1 branch 0 tags

[Go to file](#)[Code](#)

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



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

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

Used by 37



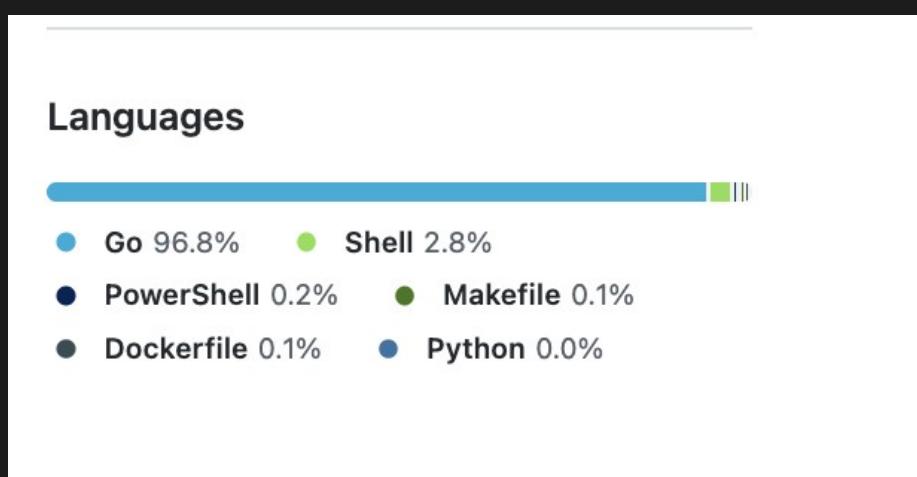
+ 29

Contributors 3,228



+ 3,217 contributors

k8s first commit on June 7 2014



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

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

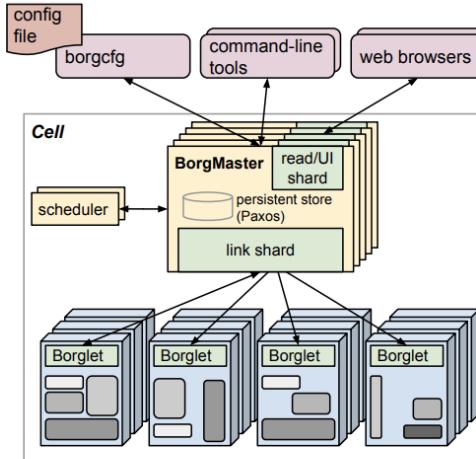


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

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

User view

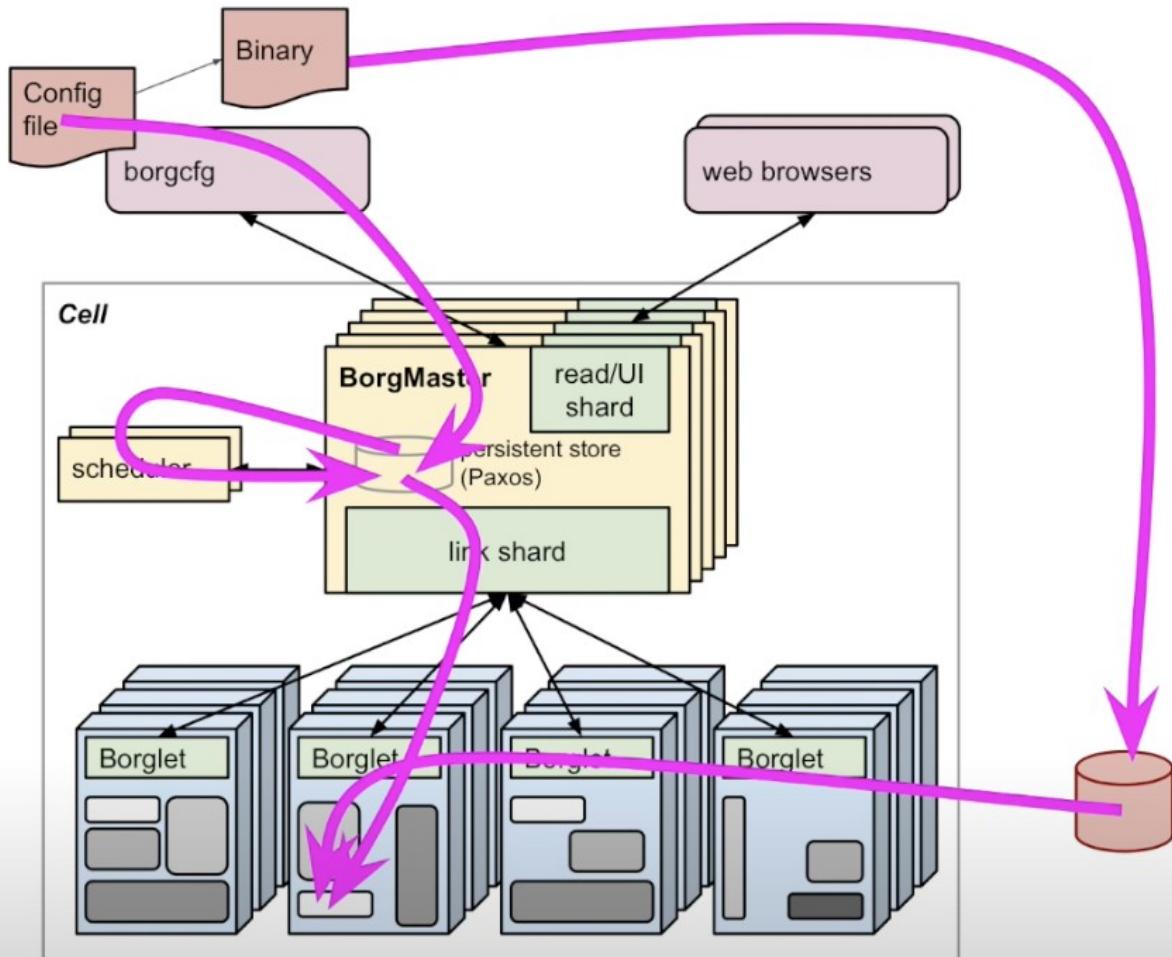
```
job hello_world = {  
    runtime = { cell = 'ic' }                      // Cell (cluster) to run in  
    binary = '.../hello_world_webserver'           // Program to run  
    args = { port = '%port%' }                     // Command line parameters  
    requirements = {     // Resource requirements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



10 years how

User view

What just
happened?





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

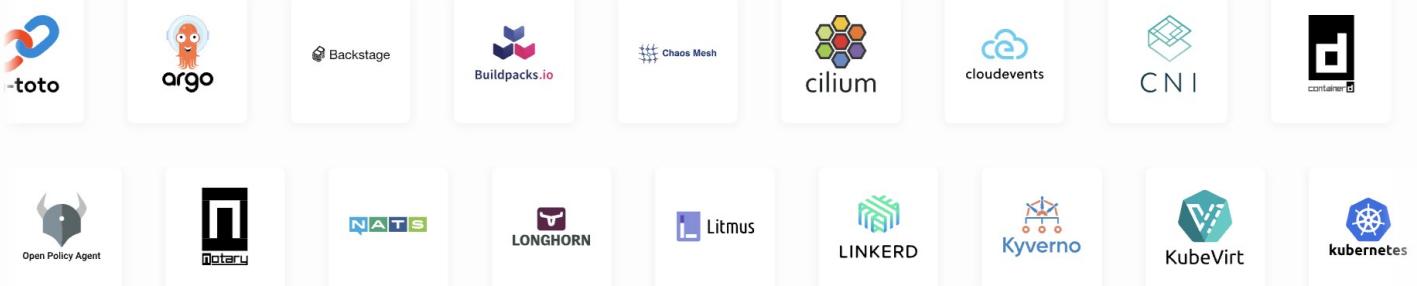
About Projects Training Community Blog & News Join 

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](#)



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 spirit, 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 spirit, 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

Contents

k8s security

k8s is not a security system

Sec wasn't built in natively

k8s sec improving (1.22+)

K8s sec

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](#)

- 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

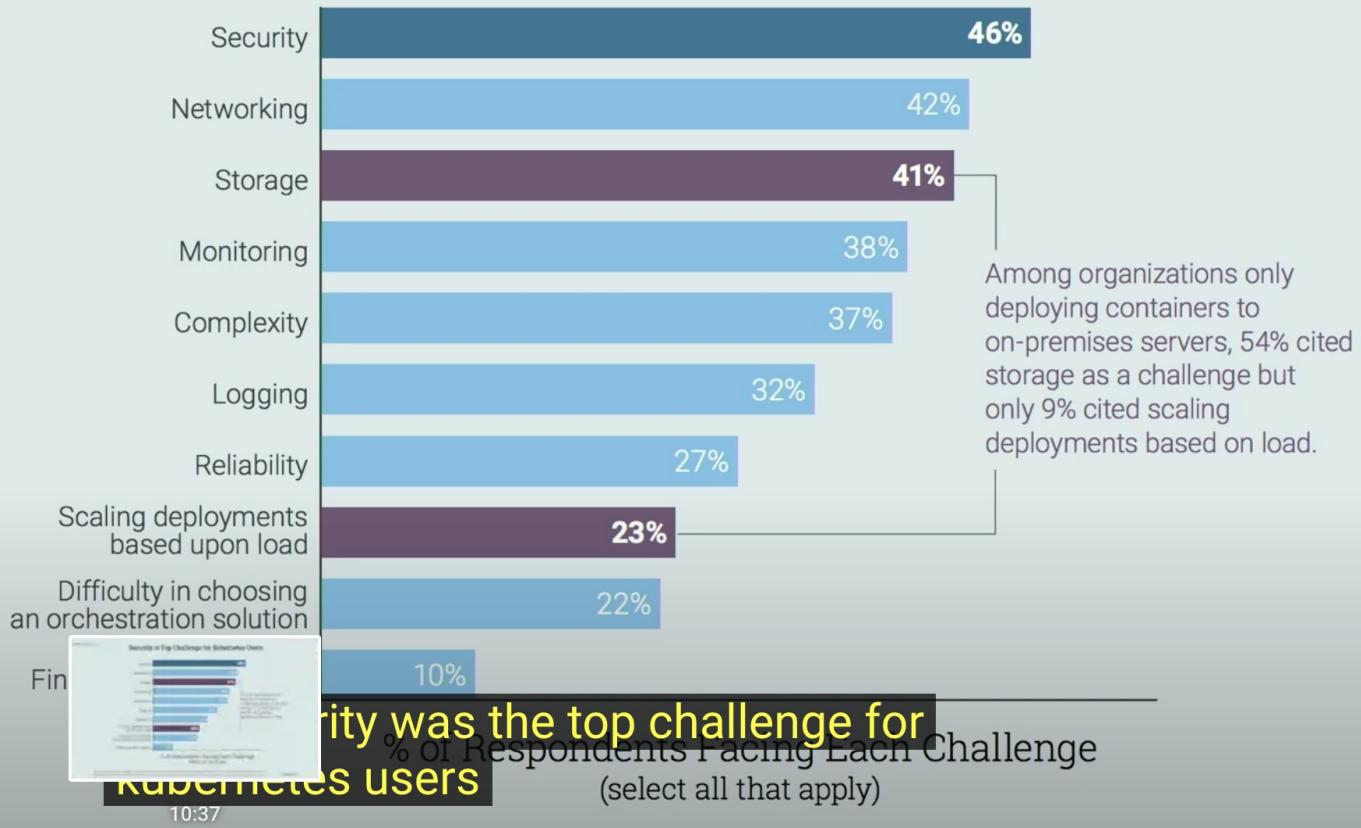
Secure Kubernetes with Aqua

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.



Security is Top Challenge for Kubernetes Users



Source: The New Stack Analysis of Cloud Native Computing Foundation survey conducted in Fall 2017. Q. What are your challenges in using/developing 0:36 / 41:34 (check all that apply). n=527. Note, only respondents managing containers with Kubernetes were included in the ch





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**

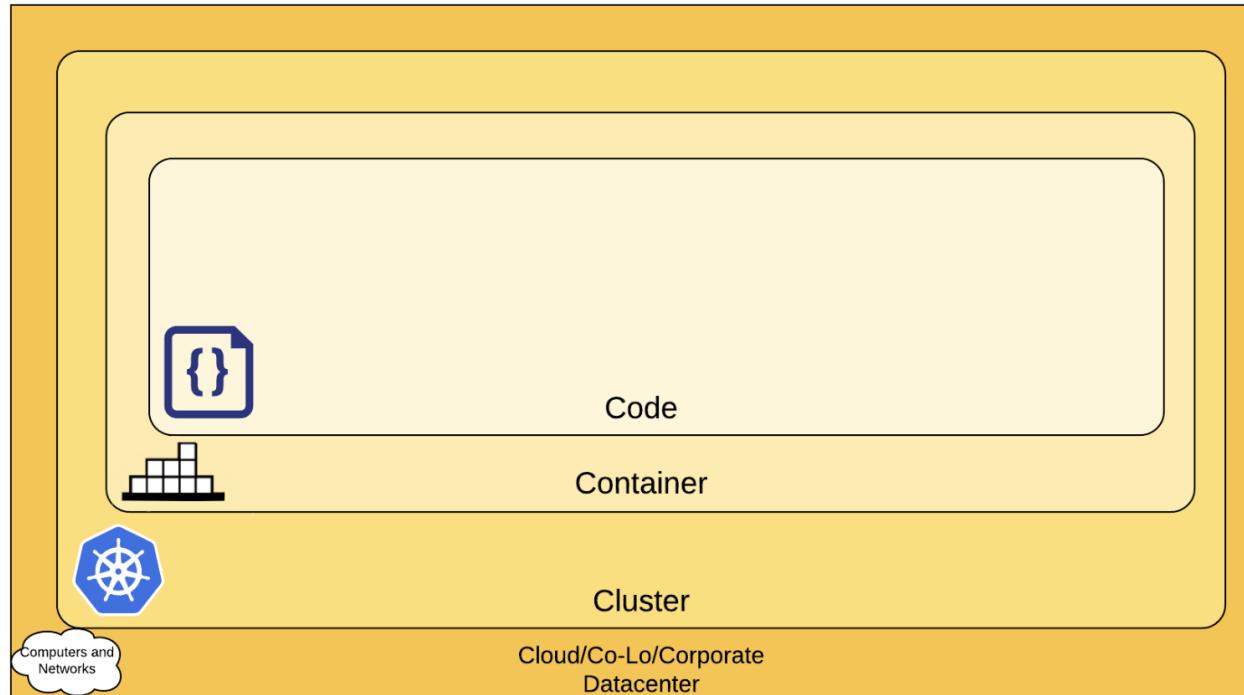
This comes as no surprise: 38% of respondents suggest that security is the most important consideration whether when operating Kubernetes, building container images or defining an edge strategy. Keeping clusters up-to-date is a definitive best practice to solve security issues. However, according to 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.

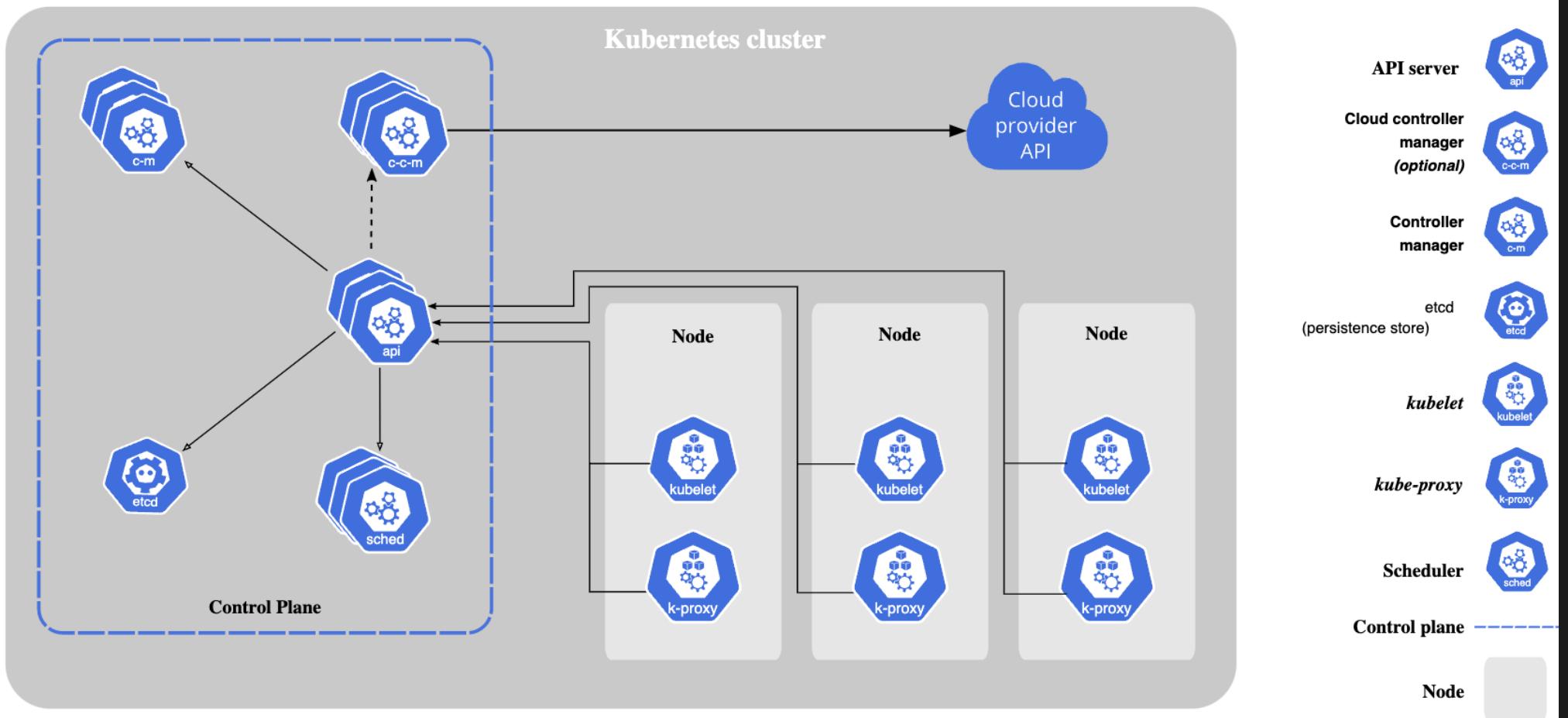
4c

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

cni install control plan

Apiserver

Kubelet

Schedulercontroller

Enable RBAC

Enable admission control (image pull policy, pod security standard)

Enable TLS

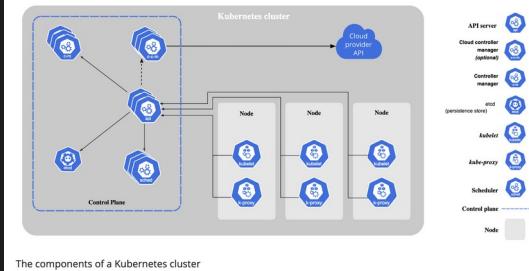
Remove anonymous

Network Policy

Secrets / ConfigMaps

etcd tls

etcd encryption



The components of a Kubernetes cluster

k8s node sec

Kubelet auth

Kubelet TLS

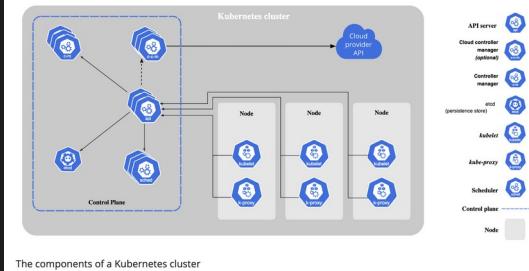
Kubelet config remove anonymous

Labels

Taints

Affinity

Resource Quotas

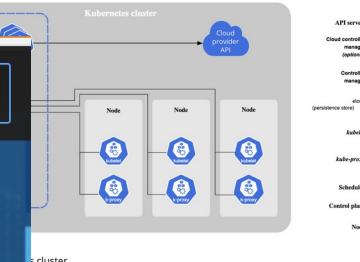


The components of a Kubernetes cluster

YouTube GB

sans cloud security

X Q SIGN IN



Upgrade the Cluster

- Even privileged pods became harder for an attacker to use in more recent Kubernetes versions.
- Version 1.22 massively de-privileged the kubelet.
- Recent versions moved service account tokens from normal volumes to "projected volumes," provided by tmpfs.
- Kubernetes defaults get stronger and stronger, thanks to the project's information security-focused members.

Beale

40

INGUARDIANS™

Jay Beale / InGuardians

CloudSecNext 2022 SANS SUMMIT

35:36 / 35:59

◀ ▶ ⏸ 35:36 / 35:59

II ⏹ ⚙️ ⏷ ⏹ 🔍

Kubernetes Attack and Defense: Break Out and Escalate!

The screenshot shows a web browser displaying a blog post from the Kubernetes website. The URL in the address bar is kubernetes.io/blog/2021/08/04/kubernetes-1-22-release-announcement/. The page title is "More secure control plane with kubeadm". The main content discusses a new alpha feature for running kubeadm components as non-root users. It also mentions a new v1beta3 configuration API and the deprecation of v1beta2.

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

A screenshot of a web browser window displaying a blog post from the Kubernetes website. The URL in the address bar is kubernetes.io/blog/2021/12/07/kubernetes-1-23-release-announcement/. The page header includes the Kubernetes logo and navigation links for Documentation, Kubernetes Blog, Training, Partners, Community, Case Studies, Versions, and English language selection. On the left sidebar, there are links for the years 2022 and 2021, and categories like Kubernetes-in-Kubernetes and 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).



[Documentation](#) [Kubernetes Blog](#) [Training](#) [Partners](#) [Community](#) [Case Studies](#) [Versions ▾](#) [English ▾](#)

 Search

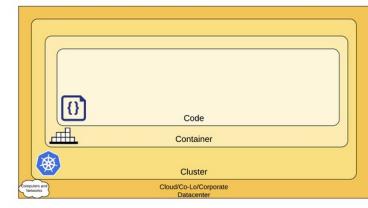
2022

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](#).

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.



!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

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

open source tools:

code: go simple, readability, promise backward compatibility, static, compiled, grpc, no repo

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

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

philosophical thinking

1 computer

1 network

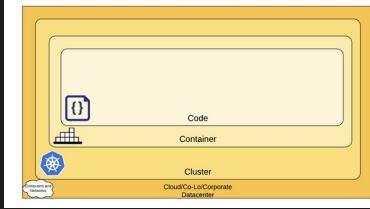
1 tool

EBPF

XDP

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.



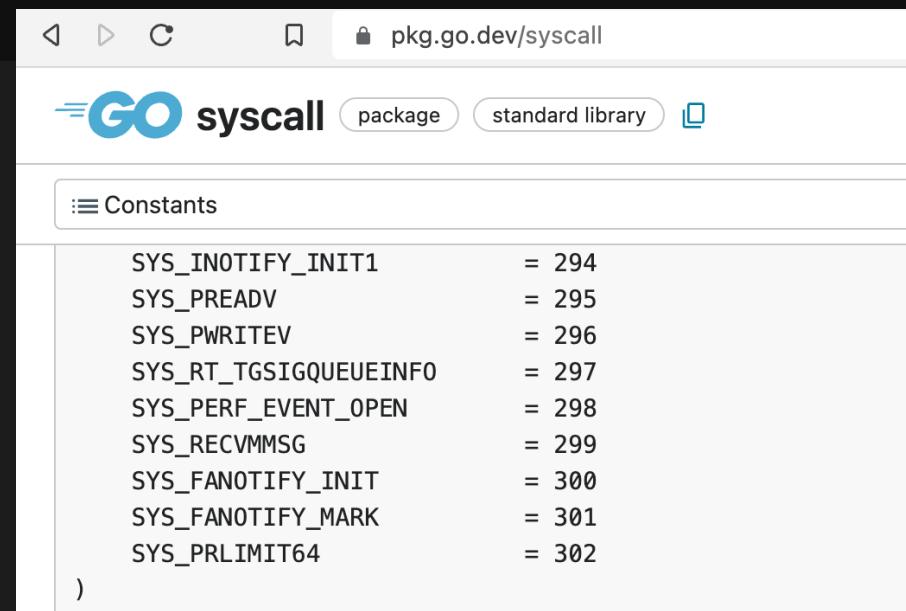
ebpf



Eye of Sauron

syscalls

```
user@w02:/usr/src/linux-headers-5.15.0-46/include/linux$ grep "asm linkage 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$
```



The screenshot shows the GoLand IDE interface with a search results window. The title bar says "pkg.go.dev/syscall". The main area displays the "syscall" package with the "standard library" tab selected. A table lists various constants:

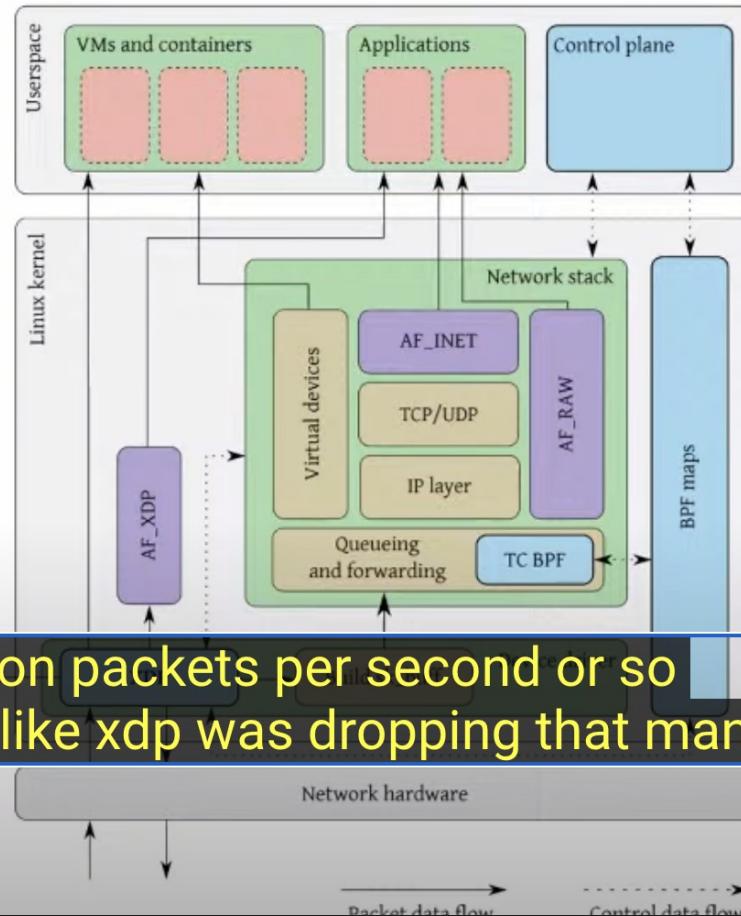
	Constant Name	= Value
	SYS_INOTIFY_INIT1	= 294
	SYS_PREADV	= 295
	SYS_PWRITEV	= 296
	SYS_RT_TGSIGQUEUEINFO	= 297
	SYS_PERF_EVENT_OPEN	= 298
	SYS_RECVMMMSG	= 299
	SYS_FANOTIFY_INIT	= 300
	SYS_FANOTIFY_MARK	= 301
	SYS_PRLIMIT64	= 302
)		

xdp ddos drop pkts

g packet processing approach, which is a high-cost approach, as it requires external kernel APIs. This is not surprising, as most of the frameworks that have, the exception of [4] virtual switch and bridge frameworks, have a wide scope, allowing for a variety of uses. XDP provides a low-cost packet processing environment, and thus offering the opportunity to interface the kernel directly. This can completely bypass the existing performance bottleneck into the existing

into the operating system. The programs send packets to a special queue in the kernel networking stack to further lower the latency of the approach. This is similar to the approach used by RING [11], which

CoNEXT '18, December 4–7, 2018, Heraklion, Greece



Drop million packets per second or so
and like xdp was dropping that many



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

Demo

RBAC (ubuntu 3 nodes)

Network Policy (same)

eBPF/BPF (fedora / docker)

Cilium (minikube)

Contents

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:~$ █
```

Contents

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:~$
```

Contents

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

```
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)

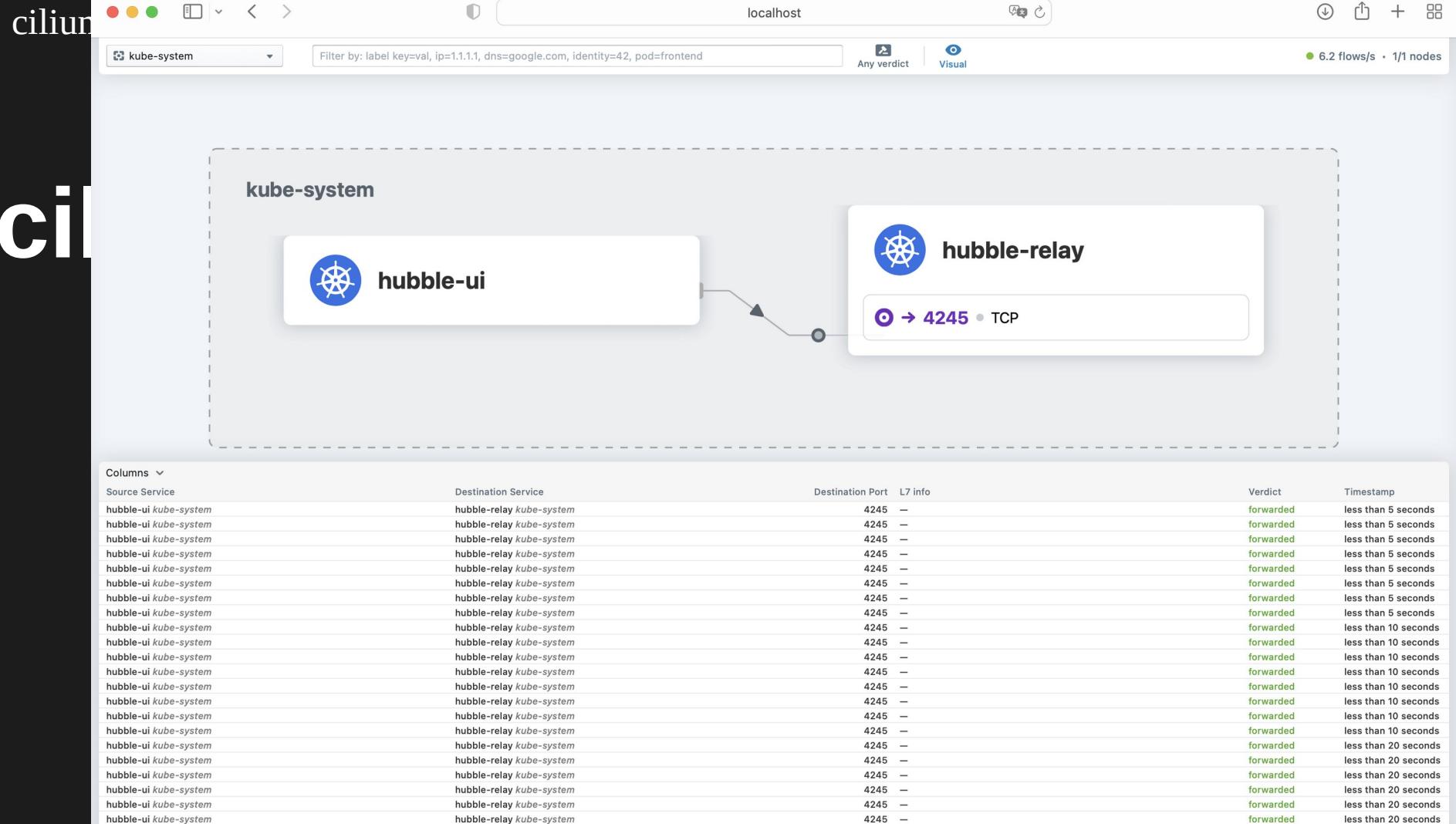
cilium

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{,.sha256sum}
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{,.sha256sum}
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
```





[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..

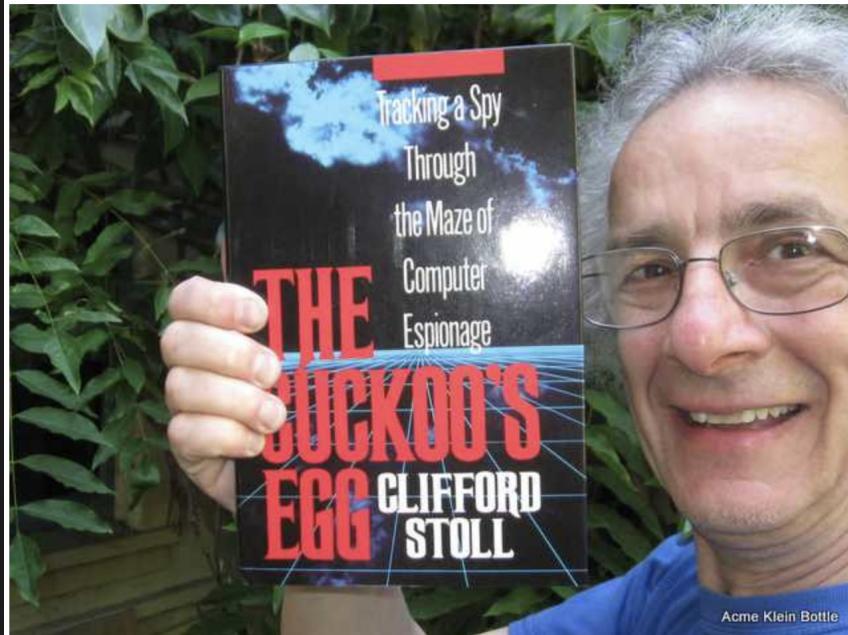
By default, I will autograph this book to you. Alternatively, I'm happy to sign it to a friend of yours, or leave it unsigned. (tell me what to do in the comments section of the order form)

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...



```
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$
```

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

:

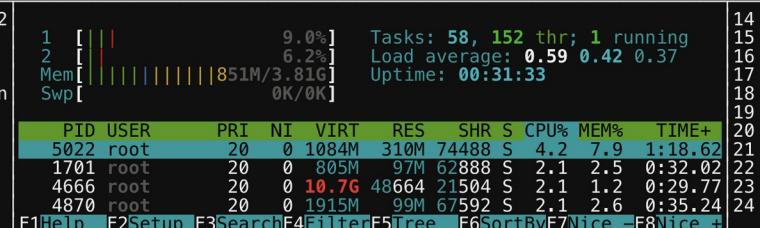
Every 2.0s: kubectl -n2 -n kube-system get pods,deploy,svc --show-labels -o wide

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-plane
pod/kube-controller-manager-cp01	1/1	Running	1 (32m ago)	25m	192.168.0.38	cp01	<none>	<none>		component=kube-controller-manager,tier=control-plane
pod/kube-scheduler-cp01	1/1	Running	1 (32m ago)	25m	192.168.0.38	cp01	<none>	<none>		component=kube-scheduler,tier=control-plane

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



hund battles

Art of wars

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

thoughts / philosophy

Winning k8s sec without fighting

Nature
Opportunity
Environment
Leadership
Tools

sources learning

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