SUMMARY

Devoxx 2022

Day 1

> **Build and deploy microservices on Kubernetes with Istio**

Istio

#istio #k8s #serviceMesh #java

What's istio

- Secure service to service communication
- Service discovery
- Load balancing
- Routing / fail overs
- Policy layer access control rate limiting
- Automatic metrics

Istio architecture

- Control plane (config etc)
- Data plane (pod / consul ingress / egress)

Demo

- Prepare a cluster (4 nodes min + cpu++ recommended)
- Install istio : Single curl ? -> istioctl to manage & install istio Getting Started
- **Egress can be skipped if no external traffic is needed**
- Install Kiali / Zipkins + others created in the istio namespace (prometheus/grafana)
- JDL = JHister Domain language
  - Used to define a bunch of random stuff and create all kube ressources
- Istio proxy = consul proxy
- Manual deploy = happen at the deployment level
- Automatic deploy = pod level

Virtual services from istio

Define how trafic is routed to apps
Istio can merge similar routing rules (can do weird things sometimes)

Destinations rules from istio

Applied after routing is decided from istio. (Load balancer / circuit breaking etc)

Gateways from istio

Can be ingress or egress gateway

Istio service entries

Manage service external to the mesh

Sidecars

- Fine-tune the set of ports and protocols that an Envoy proxy accepts.
- Limit the set of services that the Envoy proxy can reach.

Delay & Fault

- Can specify a delay on virtual service to test how the mesh is behaving with bad response time
- Can specify a fault on virtual service to test how the mesh is behaving with lot of faulty responses

Auth / Security

- Can do MTLS all over the data plane for E2E traffic encryption
- Can handle oauth (easier to use a plugin)

Is a service mesh worth it ?

Pros

- kube native microservice
- Reduced resp for dev
- No need to write / maintain code
- AB canaray release etc

Cons

- Complex to debug
- Higher resource usage / running costs
- Business logic related policies might be trickier

Slides

Blockchain, NFT and smart contracts for my (Java) application

Blockchain / NFT / Smart Contracts

#blockchain #nft #smartcontract

Learn

- Blockchain
- Token
- NFT
- Smart Contract
- Public Ledger

44% own some crypto
57% Never used a blockchain

Cryto history

Goals

- less transaction cost
- cryptographic proof

The creator(s) own 1m btc.
3-4m BTC are already lost.

A wallet is just a public key(receive) / private key(make transactions/send) pair.

btc v1 - eth v2 (mostly bs)

EVM can execute code (smart contracts)

Presenting AXIE INFINITY (a game on eth)

Whole things is bias, those guys are from hedera and comparing apples and bananas.

I've leaved the conf, didn't liked it because mostly bias and not that interesting.

Let's kutomize with style

Kustomize with style

#k8s #kustomize #sops

Two ways to deploys :

- Pull from repository
- Push to the cluster through CI/CD

Helm Hell - Helm is not the solution

Kustomize

Peace of mind features

- Decoupe par app
- Then create recipe kustomization and include the list of yaml to deploy
- Advanced features included in kustomize cli that aren't available in kubectl
- Kustomize can generate configmaps (configMapGenerator)
  - A digest of the content is added at the end of the name of the configmap
- Set the namespace in the kustomization
- Can add suffix prefix
- List images
- Replicas
- Can use helm chart in kustomize

Deploy multiple different applications

- Overlays : override the base kustomize
  - Can reference a repository

- PatchesStrategicMerge : Only need to define the minimum info needed to override a definition
- Patches: define a path / value / ... an apply it to a ressource. (Mostly ingress)
- Components : provide extra features on demand (ex: openldap)
  - Remove duplication on specific features set needed
- Can extend kustomize with go (yay)
- Transformers (ex get a secret from sops and deploy a clear version)

Helm vs. Kustomize the frenemies that soothe Kubernetes

Helm vs. Kustomize: the frenemies that soothe Kubernetes

#k8s #kustomize #helm

Tools

- #Dekorate
- #Jkube

Demo

Deploy #cassandra & #zipkins
Kustomize can add commonLabels

Best pratices

keep custom ressources and instances in separate packages

```
kubectl kustomize cfg fmt file_name
```

Demo helm

Manage env with multiple values files
Avoid hardcoding
Always try dry install

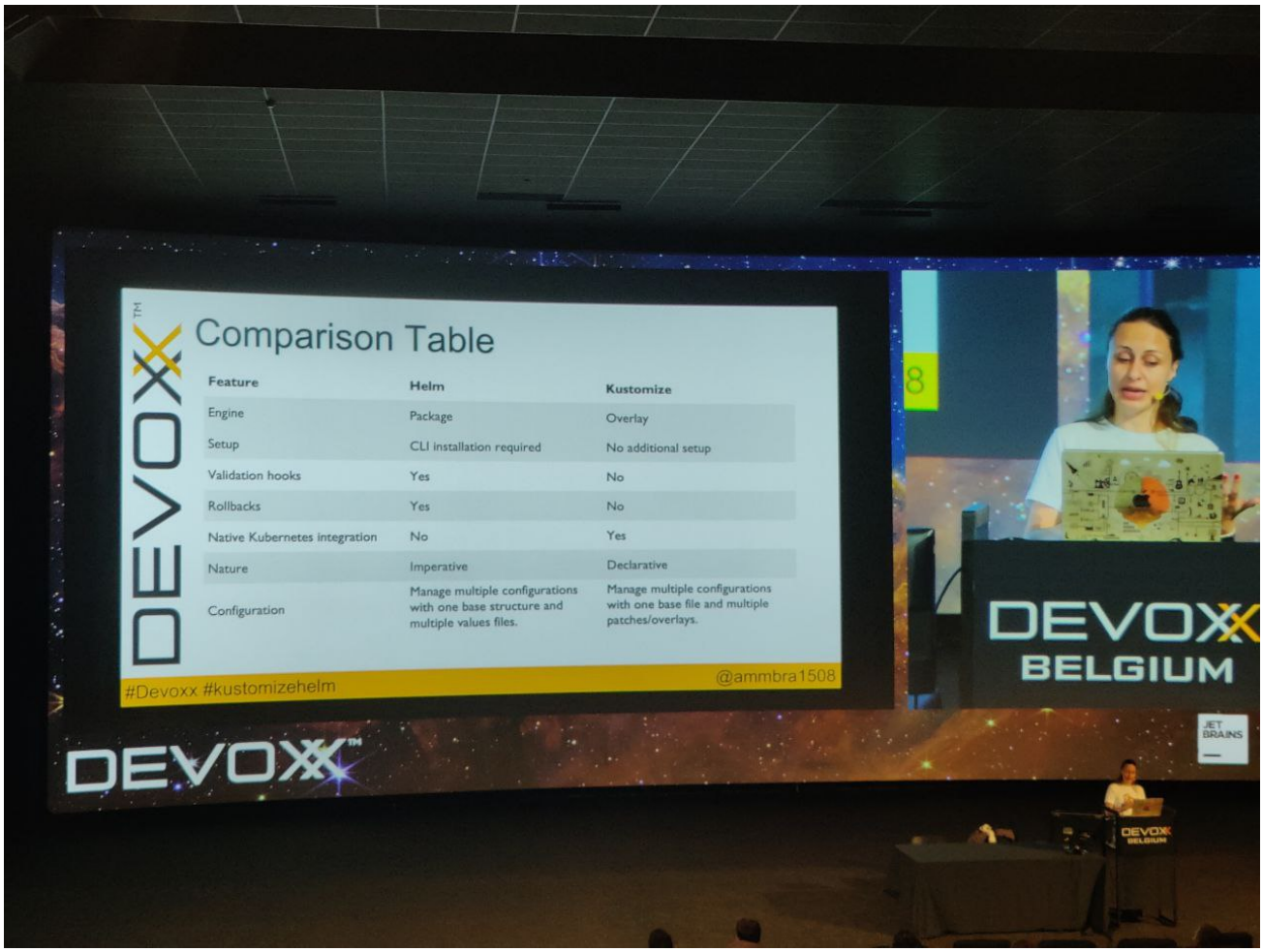
Demo helm+kustomize

Create yaml with helm then supercharge with kustomize
Kustomize build ––enable-helm

helmChartInflationGenerator

- Override helm charts with kustomize

Comparison table



Github

Personal note from the speaker : It's easier to generate then refine the files

OpenTelemetry – An Observability Framework for Cloud-Native Software

OpenTelemetry

#open-telemetry #cloud

What is observability ?

Detect -> Investigate -> Remediate
Observability tells you why it isn't working

Logs -> Traces -> Metrics -> ... (repeat)

Few implemented traces (may help debugging users errors not logged bc of N scenarios)

What is OpenTelemetry ?

A collection of tools / APIs / SDKs.
You can use it to instrument / generate / collect / export telemetry data (metrics, logs, traces).

Demo

Blog Post

Day 2

Istio 0 to 60

Istio 0 to 60

#k8s #serviceMesh #istio

Hands-on lab

Lab

How to secure a Kubernetes cluster from scratch?

How to secure a Kubernetes cluster from scratch?

#k8s #security #open-policy-agent

Managing a Kubernetes Cluster

Attack vectors at multiple levels.

- App
- App runtime
- Base image
- Container runtime
- Kube cluster

Critical components to protect (etcd/kubelet/etc...)

Kube attack matrix by Microsoft

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	

CIS Kubernetes Benchmark (heavy to read)

Detect config errors with #kube-bench

- Permissions on sensitive files
- Sensitive features enable etc...

Can be run in a container or installed on the master directly.

#Popeye : analyse ressources deployed on the cluster

Demo using #Krew manager plugin

#Kubehunter : pentest your cluster

- Can check from a external machine,
- On a cluster node
- From a pod

Focus on RBAC / Deployment control / Network policy

RBAC :

Users -> RoleBindings -> Roles -> Ressources -> Verbs
Demo

Limitations : Allow / Deny

Control Deployments

With securityContext either at the pod level or container.
Only work at runtime.

PodSecurityAdmission

Creation Time
Modes :

- Enforce : Strict application
- Audit : Will log in the audit log
- Warn : Warning message after deployment
  Levels :
- Privileged : No restriction
- Baseline : Host isolation / Privilege escaltion / limit capabilites
- Restricted : run as non root

Can be defined at cluster level

Analyse the impact of a policy with a dry run

root user needed only when :

- Changes on host
- Volumes access
- Dependencies
- Privileged ports
  Check bitnami doc to build non root images

Distroless images

- Reduce surface attack
- Process isolation

- Protect secrets

Network policies

Isolate pod / namespace between them.
Deny everything then allow what's necessecary

Use a CNI that support Network Policies (#Calico / #Cilium / #Kube-router...)

Enrich network policies with flexible def. (deny outbound traffic based on namespace / deny traffic on some ports depending on pods)

Some CNI offer L7 filtering (http methods & path)

- #istio RBAC
- CiliumNetworkPolicy

Admission Controllers

Can plug own code with webhook

- Prefered solution : Open policy agent
  - Use Rego language
  - Policy as code
  - Declarative & context-based
  - Fine-grained governance
  - Agnostic and integrated with Kubernetes
  - Define policy in config maps in Rego
    - Input based policies
    - Context based policies
- Other solution own code

Compliance pipeline :

- Kubeval : check if yml is well written
- Conftest : Fetch existing policies from the cluster at runtime to evaluate the new yml deployments

Unit tests pipeline :

- opa debug image : use command test to run the unit tests
  - https://play.openpolicyagent.org

Monitor OPA to make sure it doesn't run slow and slow down your network. (mandatory)

Demo

Needs to learn Rego
Prefer the deployment with yml files rather than the OPA chart

Impact third party deployments (Charts)
Conflict with cluster admission controllers (cloud)
Silent blocking of low lever kubernetes objects

OPA security mecanims

- Authenfication & authorization on endpoints

- TLS encryption
- NetworkPolicy to inhibit incoming traffic

Main alternative : #Kyverno
Kube native
No rego just CRD

Features/Capabilities	Gatekeeper	Kyverno
Validation	✓	✓
Mutation	✓*	✓
Generation	X	✓
Image Verification	X (via extensions)	✓
Image Registry lookups	X (via extensions)	✓
Extensions	✓	X
Policy as native resources	✓	✓
Metrics exposed	✓	✓
OpenAPI validation schema (`kubectl explain`)	X**	✓
High Availability	✓	✓
API object lookup	✓	✓
CLI with test ability	✓***	✓
Policy audit ability	✓	✓
Self-service reports	X	✓

Link

Threat detection

AuditD -> log every commands executed
Ossec / Wazuh -> Activities monitoring
Falco -> Activities monitoring in containers (container runtime analysis)

Falco:

Input : Sys input / events
Have : rules / macros

Listen to syscalls from containers either with kernel module (only if you own the machine) or e bpf probe

Can receive k8s audit / custom sources

Install falco on the host or deploy falco on the cluster as a deamonset

Rules : define a behavior for which an alert should be trigerred
Macros : Reusable condition
Lists : set of reusable conditions

All can be defined on container / app level
A lot of rules already exists

Outputs : Broadcasting alerts -> #fluentd / traces / falco sidekick / mail

#Falco + #argo : deploy a sensor with argo. Falco calls the sensor that will create a network policy to block any traffic in the namespace. Then patch the pod to add the label of the network policy.

Advanced usecase with opsgenie to escalate steps by steps

Limitations :

- A lot of alerts are triggered with the default rules
- Error dropped events syscall
- Error if cluster > 400 nodes
- Follow up syscalls changes on kernel

Recommendations :

- Reduce the nb of elements to watch
- Start with a set of limited rules
- Structure pipeline to collect & store alerts from the beginning
- Reuse existing rules

Scanning image vuln

Tools :

- #Clair (already installed in #gitlab / #harbor / #quay)
- #Aqua Trivy : have a CLI binary

Recommendations :

- Update vuln db everyday
- Rely on multiple scanner
- Build and share reliable images
- Inspect third-parties images (dive)
- Sign your images and restrict deploys in kube (notary project)

Ressources :

- Kubernetes security (book)
- Kubernetes hardening guidance NSA
- CIS Kubernetes Benchmark
- learnK8S

The Practice of Securing Kubernetes

The Practice of Securing Kubernetes

#k8s #kubescape
Kubescape open source software created by the company of the speaker

Focused on security of the control plane

API auth

Client secrets :

- private key
- token auth
- static password
- OpenID Connect
  Rotation is an important protection factor

ABAC / RBAC Attribute/Role base access control
Webhook based decision delegation
Node authorization
AlwaysAllow

RBAC :

Role(ApiGroups/Resource/Verb) <–> RoleBinding <–> Subject (User/Group/ServiceAccount)

Use least privilege principle

Admission controller

RBAC is simple but limited
Limit what configuration a resource can have
Fine grained policies can be implemented by Admission controllers

Audit Loging

Can be integrated to prometheus
Webhook / cloud provider

Trust and secrets
API server TLS identity
Kubelet client TLS identity
ETCD SSL keys
Disable anonymous access

ETCD

Store all k8s secrets
Access control –> tls auth
API server can be configured to encrypt data

Kubelet authorization & authentication

Publish two listeners : port + read only port (hide/disable)
Anonymous auth (only for testing)
Cert based auth (mtls)
Token based auth (tls + token)

Node sensitive files

CA bundle file
Private key + cert for the node
Token webhook secrets
chmod 600

Network access

Kube node should not be accessible from the public internet
Including kubelet port(s)
The only place to enable is the incoming traffic (node ports)

Node bootstrapping

Bootstrap protocol

7 tools to help you secure your Kubernetes cluster

7 tools to help you secure your Kubernetes cluster

#vault #kubescore #kubiscan #falco #GCR #open-policy-agent #kubovisor

Frequent cause for security issues

- Sensible data
- Infra
- Cluster config
- Authorizations
- Workload content or config
- Lack of policies
- Failed components or applications
- No respect of best practices

Vault

Tool to manage secrets and protect sensitive data
Vault agent to inject secrets with annotations
What's wrong with kube secrets ?
can be encrypted at REST
but just encoded to base64

KuboScore (SAAS)

Automated solution to perform an audit of your kube cluster
Score & scenarios are free
Detailed reports aren't

KubiScan

Analyse risky RBAC permissions
Helps maintain the principle of Least privilege

Falco

Detects threats at runtime by observing the behavior of app & containers

GCR with vuln scanning

Container image registry with vuln scanning

Open Policy Agent

Allows to apply fine-grained policies on the cluster

KuboVisor

Easy to use tool to supervize your Kubernetes clusters
Subscription based

Fuzzing in go

Fuzzing in go

#fuzzing #golang

go test trace

check cpu work

go test fuzz

Fuzzing is a testing methodology based on random data
Random data based on a initial data set.
The fuzzer tries to mutate the data set until it reach some code coverage never reached before until a crash / error.

When to do fuzz

- For web servers / rest services
- Databases
- Codecs

Day 3

Keynotes

Welcome to 19th edition of Devoxx Belgium

Tickets selled too fast

Artificial Intelligence: You Are Here

A.I. is scary

Java *is* agile

Make jokes about js

JetBrains Fleet

Fleet

#jetbrains #fleet

What's fleet ?

Fleet is an editor
Central panel with code editor + 3 side panels

Editor mode

Code higlight but no code completion

Smart mode

Add language server
Add code processing (powered by intellij)
Can enable multiple language server at the same time

Remote dev

Can connect to another :

- fleet
- source code
- language server
- code processing

Remote collaboration

Same as remote dev with others connecting to the same instance(s)

Languages

Python / Java / JS / C# / PHP / TS / Go / Kotlin / Rust / JSON / HTML

Architecture

Frontend -> Workspace <- Code Engine
/
File system (File system deamon / code engine)

Demo

NewSQL, the holy grail of databases?

NewSQL, the holy grail of databases?

Focus on

- #VoltDB
- #NuoDB

VoltDB

Partitionning

- In memory storage
- Distributed serialized processing
  - If transactions affect multiple partitions they can be processed in parrarel
  - Timestamp ordering
  - No need for locks
  - Deadlock not possible

- Replicating tables
  - Best case few write multiple reads
- Sql compliant but not quite
- Does not implement forein key constraint

NuoDB

- Partitioning is range based
- Multi version concurrency control
  - Needs high precision clock sync of the nodes
  - Readers do not block writers
  - Writers do not block readers
  - But deadlocks are possible
- Allows unique secondary indexes

NewSQL can be

- Overkill
- Lack standards (diff features / limitations)
- Small community
- Lack of documentation

When to use

- Large amount of data
- High volume of data
- Low impedance

Is newsql the holy grail ?

No

What is the holy grail ?

- CAP Theorem
- New sql claims to be Consistent / Partitionned

A postgres-compatible DB on Kubernetes YugabyteDB

A postgres-compatible DB on Kubernetes: YugabyteDB

#k8s #postgres

Archictecture

Cluster with 4 nodes

- Local storage on nodes. The db will distribute the data
- 3 master (small pods) (admin clients)
- n server (persistent volume) (app clients)
- Statefull set in every availability zones

Failure case with replication factor 3

One node fails without sql processing

- New shards leaders will be elected to another node
  With sql processing
- Wait 3s to elec new leaders on another node
  When connected to a pod that died
- App will get an error and will retry to connect to another if you have a pool of connection

Feature

Fully compatible with postgres / open source

2022 A GitOps Odyssey

2022: A GitOps Odyssey

Started with a monolitc app
Deployed on bare metal / VM
Upgrade conflicts
Service unavailability

Moving it to the cloud

- Containerized
- Scalability
- Distributed caching
- S3 compliant storage
- Blue/Green upgrades

Product evolution

- Next generation
- cloud native
- microservices
- HA
- CD

Deploy strat

- private
- public
- hybrid

Challenges

- SaaS not always possible
- Unique customer demands
- Company hosted or on premise
- Cloud agnostic approach
- Vendor lock-in
- Embedded API gateway / IAM / db / redis **...**

Target environments

- Certified kube env
- Customer Hosted (AKS/EKS/PKS)
- AKS hosted with terraform

Introducing GitOps

- Everything is declarative (infra = code)
- Automation
- Auditable
  – Commit trail
  – Audit log
  – Transaction log
  "Stop scripting and start shipping"

FluxCD

Control loops

- Actual state <–> Desired State
- Observe –> diff –> act –> repeat
- Pull vs Push
  - Flux reconcile & adjust from the source code
- Controllers
  - Source controller
  - Helm controller
  - Kustomize controller
- Flux handle CRDs
- Define kube ressource for helm / kustomize / git ressources with values etc...

Build the cluster

Terraform

- Create infra cluster
- Bootstrap

Foundation

- General components (prom/graf/etc)
- Define once
- All clusters
- Networking
- Security

Platform

- Application based
- Define once
- Data platform (postgres..)
- Get bases add config combine them to make a "flavor" (customer specific deployment)
- Branch deployment ((release/flavor)/team/(env/tag))

Day 4

The State of OpenTelemetry for Java Developers

The State of OpenTelemetry for Java Developers

#open-telemetry

You can have agent on apps
and the collector that send data to the opentelemetry protocol and they will be obeserved by vendor specific exporters

Sprint boot 3 they wrapped open telemetry into micrometer.

The data collection is generic, the vendor isn't.

Let’s Go Triple Active with Three Clouds and Cilium

Let’s Go Triple Active with Three Clouds and Cilium

#k8s #Cilium

Why do we need multi-cloud connectivity ?

- Reduce cloud outage
- High-availability
- Resiliency
- Scalability
- Performance
- Security

Cloud agnotic tech
kube / cilium / nats / cockroachdb

requirements (focused on one region)

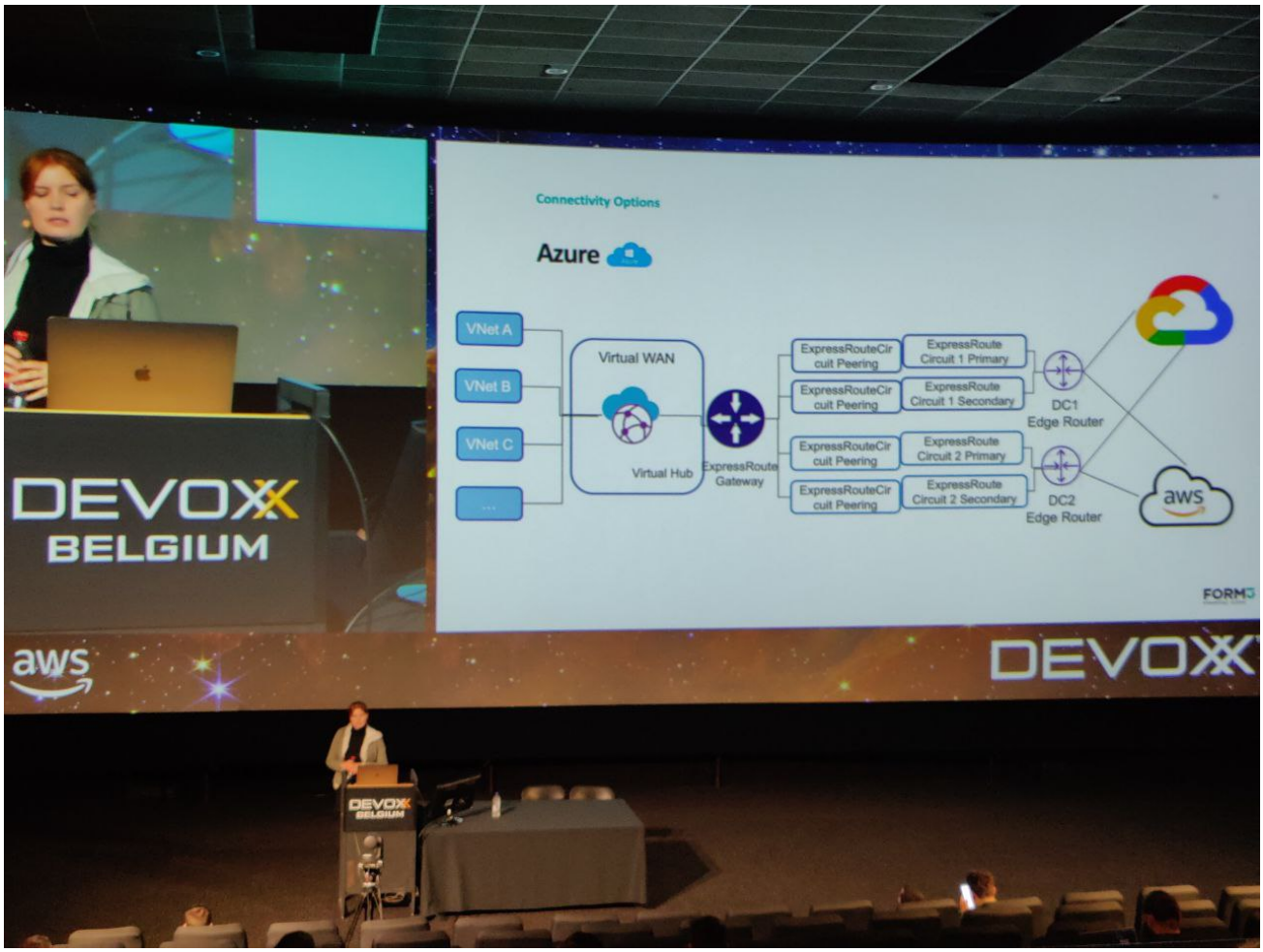
low latency 5ms
steady bandwidth
fault tolerance with active-active POP (Point of precense)

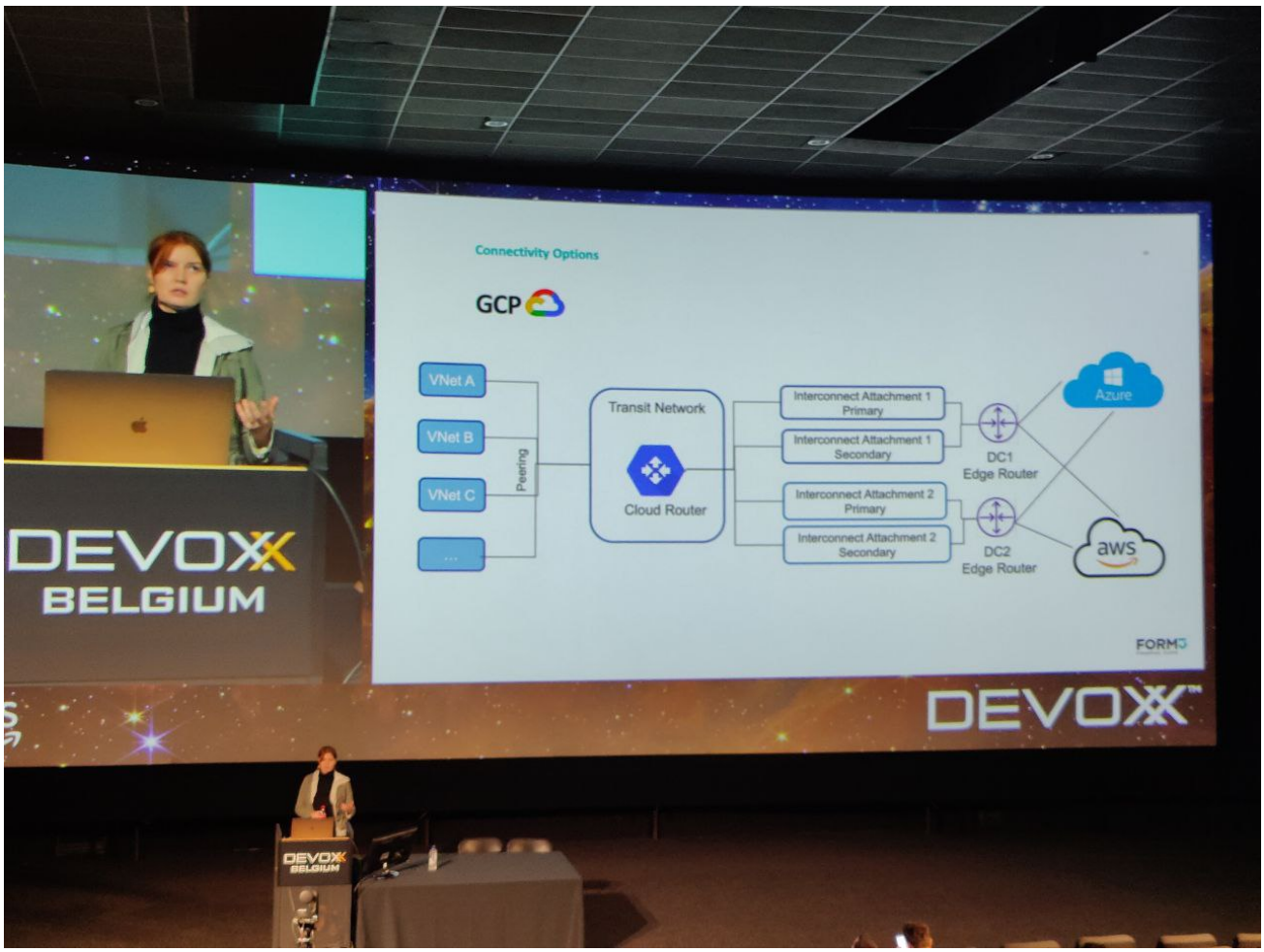
connectivy options
VPN / Direct Connectivity via cloud exchange provider / Direct Connectivity via data center

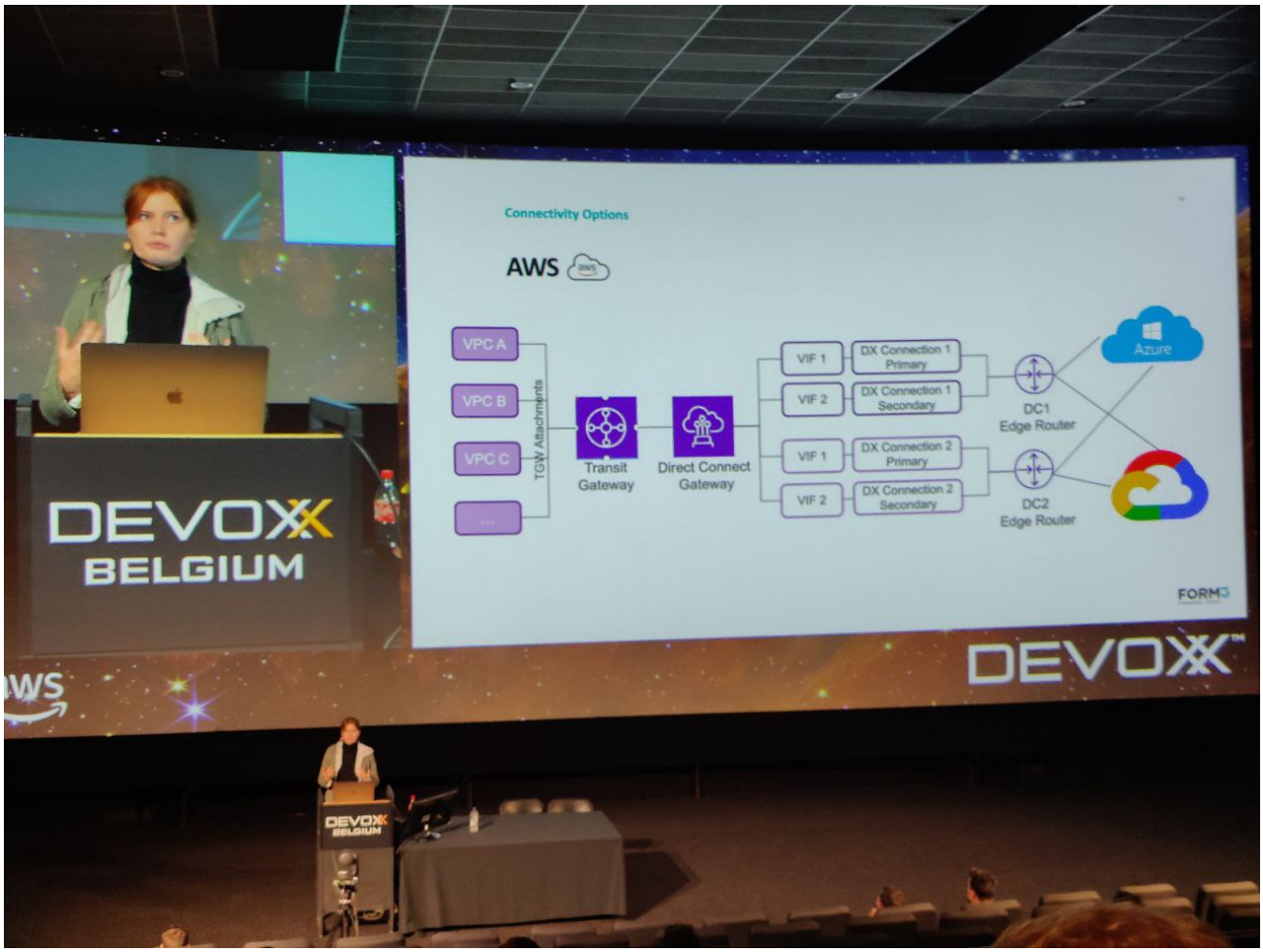
vpn is pretty instable bc it relies on internet (8/12 ms but can go as high as 100ms and unpredictable routing)

They tried DC via CEP but megaport ha second route was using francfort that added too much latency.

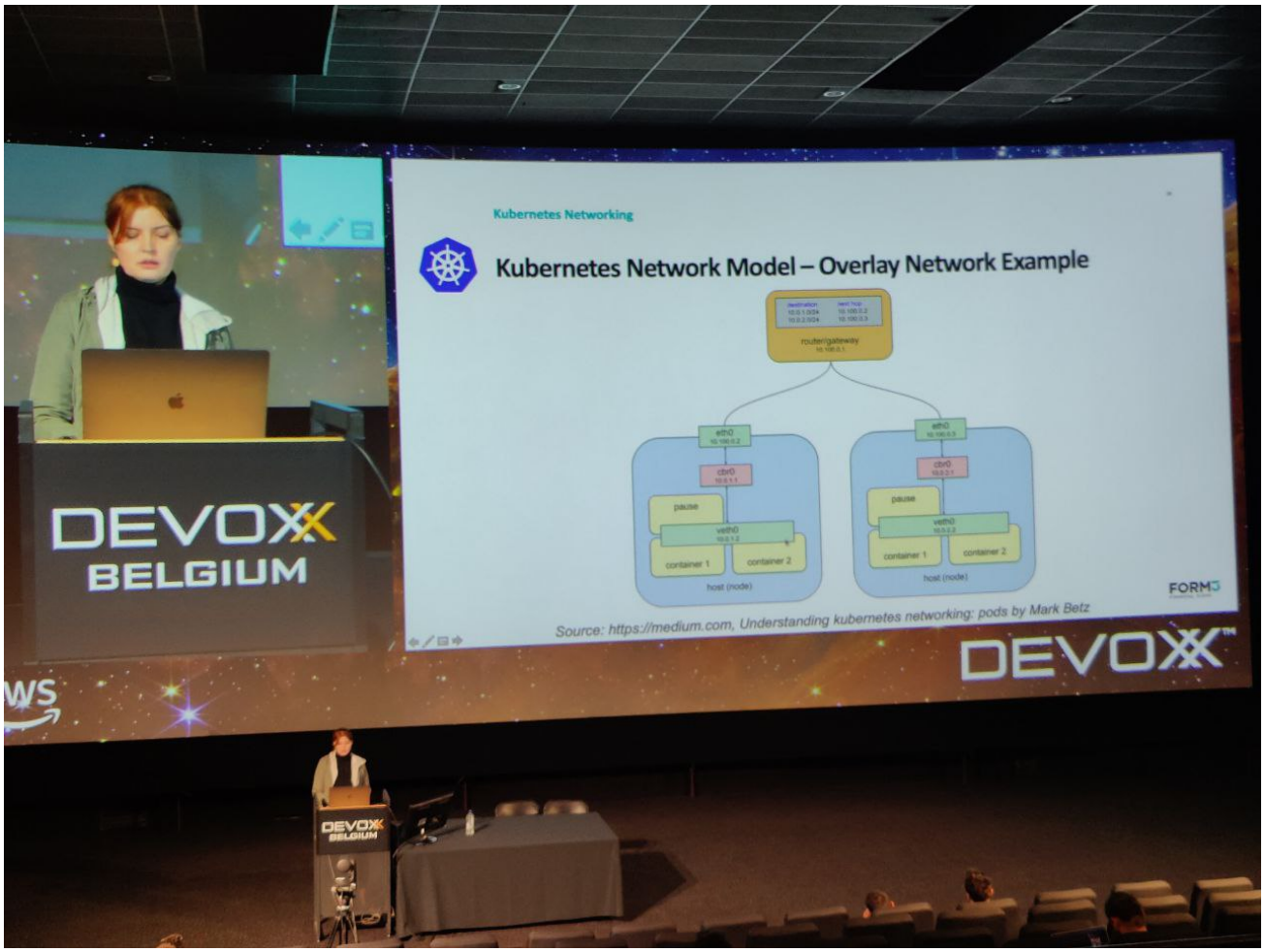
They used DC via DC.


Connectivity Options
Azure
VNet A
VNet B
VNet C
...
Virtual WAN
Virtual Hub
ExpressRoute Gateway
ExpressRouteCir cuit Peering
ExpressRouteCir cuit Peering
ExpressRouteCir cuit Peering
ExpressRouteCir cuit Peering
ExpressRoute Circuit 1 Primary
ExpressRoute Circuit 1 Secondary
ExpressRoute Circuit 2 Primary
ExpressRoute Circuit 2 Secondary
DC1 Edge Router
DC2 Edge Router
aws
FORM3
DEVOXX
BELGIUM


Connectivity Options
GCP
VNet A
VNet B
VNet C
...
Peering
Transit Network
Cloud Router
Interconnect Attachment 1 Primary
Interconnect Attachment 1 Secondary
Interconnect Attachment 2 Primary
Interconnect Attachment 2 Secondary
DC1 Edge Router
DC2 Edge Router
Azure
aws
FORM3
DEVOXX
BELGIUM



Kube network model



Cilium

Operate in L3/4 and L7
Use eBPF
Load balancing
Security
Network connectivity
Does network through the linux kernel instead of iptables

Cilium multi cluster = cilium cluster mesh

Features

- Service discovery & load balancing
- Network policies
  - Identity based
  - Multi cluster
  - L3/L4, L7
- Encryption
  - IPSec & Wireguard
- Pod IP routing
  - Tunneling & direct routing

Cluster mesh config
Each cluster domain name must be resolvable to an ip address

AWS load balancing are DNS based : to fix that they patched cilium agent added to local /etc/hosts

Lessons Learned

cilium-cli doesn't work on aws bc of hostname based load balancers
Open Source docs better than enterprise version
Need to learn how to clean properly existing CNI

- Need to taint until cilium is ready
  API Server
  Loadbalancer
  DNS-Based for aws
  IP based for Azure/GCP

Static IP addresses and host aliases

GCP/Azure : simple
AWS : more complex and may have collisions (dedicated subnets / reserved pools)

External DNS -> cloud dns -> multi-cloud-dns

Does Cilium deliver on it's promises ?

Similar perf than standard cloud CNI
eBPF doesn't work on cloud bc the cloud provider is using their own network
Encryption isn't done on a single node but between nodes.

Load Testing Crash Course with Gatling

Load Testing Crash Course with Gatling

#gatling #load-test

Why you must do load-testing ?

QA conditions =! prod conditions
Large audience
Seasonal peaks
Fast growth
Application sprawl
Product launch

Poor UX (crash/slowdowns) = Losses (sales/brand damaged)

Operating costs = Expensive
Scaling out = Complex
The cloud won't take care of perf.

What is load-testing ?

Trying to simulate artificial traffic.
Validate application meets perf requirements

- Analyse response times
- Discover bottlenecks
- Size your infrastructure

What load-testing is not ?

- Webperf (pagespeed)
- Running headless browers at scale

Build realistic performance test

- Stateless
  - Traffic is anonymous
  - Can be executed in any order
  - Can replay recorded traffic
- Statefull
  - Logicial flows
  - Implement scenarios
  Focus on most critical features
  Iterate and improve

Type of traffic

Simulate public facing traffic -> gatling default behavior
Testing microservices -> Must use shareConnections()

Type of model

Open system model : new jobs arrive independently of jobs completion.

- Users keep on arriving

- Concurrent users is an output
  Closed system model : new jobs only arrive after jobs completion.
- Capped concurrent users
- Injection slows down to limit

Type of test

Capacity test

How your app scales

Stress test

How your app behaves under sudden load peak

Soak test

If your app degrades over time

Smoke test

Making sure that your test work correctly

Test data

Don't test your caches

- No static values
- Inject test data into virtual users

How to not measure

Issue with average on response time

- tong tail and outliers
- multiple modes
  Don't use standard deviation
  Good metrics : percentiles (50% median value)
  You cannot average percentiles
  Don't assume distributions, measure them properly.
  Percentiles can define acceptance criterias.

Dev lifecycle

Old way

- 1–2 times a year
- After dev and func tests
- 2 weeks before go-live

Today

- Integration in CI/CD

By who

Old way

- experts
- central team
  Create bottleneck / no collaboration

Today

- Collaboration: dev, ops, QA
- Dev for crafting & maintaining

Tooling

Provisioning
Monitoring
Load generator

Gatling

Open source (core)
Code oriented
Support a lot of protocols

The Hacker’s Guide to Kubernetes Security

The Hacker’s Guide to Kubernetes Security

OWASP Top 10

Protecting the world’s greatest open source ecosystem with Sigstore

Protecting the world’s greatest open source ecosystem with Sigstore

#sigstore

Why no one check the integrity of signed artifacts ?

Because dev are lazy & don't want the toil associated to managing keys.

TUF – The update framework

Provides a framework for distributing updates that is safe from a variety of attacks :

- Vuln to key compromises
- Malicious mirrors
- Roll back attacks
- Fast forward attacks

Sigstore keyless signing

Fulcio

- Embed your verified identity
- Short lived signing certificates

- Discard private keys after use
- Auditable Certificate Transparency logs

Who signed this ?

- OIDC tokens are issued by various identity authorities for people or machines

Rekor

Transparency log

Clients

go / js / python / java

Distributed applications – Better off with frameworks, Kubernetes, service meshes or all

Distributed applications – Better off with frameworks, Kubernetes, service meshes or all of them at once?

#k8s #serviceMesh #ebpf

Why distributed architecture ?

- Run multiple differents tasks that use multiple technologies
- Redundancy
- Load-balancing
- Resilience
- Security
- Dynamic updates (blue/green)
- Routing / traffic shapping
- Service registration
- Service discovery
- Distributed config / logging / tracing

Day 5

Debugging distributed systems

Characterictis of distributed systems

- concurrency of components
- lack of a global clock
- Independent failure of components

Fallacies of distributed computing

- Network is reliable
- Latency is zero
- Bandwith is infinite
- Network is secure

- Topology doesn't change
- There is one admin
- Transport cost is zero
- The network is homogeneous

Structured approach

Observe document

- Inspect log / errors / metrics / tracing
- Draw path from source to target.
- Document what you know
- Can we reproduce in a test ?
  - By injecting errors
- Tools
  - Whiteboard
  - Documentation
  - Tracing / logging / metrics
  - Tests

Mininal reproducer

- Maximise the amount of debugging cycles
- Focus on short development iterations / feedback loops
- Get close to the action
- Tools
  - IDE / Shell script / SSH tunel / curl

Debug client side

- Focus on eleminating anything that could be wrong on the client side
- Are we connecting to the right host ?
- De we send the right message ?
- Do we receive a response ?
- Not much different from local debugging
- Tools
  - IDE, Debugger, logging

Check DNS and routing

- DNS
  - make sure you know what IP address the hostname should be resolved to
  - Verify that the client have the same result
- Routing
  - Verify you can reach the target machine
- Tools
  - host
  - nslookup
  - dig
  - etc...

Check connection

- Can we connect to the port
- Do we get a reject or a drop ?
- Does the connection open and stay open ?
- Are we talking TLS ?
- What is the connection speed between us ?
- Tools
  - ifperf
  - telnet
  - nc
  - curl

Inspect traffic / messages

- Do we send the right request ?
- Do we receive the right response ?
- How do we know ?
- How do we handle TLS ?
- Any load balancers or proxies in between ?
- Tools
  - tcpdump
  - curl
  - wireshark
  - network tab in browser
  - tls proxy

Debug server side

- Inspect the remote host
- Can we attach a remote debugger ?
- Profiling
- Java flight recorder
- Strace
- Tools
  - ssh tunnel
  - debugger
  - strace

Wrap up & post mortem

- Document the issue
  - Timeline
  - What did we see ?
  - Why did it happen ?
  - What was the impact ?
  - How did we find out ?
  - What did we do to mitigate and fix ?
  - What should we do to prevent repetition ?
- Tools
  - Whiteboard / documentation

Summary

- Observe and document
- Create minimal reproducer
- Debug client side
- Check dns
- Check connection
- Inpect traffic
- Debug server side
- Wrap up

Misc

Investigate

- Kiali – The Console for Istio Service Mesh
- Zipkins – distributed tracing system
- JIB – google tool to generate docker containers without docker files for java apps
- Wildcard dns is not safe ???

Tools

K8S

- KDash
  - Helm
    - #Dekorate
    - #Jkube
- KubeScape
- gVisor
- Vault
- Velero
- VCluster
- Istio
- Flux
- Argo