Kubernetes + Sikkerhet = Sant?

Marius Sandbu
Skyevangelist @ Sopra Steria

sopra steria

Hva er de største utfordringene med å sikre Kubernetes?

Per september 2021, noen av de største utfordringene med å sikre Kubernetes inkluderte:

1. Kompleksitet: Kubernetes-økosystemet er omfattende og komplekst, og det kan være utfordrende å forstå alle aspekter av sikkerheten. Det er mange konfigurasjonsmuligheter og integrasjoner som krever grundig forståelse for å unngå sikkerhetsfeil.
2. Misbruk av privilegier: Kubernetes bruker RBAC (Role-Based Access Control) for å administrere tilgangen til ressurser, men feilaktig konfigurering kan føre til overflødige privilegier. Hvis en angriper får tilgang til for mange ressurser, kan dette føre til at de får kontroll over hele klyngen.
3. Sårbarheter i containere: Containersikkerhet er avgjørende, da sårbarheter i containere kan utnyttes for å få uautorisert tilgang til podder og klynger. Det er viktig å kontinuerlig overvåke, skanne og patche containerbilder for å unngå eksponering av kjente sårbarheter.
4. Orkestrert angrep: Angripere kan målrette hele Kubernetes-klynger i stedet for individuelle ressurser. Dette kan gjøres ved å utnytte sårbarheter i Kubernetes-kjernekomponentene eller utnytte konfigurasjonsfeil i klyngen.
5. Manglende isolasjon: Hvis podder og tjenester ikke er tilstrekkelig isolert, kan en kompromittert pod gi angripere muligheten til å bevege seg sidelengs i klyngen og få tilgang til andre ressurser.

Hva er de største utfordringene med å sikre Kubernetes? skriv et kort svar

De største utfordringene med å sikre Kubernetes inkluderer kompleksitet, misbruk av privilegier sårbarheter i containere, orkestrerte angrep, manglende isolasjon, manglende logging og overvåking tredjepartsavhengigheter og ubevoktede API-eksponeringer.

Må også omfatte «arbeidsbenken»

Et VELDIG dynamisk økosystem

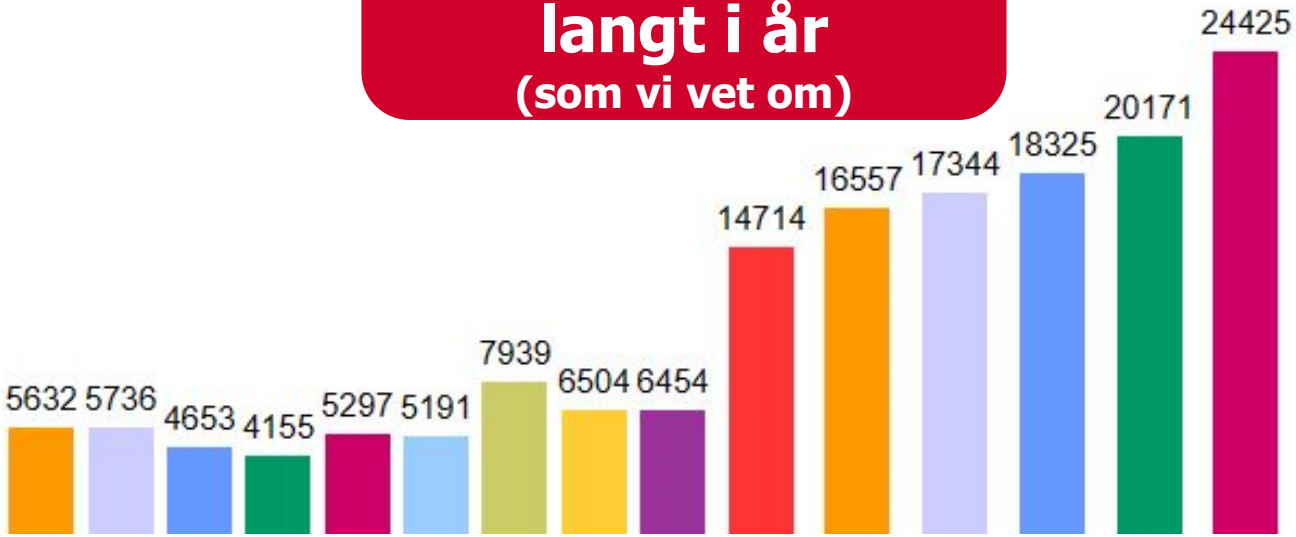
Masse muligheter men kan fort bli komplekst

NSM Risiko rapport 2023

Utnyttelse av cybersårbarheter lar ikke vente på seg

I Norge har vi fra 2019 til 2021 sett en tredobling i alvorlige cyberoperasjoner mot norske myndig-heter og virksomheter. Antallet alvorlige og svært

18500 sårbarheter så langt i år
(som vi vet om)



Antall offentlige kjente sårbarheter fra 2008 – 2022

sopra steria

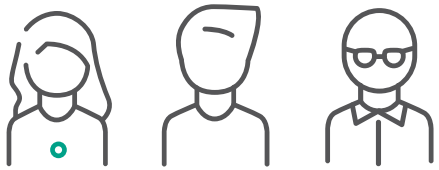
Angrepsvinkler

Platform team

- Sårbarheter
  - Sårbarheter i eksterne løsninger og applikasjoner
- Feilkonfigurerte tjenester
  - Tjenester som mangler sikkerhetsmekanismer
- DDoS-/DoS-angrep
  - TCP SYN Flood / HTTP GET/POST Flood

Samhandling

Fysisk tilgang | E-post | Web



Utviklere

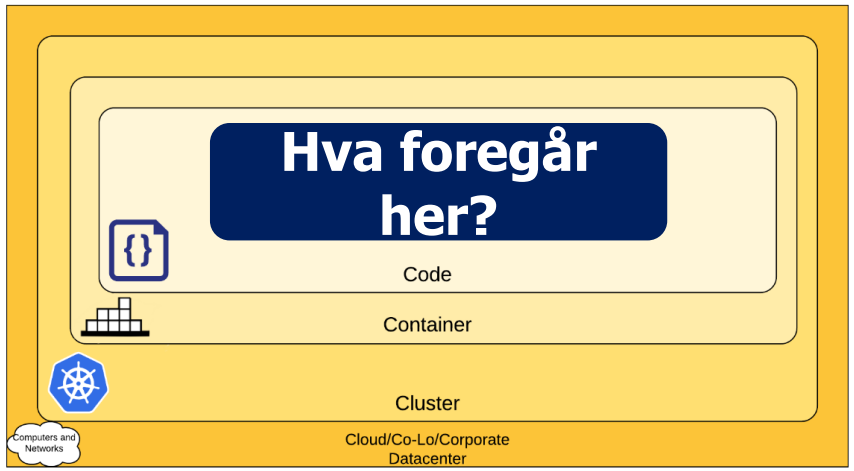
- Brukerinformasjon på avveie
  - Credential Stuffing – MFA Fatigue - Tokens
- Phishing
  - Qbot / Icedid
- Drive-by download
  - RedLineStealer

sopra steria

Risikoer i et Cloud-native miljø?

- Uauthorisert tilgang til Kubernetes API
- Uauthorisert tilgang til CI/CD og kildekode
- Sårbarheter i applikasjon eller container image
- Mangel av segmentering mellom tjenester
- Nøkler og passord lagret i klartekst i container

Container applikasjoner

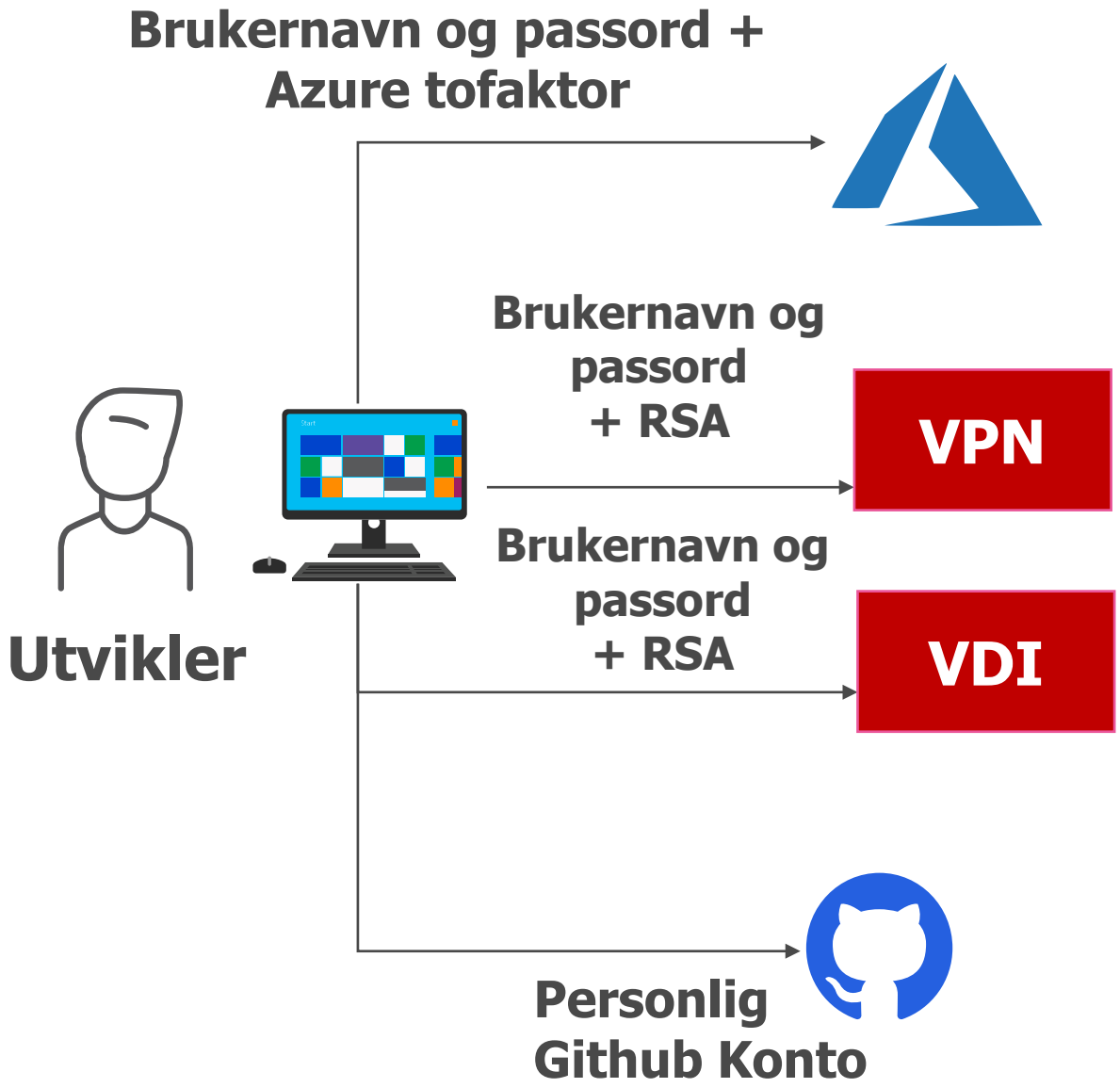


The 4C's of Cloud Native Security

- Sårbarheter i Kubernetes
- Container Escape
- Supply-chain svakheter
- Mangel av trafikkinnsikt mot tjenester
- Mangelfull tilgangskontroll

[Tactics - Threat Matrix for Kubernetes https://microsoft.github.io/Threat-Matrix-for-Kubernetes/]

Ikke en brukeropplevelse man vil ha



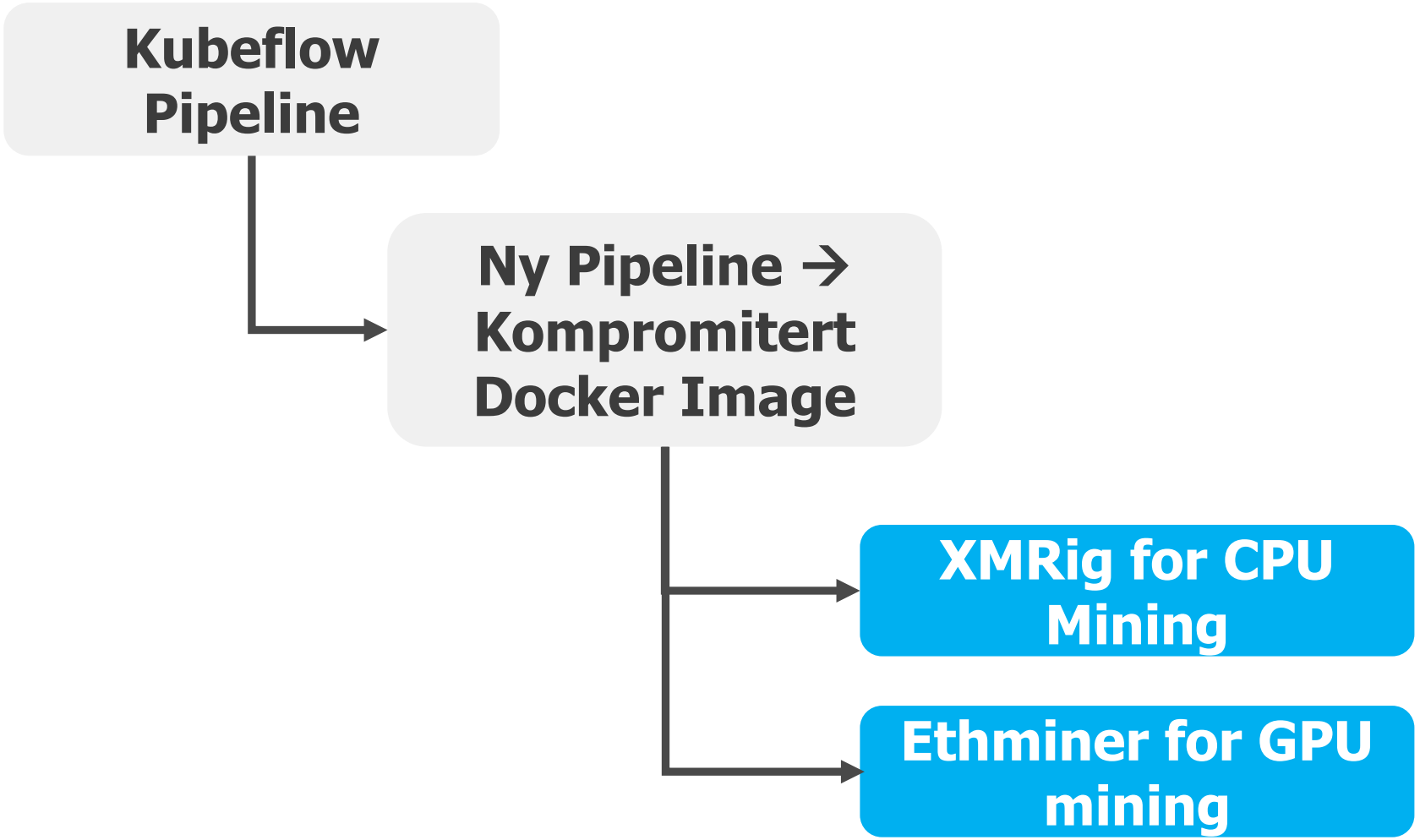
- Tilgang til interne filtjenester via VDI
- Ingen sikkerhetsjekk av endepunkt
- Flere agenter og flere to faktor løsninger for bruker
- Ingen to faktor mot Github samt ingen automatisering av bruker

Ikke alle angrep som er like farlig

Kubernetes med Kubeflow

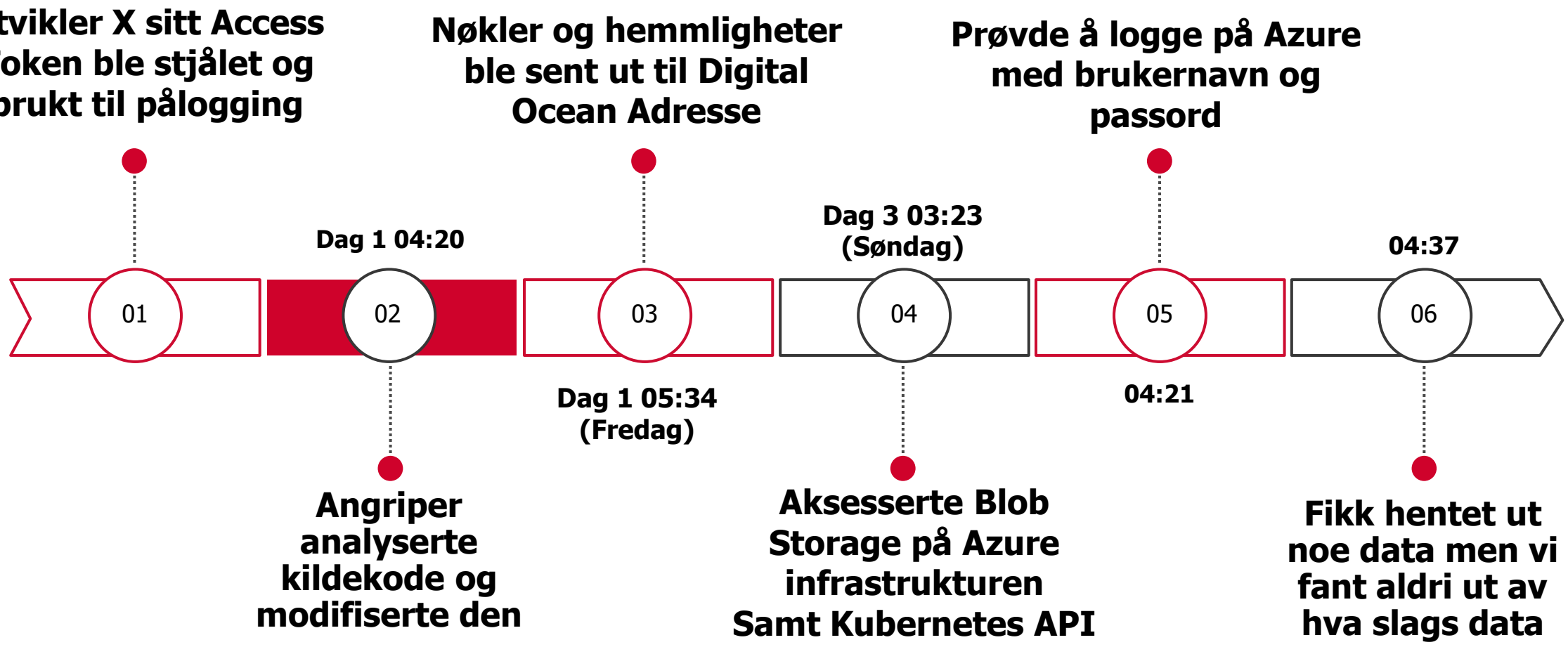
Organisasjon brukte Kubeflow for Maskinlæring jobber

Kubeflow dashboard åpnet «som standard» fra Internett



sopra steria

Komprimert Git Pipeline

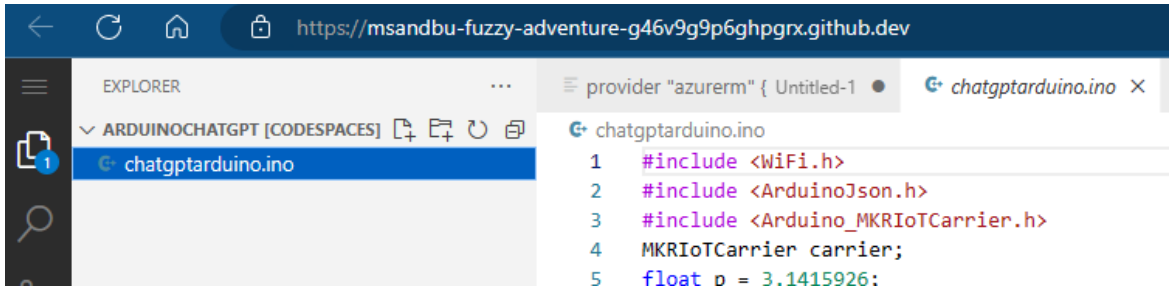


Hva slags tiltak kan vi gjøre for å øke sikkerheten?

Tiltakene og funksjonalitet er avhengig av **organisasjon / plassering** (Cloud) og ikke minst **behov / krav** til sikring samt synlighet

Sikring av verktøyene

- **VS Code**
- **GitPod**
- **Github Codespaces**
  - Leverer en dynamisk IDE via nettleser
  - Kan også benyttes mot sentral iDP
- **Virtual Desktop**
  - For utvikling som ikke er supportert i web IDE
  - Eksempel: Hardware utvikling
  - Leverer en sikker arbeidsflate med nedlåst OS

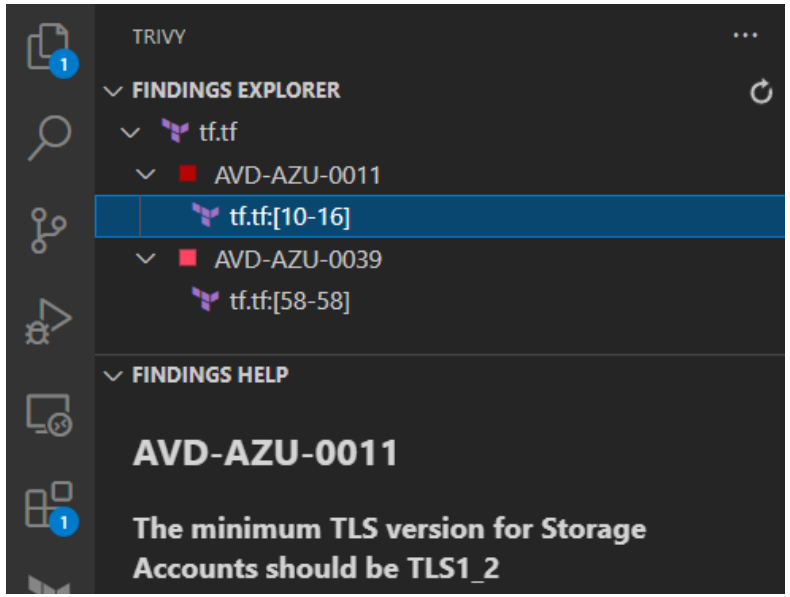


Vær forsiktig med Copilot

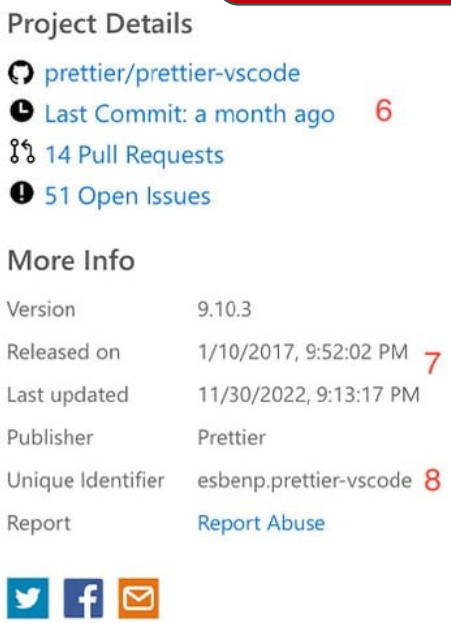
```
os_profile {
  computer_name  = "example-vm"
  admin_username = "adminuser"
  admin_password = "Password1234!"
}

os_profile_linux_config {
  disable_password_authentication = false
}
```

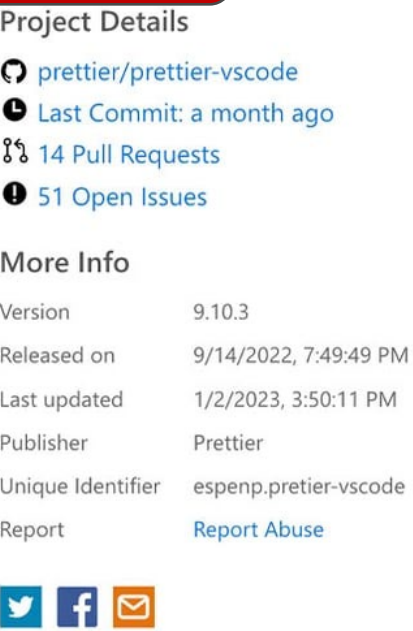
Hvordan sikre arbeidsbenken?

- **Vær forsiktig med bruk av extensios**
  - Dobbeltsjekk hvem som har publisert
  - Dobbeltsjekk når den ble publisert
- **Utvidelser for code scanning**
  - Checkov
  - Trivy
  - Tfsec
  - Snyk



Kan du se hvem som er fake?



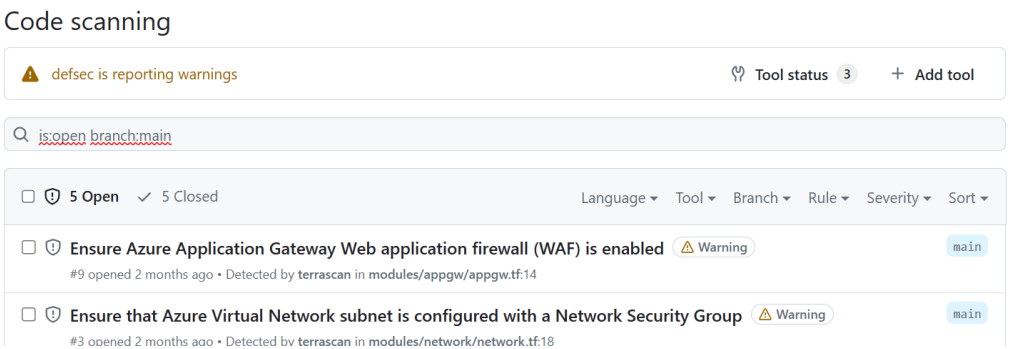


sopra steria

Sikkerhetsmekanismer for GitHub

- **SCIM** – Brukerprovisjonering fra identitetskatalog
- **SSO** og tilgangskontroll med **SAML/OAuth**
- **Self-hosted runners** og private repositories

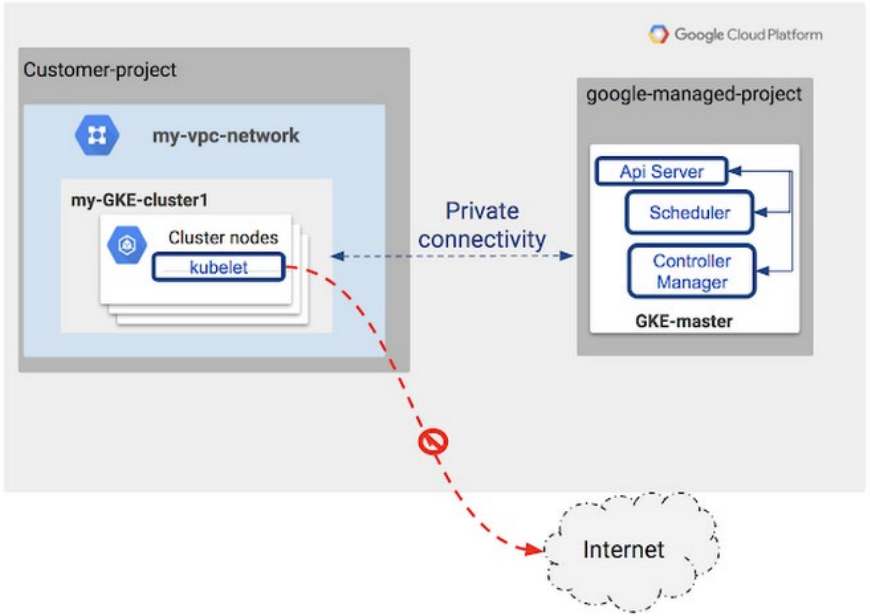
- **TFSec** = Inspisere sikkerhet i Terraform kode
- **Trivy** = Inspisere sikkerhet i TF Code og Container images
- **Git Signed Commit**
- **Github Advanced Security**
  - **Secret scanning =** Gratis tjeneste
- **Administrere API tilgang**
  - fine-grained personal access tokens
  - Kan settes som standard på organisasjonsnivå



Github Advanced Security: (Funksjonalitet for private repos)
Code scanning
Secret Scanning
Depedency Review

Private Cluster

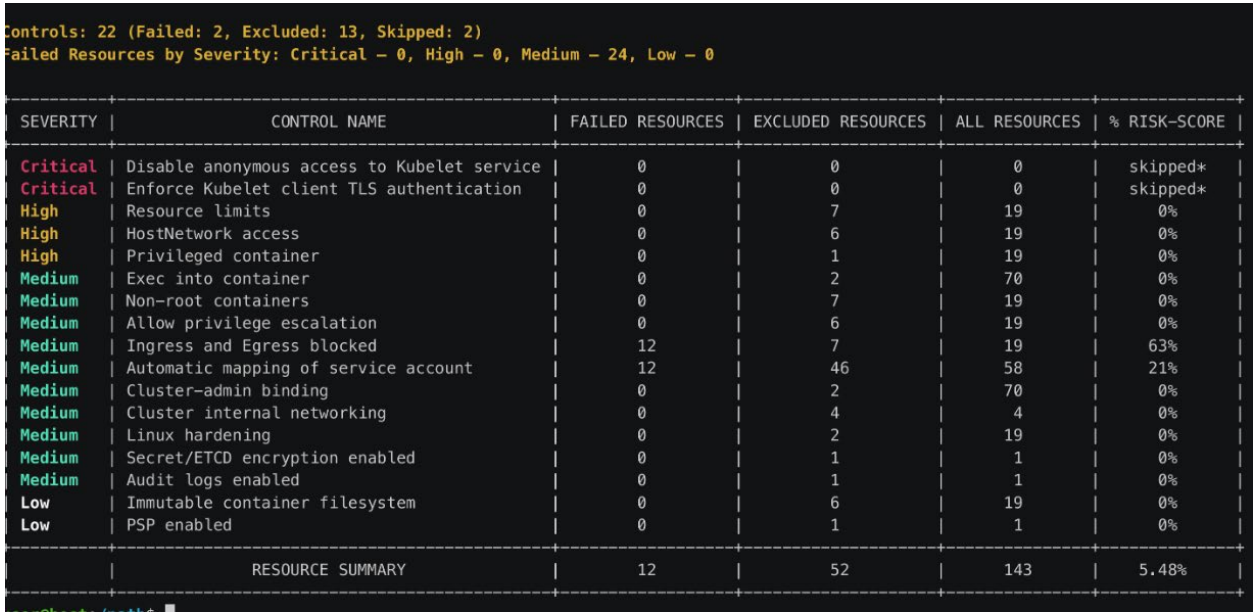
- **Ingen begrensning** på antall forespørseler på autentisering
- Mest en utfordring for Kubernetes på **offentlig sky**
- **Redusere misbruk** av f.eks tokens eller credentials mot KubeAPI
- Noen sårbarheter her i de siste årene (eks: **CVE-2022-3172**)
- Eller i det minste låse ned tilganger fra godkjente adresser



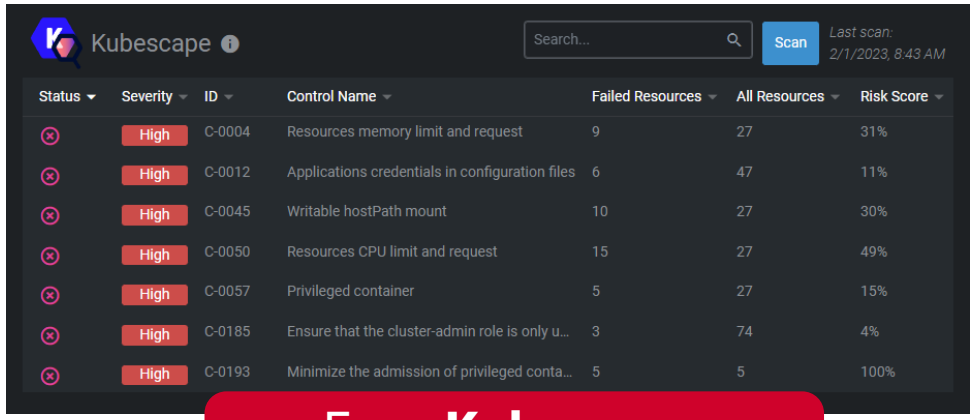
Krever ofte **at cluster blir satt opp på nytt** om det skal tas i bruk på offentlig sky

Er det i henhold til beste praksis?

- **Kubebench –** Scanner miljøet i henhold til CIS
- **Kubescape –** Scanner miljøet i henhold til NSA-CISA, CIS
  - Begge forstår lokasjon (ergo skyleverandørene)
  - Kubescape kan også kjøres via Github Actions eller CLI
  - YAML og JSON



```
Controls: 22 (Failed: 2, Excluded: 13, Skipped: 2)
Failed Resources by Severity: Critical – 0, High – 0, Medium – 24, Low – 0

+----------+-----------------------------------------+------------------+------------------+---------------+--------------+
| SEVERITY |                CONTROL NAME             | FAILED RESOURCES | EXCLUDED RESOURCES | ALL RESOURCES | % RISK-SCORE |
+----------+-----------------------------------------+------------------+------------------+---------------+--------------+
| Critical | Disable anonymous access to Kubelet service |        0         |        0         |       0       |   skipped*   |
| Critical | Enforce Kubelet client TLS authentication |        0         |        0         |       0       |   skipped*   |
| High     | Resource limits                         |        0         |        7         |      19       |      0%      |
| High     | HostNetwork access                      |        0         |        6         |      19       |      0%      |
| High     | Privileged container                    |        0         |        1         |      19       |      0%      |
| Medium   | Exec into container                     |        0         |        2         |      70       |      0%      |
| Medium   | Non-root containers                     |        0         |        7         |      19       |      0%      |
| Medium   | Allow privilege escalation              |        0         |        6         |      19       |      0%      |
| Medium   | Ingress and Egress blocked              |        12        |        7         |      19       |      63%     |
| Medium   | Automatic mapping of service account    |        12        |        46        |      58       |      21%     |
| Medium   | Cluster-admin binding                   |        0         |        2         |      70       |      0%      |
| Medium   | Cluster internal networking             |        0         |        4         |       4       |      0%      |
| Medium   | Linux hardening                         |        0         |        2         |      19       |      0%      |
| Medium   | Secret/ETCD encryption enabled          |        0         |        1         |       1       |      0%      |
| Medium   | Audit logs enabled                      |        0         |        1         |       1       |      0%      |
| Low      | Immutable container filesystem          |        0         |        6         |      19       |      0%      |
| Low      | PSP enabled                             |        0         |        1         |       1       |      0%      |
+----------+-----------------------------------------+------------------+------------------+---------------+--------------+
|          |              RESOURCE SUMMARY          |        12        |        52        |      143      |    5.48%     |
+----------+-----------------------------------------+------------------+------------------+---------------+--------------+
```



Egen **Kubescape** utvidelse for LENS IDE

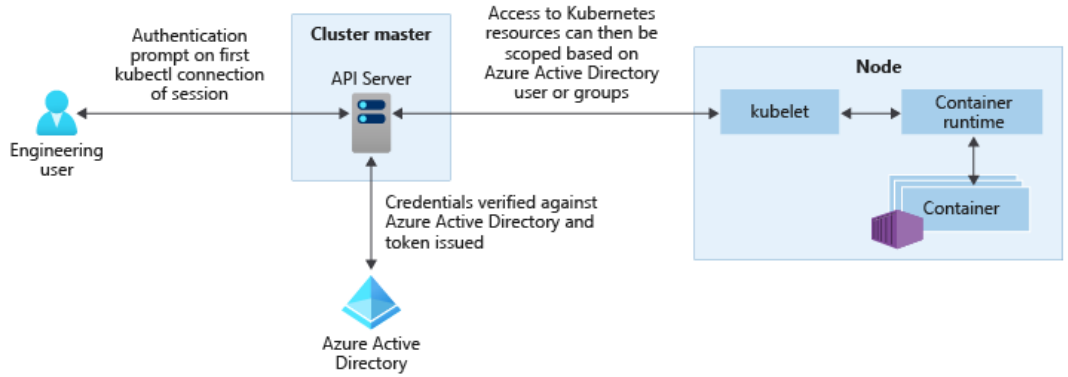
IAM mekanismer og RBAC

- Autentisering mot KubeAPI enten via
  - **Token**, **Sertifikat** eller **Autenseringsproxy**
  - Sertifikat i Kubernetes kan ikke bli revoked
  - Ingen standard LDAP integrasjon
    Eks: **Azure AD, Google** eller **OpenID Connect**
  - **RBAC er for å legge til tilganger (ingen deny)**
  - **Roller kan settes på enten namespace eller Cluster**

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
namespace: default
name: pod-reader
rules: - apiGroups: [""] # ""
resources: ["pods"]
 verbs: ["get", "watch", "list"]
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: read-pods
 namespace: default
subjects:
 - kind: User
name: minikube
apiGroup: rbac.authorization.k8s.io
roleRef: kind:
Role name: pod-reader
apiGroup: rbac.authorization.k8s.io
```

dexidp/dex eller pinniped



RBAC og API Objekter
Alle tilganger kan delegeres (CRUD)

Role | **Rolebinding** | **Namespace**

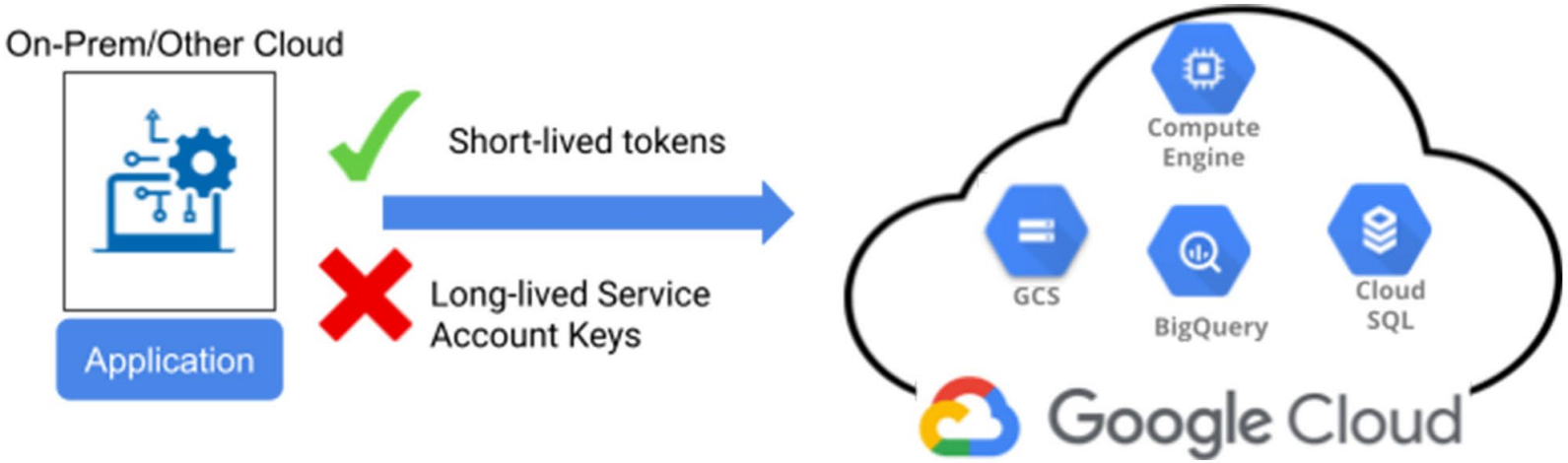
ClusterRole | **ClusterRoleBinding**

sighupio/permission-manager

sopra steria

Workload Identity

- Sikker autentisering fra **container** til **PaaS tjenester**
- Foreløbig støttet av Google og Microsoft
- Funker mot Kubernetes cluster som kjører «hvor som helst»
- Federert autentisering via OpenID Connect
- Unngå bruk av nøkler i containere for autentisering

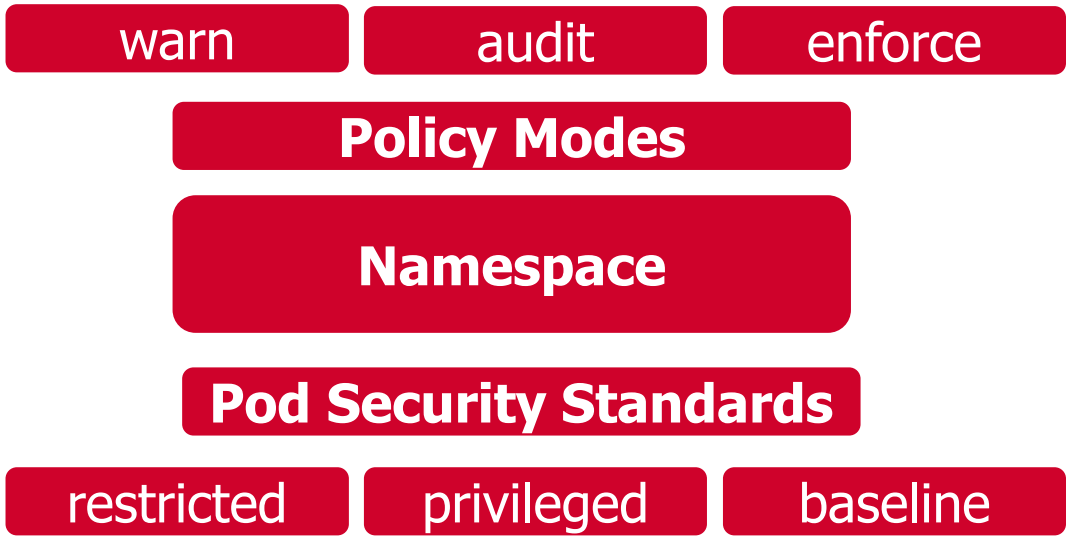


Pod Security Admission (PSA)

- **For å ha bedre kontroll på tjenester/pods**
- **Kan definere ulike sikkerhetsnivåer**
- **3 innebygget nivåer**
  - Eks: Privileged gir ingen begresninger
  - Start med restricted som standard

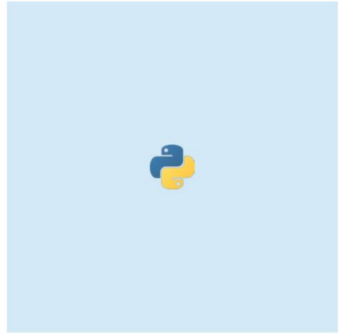
kubectl label --overwrite ns test-privileged pod-security.kubernetes.io/**enforce=privileged** pod-security.kubernetes.io/**warn=privileged**

kubectl label --overwrite ns test-restricted pod-security.kubernetes.io/**enforce=restricted** pod-security.kubernetes.io/**warn=restricted**



Sikring av Container Registry

- **Mange bruker images med stort avtrykk**
  - Ofte krever at man endrer hvilket image man bruker (eks: Python)
- Ofte består applikasjoner av egen kode samt open-source kode
- **Private vs Public Image Registry**
- **Låse ned tilganger fra CI/CD byggprosess**
- **Kun tillatte «godkjente» image**
- Tilgangskontroll og autentsering med sentral identitet
- Image Scanning funksjonalitet for å avdekke sårbarheter
  - De fleste leverandører har innebygget funksjonalitet
  - Anbefaler heller at man gjør scanning før image blir pushet til repository



python

882 MB
431 dependencies
268 vulnerabilities
66 high severity

python: 3-slim-buster

113 MB
94 dependencies
75 vulnerabilities
1 high severity

Note: none of the high severity vulnerabilities currently have fixes available, nor do they have an exploit in the wild.

sopra steria

Oppgradering og patching

- **1 år aktiv support på en minor release**
- Vedlikehold i 14 måneder
- Noe ulik standard på hva som er supportert
  - Forskjelling fra leverandør til leverandør
- Kubernetes patcher kommer ofte ukentlig
- Worker og master noder OS patcher
  - **Mange oppdateringer krever restart**
    - Litt avhengig av underliggende OS
    - Eks: VMware - Photon, Microsoft – Mariner
  - **Kured (Kubernetes Reboot Daemon)**
    - Ser etter /var/run/reboot-required og rebooter om nødvendig
  - **Enten statisk vedlikehold eller bygge noe maskiner**

K8s version	Upstream release	AKS preview	AKS GA	End of life
1.24	Apr-22-22	May 2022	Jul 2022	Jul 2023
1.25	Aug 2022	Oct 2022	Dec 2022	Dec 2023
1.26	Dec 2022	Feb 2023	Apr 2023	Mar 2024
1.27	Apr 2023	Jun 2023	Jul 2023	Jul 2024

Få varsel om usupporterte APIer

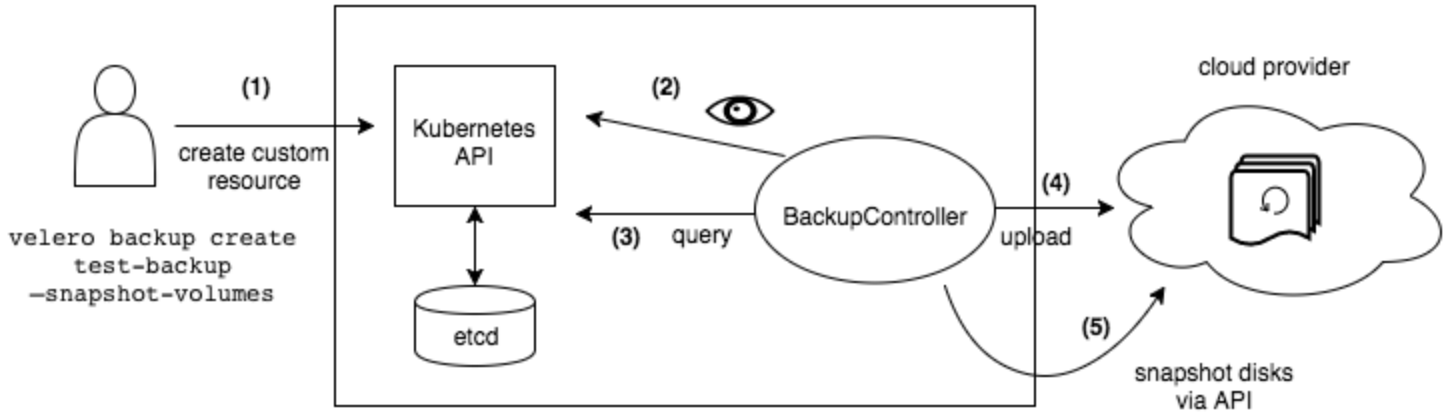
FairwindsOps/pluto

sopra steria

Backup og data beskyttelse

- **For tjenester som krever persistente data**
  - Provisjoneres ofte via innebygget CSI (storage interface)
    - Cloud leverandørene, Dell, HP, NetApp, IBM osv.
  - F.eks lagring til database tjenester eller annet som krever read-write.
  - Backup er noe som må settes opp utenom
    - Bruk av verktøy som **Velero**, **Kasten** eller **Portvorx Backup**
    - Noen lagringsleverandører har innebygget backup

Kubestr kan brukes til å validere CSI drivere samt kjøre ytelsestesting



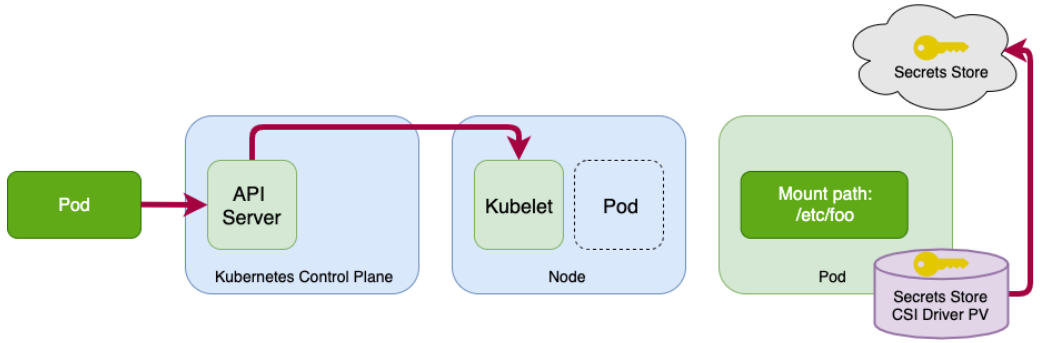
Secret håndtering

Caution:
Kubernetes Secrets are, by default, stored unencrypted in the API server's underlying data store (etcd). Anyone with API access can retrieve or modify a Secret, and so can anyone with access to etcd.
Additionally, anyone who is authorized to create a Pod in a namespace can use that access to read any Secret in that namespace; this includes indirect access such as the ability to create a Deployment.

- **Etcd har ingen innebygget versjonering**
- **Data er ikke kryptert At-Rest**
- Anbefaling er enten Secret Store CSI Driver eller Kubernetes External Secret Operator
- Ekstern Secret Provider flytter secrets ute av Kubernetes miljøet

Features \ Providers	Azure	GCP	AWS	Vault
Sync as Kubernetes secret	Yes	Yes	Yes	Yes
Rotation	Yes	Yes	Yes	Yes
Windows	Yes	No	No	No
Helm Chart	Yes	No	No	Yes

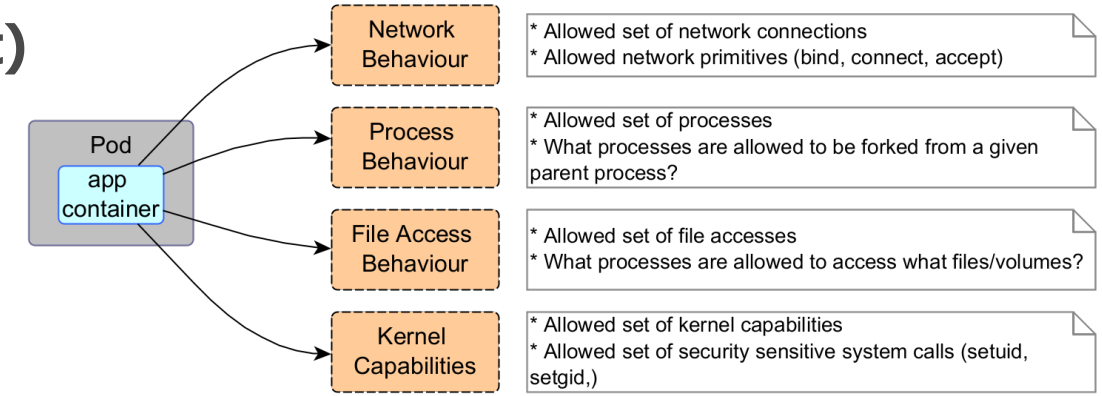
Supportert for Secret Store CSI



sopra steria

Hva skjer inni tjenesten?

- **Falco + Falcosidekick (Audit + Synlighet)**
- **Tracee (Audit + Synlighet)**
- **Tetragon (Audit + Synlighet)**
- **KubeArmor (Audit + Beskyttelse)**
  - Alle benytter eBFP
  - KubeArmor Kan f.eks automatisk generere network policies
  - Basert på trafikk synlighet (avhengig av riktig CNI)




```
process default/test-pod /usr/local/bin/curl -L github.com
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
dns     default/test-pod /usr/local/bin/curl [github.com.] => [140.82.121.4]
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
connect default/test-pod /usr/local/bin/curl TCP 10.124.0.10:36526 => 140.82.121.4:80 [github.com.]
http    default/test-pod /usr/local/bin/curl github.com GET / 301 Moved Permanently 7.773241ms
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
dns     default/test-pod /usr/local/bin/curl [github.com.] => [140.82.121.4]
dns     default/test-pod /usr/local/bin/curl [github.com.] => []
connect default/test-pod /usr/local/bin/curl TCP 10.124.0.10:60770 => 140.82.121.4:443 [github.com.]
tls     default/test-pod /usr/local/bin/curl 140.82.121.4:443 github.com TLS1.3 TLS_AES_128_GCM_SHA256
exit    default/test-pod /usr/local/bin/curl -L github.com 0
close   default/test-pod                                                    hub.com.] tx 74 B rx 85 B
close   default/test-poc                                                    hub.com.] tx 701 B rx 218 k
```

Eksempel med Tetragon

sopra steria

Network Policies

- **Muliggjør trafikkstyring på lag 3 og 4**
  - IP, Port, Protocol, Pod label
  - **Som standard I Kubernetes er alt åpnet**
- **Krever en nettverk CNI som kan styre reglene**
  - Calico, WeaveNet, Azure CNI, GKE CNI, Cilium (eBFP)
  - **Start med en deny-all regel**
  - Flannel (støtter ikke Network Policies)
  - Styring av regler via YAML konfig

Noen gratis verktøy for enkel visualisering
https://orca.tufin.io/netpol/
https://artturik.github.io/network-policy-viewer/

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: frontend-to-sqldatabase
  namespace: default
spec:
  podSelector:
    matchLabels:
      app: sqldatabase
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
    - namespaceSelector:
        matchLabels:
          project: myproject
    - podSelector:
        matchLabels:
          role: frontend
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
    ports:
    - protocol: TCP
      port: 5978
```

Destinasjon

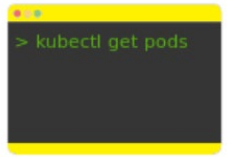
Source IP

Kilde

Target IP og Port

sopra steria

Sikkerhetsmonitorering



Kubectl apply -f

Kubernetes API overvåkning

DNS Logs fra CoreDNS

Host logger

CNI logger

Kubernetes Audit Policy

Audit Level

- None
- Metadata
- Request
- RequestResponse

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
- "RequestReceived"
rules:
- level: RequestResponse
resources:
- group: ""
resources: ["pods"]
- level: Metadata
resources:
- group: ""
resources: ["pods/log", "pods/status"]
```

Log tjeneste

- Splunk
- Sentinel
- ELK
- Loki

sopra steria

Kommersielle løsninger

- ✓ Sikker kode utvikling
- ✓ Sikre avhengigheter
- ✓ Container image scanning etter sårbarheter
- ✓ Sikker IaC koding

- ✓ Sikker kode utvikling
- ✓ Sikker IaC koding

- ✓ Cloud og Kubernetes Security Posture
- ✓ Sikker kode utvikling
- ✓ Sikring av Pipeline
- ✓ Sikker IaC koding
- ✓ Vulnerability Management
- ✓ Container image scanning etter sårbarheter

- ✓ Sikker IaC koding
- ✓ Vulnerability Management
- ✓ Container image scanning etter sårbarheter
- ✓ Cloud og Kubernetes Security Posture

sopra steria

Hvor bør jeg egentlig starte?

Basis mekanismer	Neste nivå	Per use-case
✓ Private Cluster ✓ Identitetsbasert tilgang (iDP) ✓ Sikkerhetsscanning i Kode, container og avhengigheter ✓ Nettverk Policies ✓ Tilgangskontroll ✓ Kontroll på versjonering	✓ Workload Identity ✓ Sikkerhetsmonitorering ✓ Ekstern secret håndtering ✓ Nettverk Policies ✓ Tilgangskontroll basert på Zero-Trust prinsipper ✓ GitOps ✓ Identitetskontroll via Sync/SCIM	✓ Service Mesh ✓ Backup ✓ Kubebench (CIS, NIST validering) ✓ gVisor og/eller confidental computing ✓ Andre kommersielle løsninger ✓ Falco eller Tetragon eBFP
Bygg forståelse!	**Opplæring med de som skal bruke det**	**Start med «quick wins»**


Takk for meg!
Marius.sandbu@soprasteria.com