IT-IMPRESSIVE

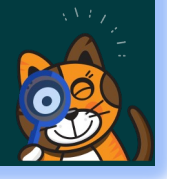
Detron

AKS enabled, what's next?

Experience out of the field

Sprekers:
Dinant Paardenkooper – Architect/Consultant

Topics:
- Wat is Kubernetes en wat kan ik er mee
- Praktische usecase
- Workshop Calico CNI en NetworkPolicies
- Optioneel Container Security



Introductie

Dinant Paardenkooper

Rol: Handson Cloud Native Solution Architect (Azure, VMWare)
Cloud Native | Kubernetes | Automation | IaC | Spreker

Drive: Innovatie, Business requirements transformeren naar praktische toepasbare IT-solutions

Hobby's: Gitaar spelen, innovatie, hardlopen, squash

E-mail: d.paardenkooper@IT-Impressive.nl
LinkedIn: www.linkedin.com/in/dinantpaardenkooper
Blog: https://dinantpaardenkooper.nl/posts

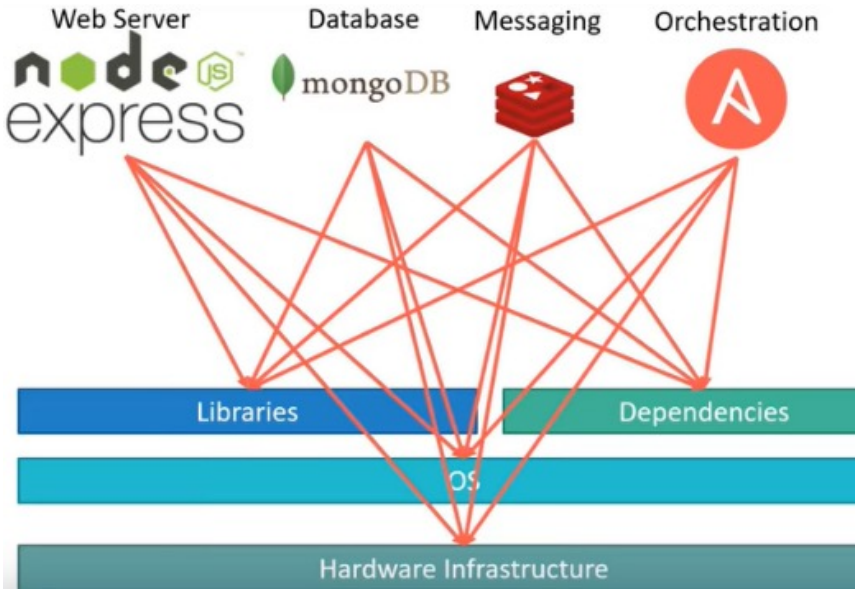
Agenda

Containers en K8s	-	Wat is het en wat kan ik ermee?
Architectuur	-	Kubernetes onder de motorkap
Azure services	-	Wat biedt Microsoft Azure?
Praktijk Usecase	-	AKS aan, en nu?
Handson	-	Calico CNI & network policies
Optioneel	-	Container Security

Q & A

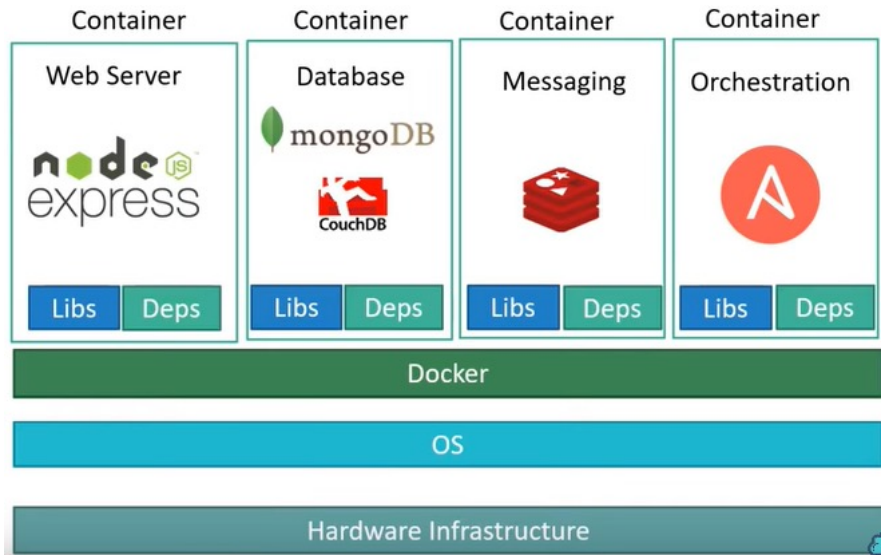
What is a container?

Power of containers

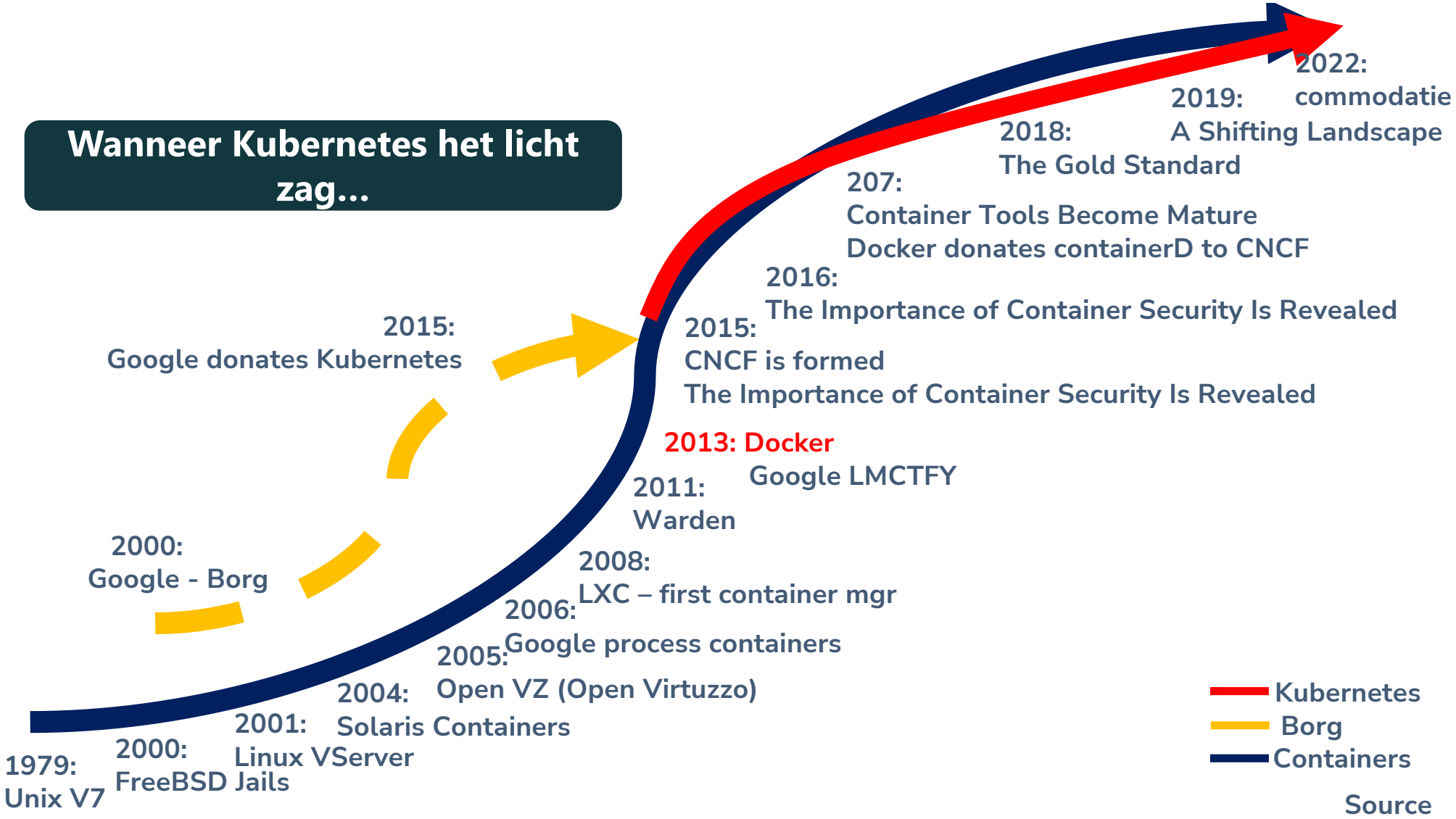


Traditioneel

VS



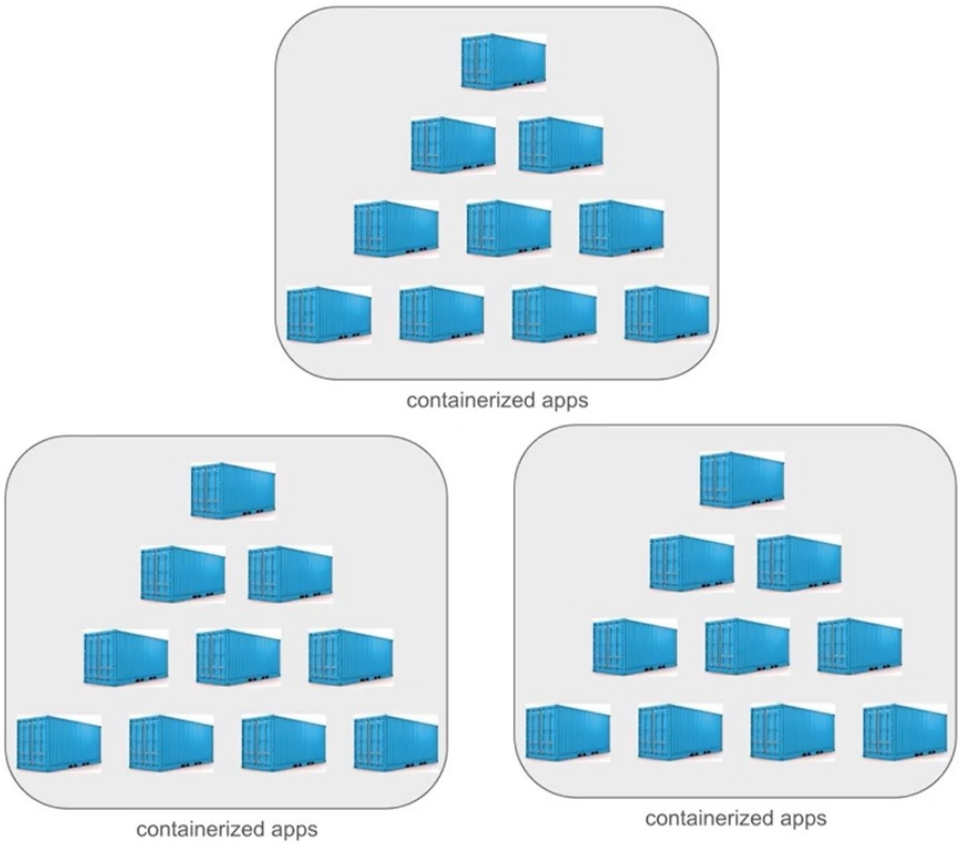
Containerized


Wanneer Kubernetes het licht zag...
1979:
Unix V7
2000:
FreeBSD Jails
2001:
Linux VServer
2004:
Solaris Containers
2005:
Open VZ (Open Virtuzzo)
2006:
Google process containers
2008:
LXC – first container mgr
2011:
Warden
2013: Docker
Google LMCTFY
2015:
CNCF is formed
The Importance of Container Security Is Revealed
2015:
Google donates Kubernetes
2000:
Google - Borg
2016:
The Importance of Container Security Is Revealed
207:
Container Tools Become Mature
Docker donates containerD to CNCF
2018:
The Gold Standard
2019:
A Shifting Landscape
2022:
commodatie
Kubernetes
Borg
Containers
Source

Nieuwe uitdagingen

Hoe op te lossen…?

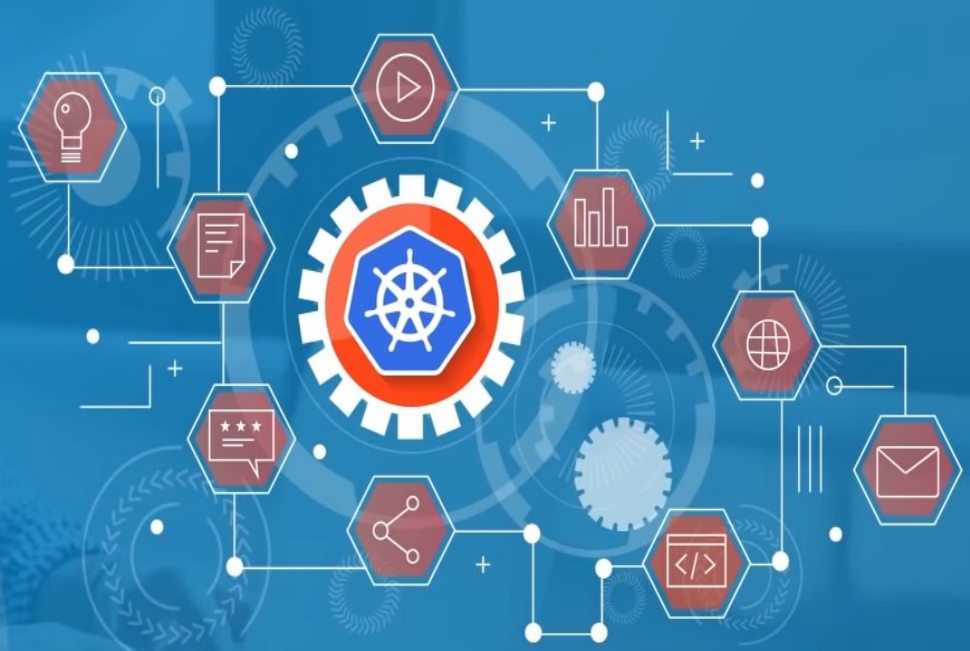
- Schalen
- Support
- Loadbalancing
- Storage
- Security
- RBAC
- En meer …




Kubernetes

What is Kubernetes?

Knowledge check

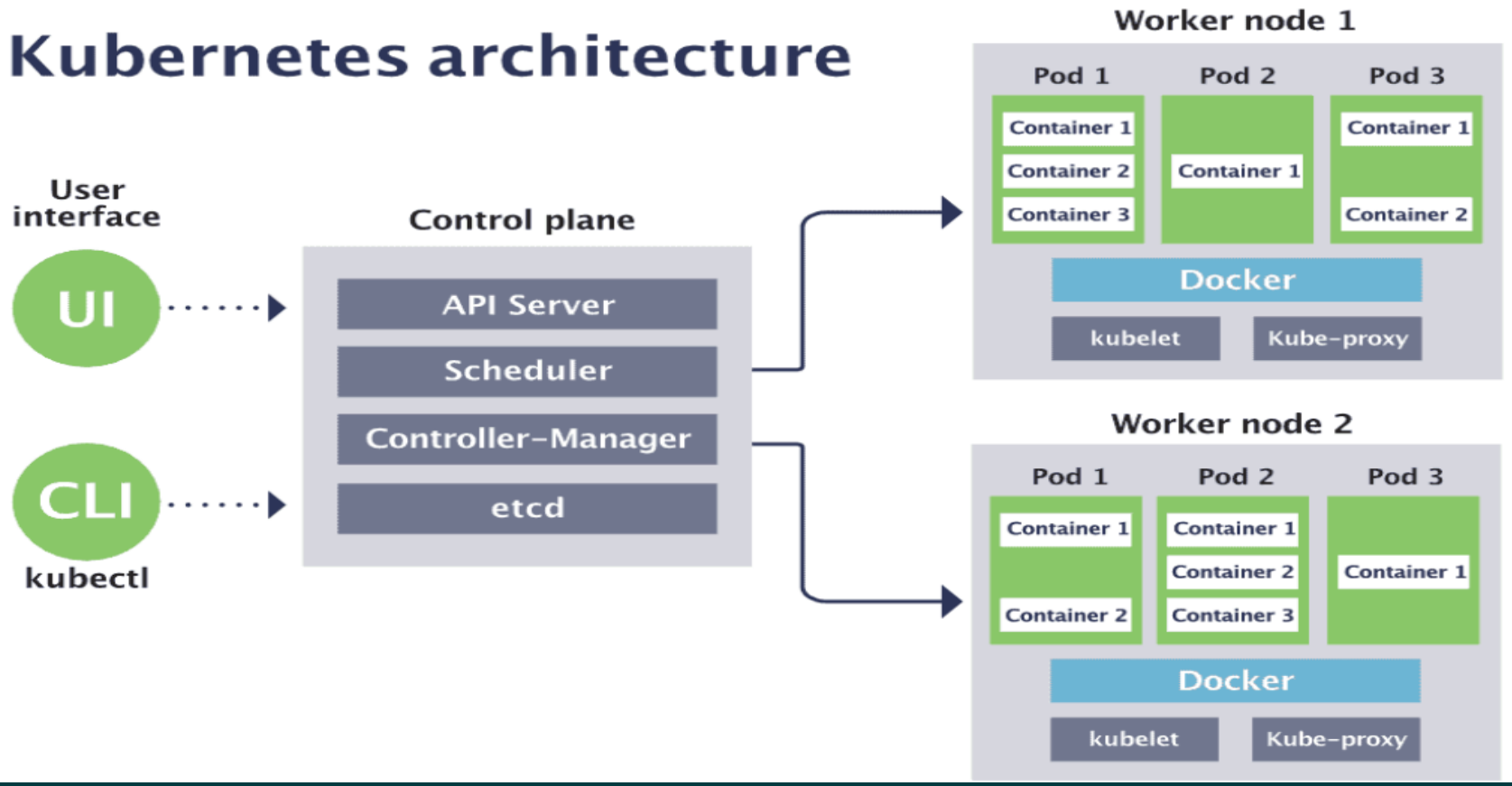


Download Mentimeter op je smartphone!

- Vul de code ….
- Klik “Join”

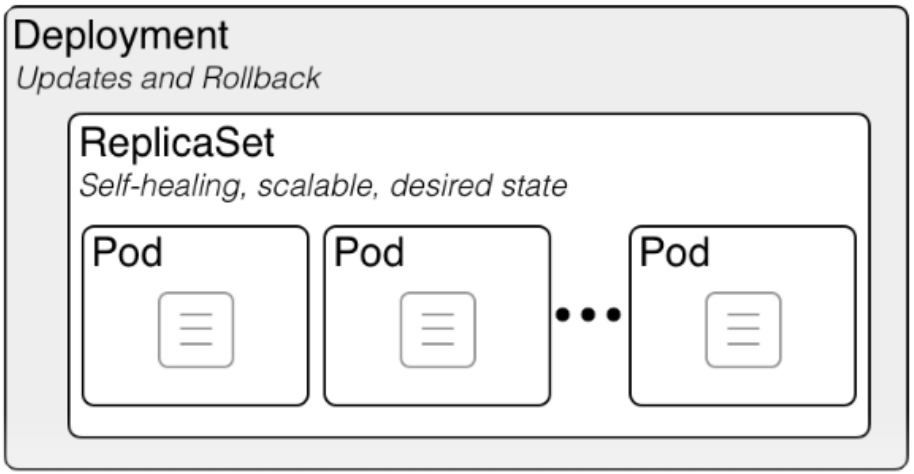
Architectuur

Kubernetes architecture



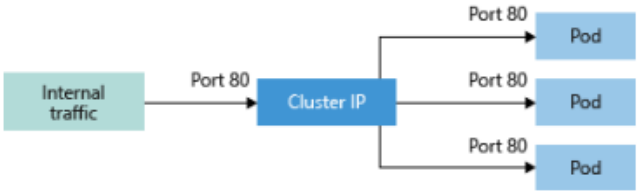
Terminologiën (1/2)

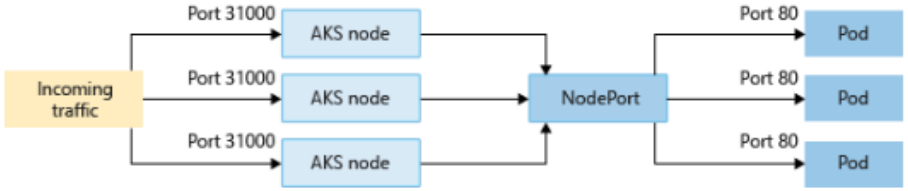
- Pod;
- Deployment;
- ReplicaSet;
- PersistentVolume en PersistentVolumeClaim;

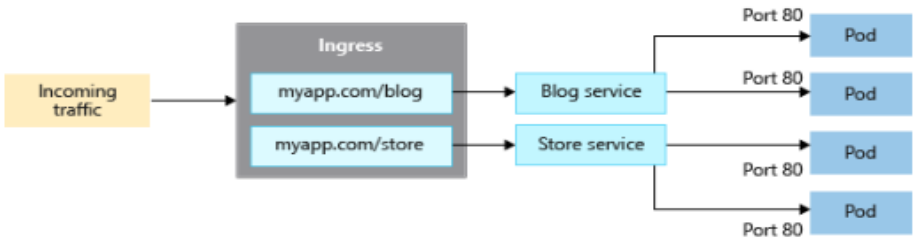


Terminologiën (2/2)

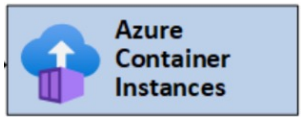
- Services
- Ingress

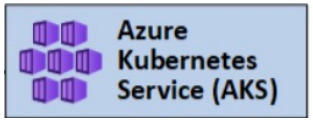


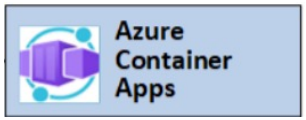


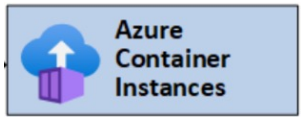


Wat biedt Azure?


Azure
Container
Instances

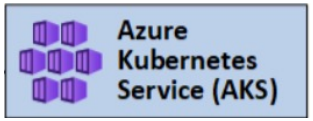

Azure
Kubernetes
Service (AKS)


Azure
Container
Apps



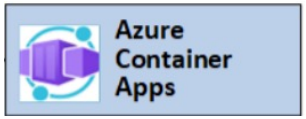
Usecase:

- Enkele ContainerApp;
- Snel bijschalen infra;
- Geen Infra beheer.



Usecase:

- Deploy via CI/CD;
- Flexibel met resources;
- Beheer enkel workernodes.

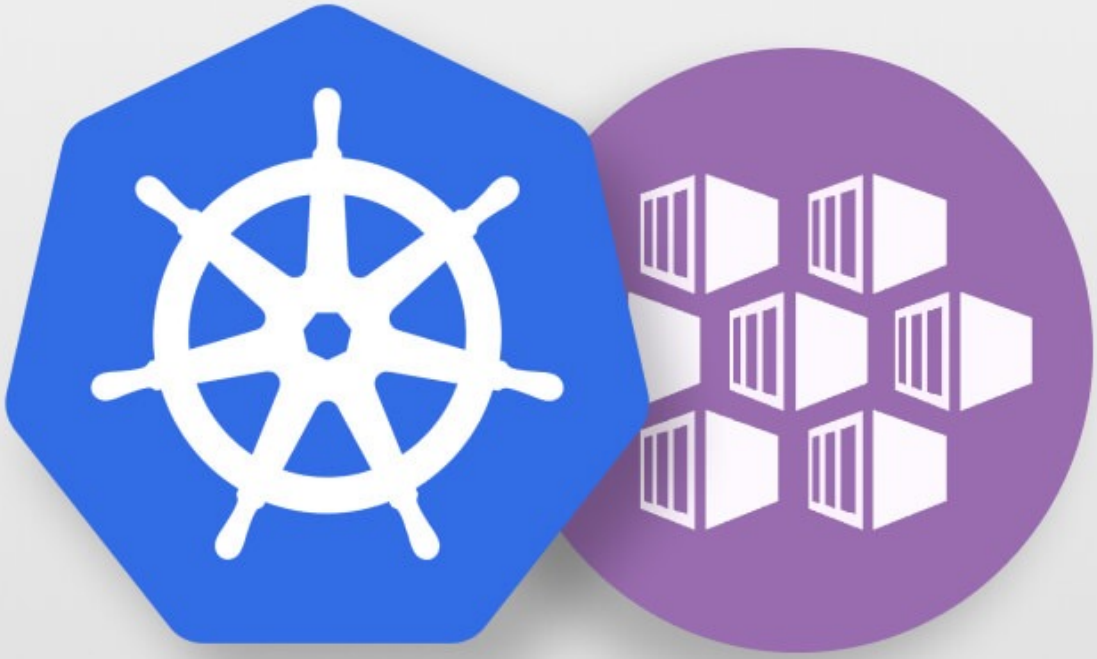


Usecase:

- Deploy files via Dappr;
- Alles weggeprovisioned;
- Geen AKS beheer.

Bron

Usecase AKS



Enabled, What's next?

Ontwerpkeuzes?

Autorisaties?

Bring your own CNI?

Encryptie?

Public / Private AKS?

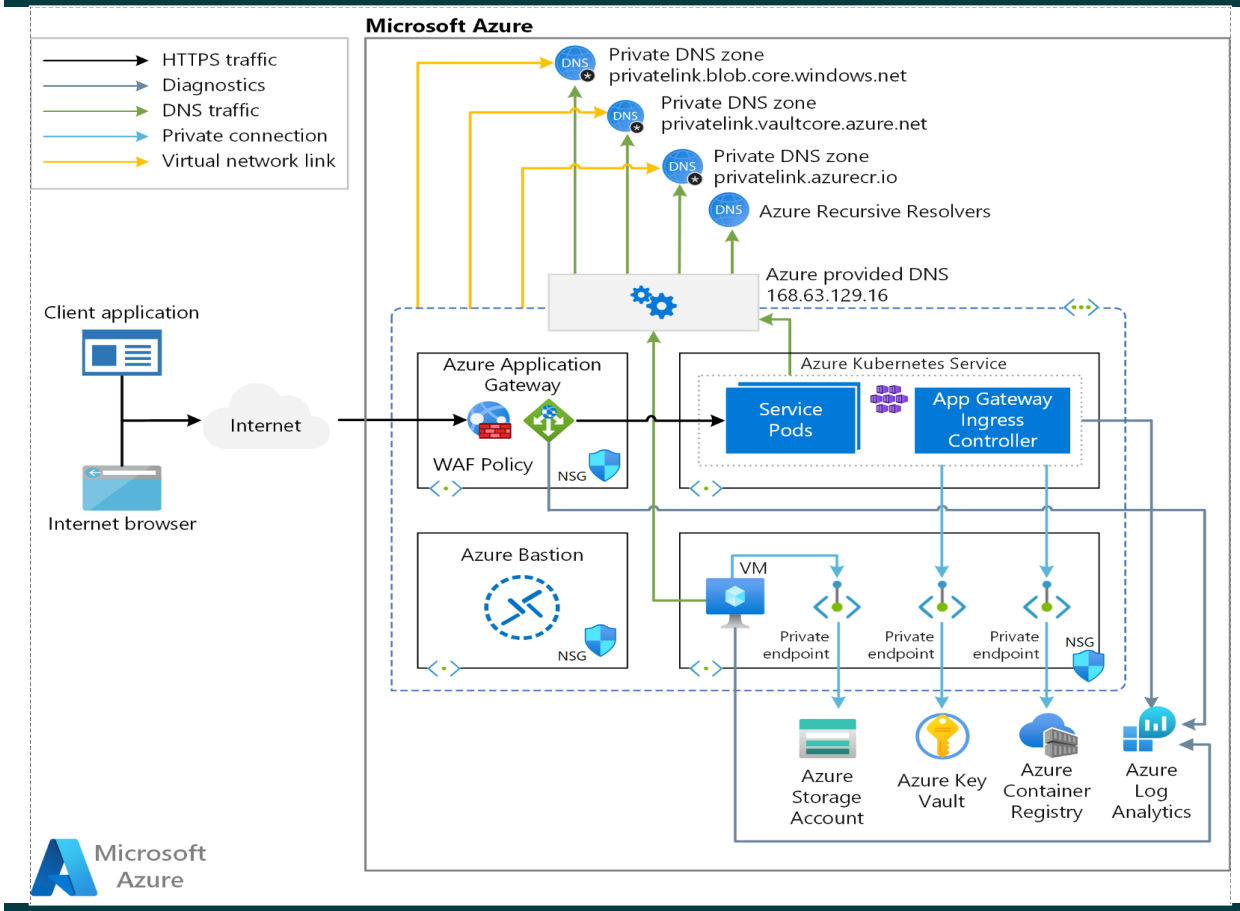
Hoe regel ik Storage?

Wat moet ik beheren

DB containers?

Security?

Upgrading?

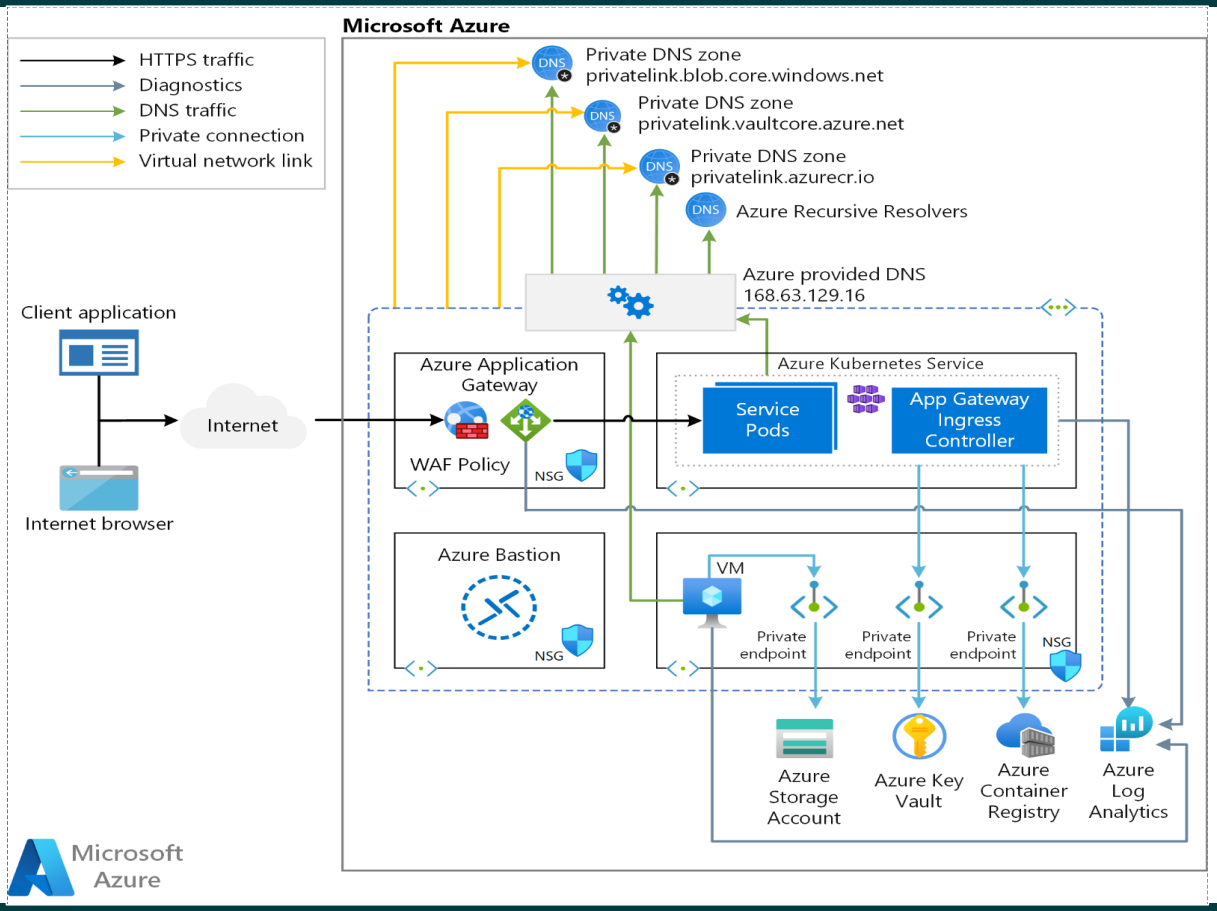


Wat biedt AKS ons?

Platform:

- AKS Private / Public;
- Add-ons (Azure Policy, AGIC, KEDA, CSI, KEDA);
- Hardend OS (o.b.v. CIS benchmark);
- Ingress Controller (AGIC / Nginx);
- Registry (ACR);
- Netwerk, AppGW en LoadBalancer;
- CNI (Kubenet / AzureCNI);
- Persistent storage (CSI driver);
- Secret management (Azure KeyVault);
- Logging (LogAnalytics);
- Monitoring (ContainerInsight);
- Authenticatie en autorisatie (AzureAD / local acc);
- Patching (KURED) en ClusterAutoScale enabled;
- SLA (Paid / Free);
- Container Security (Defender for Cloud);
- Scheiding op basis van Agent pools;
- Kost management (Azure Tags).

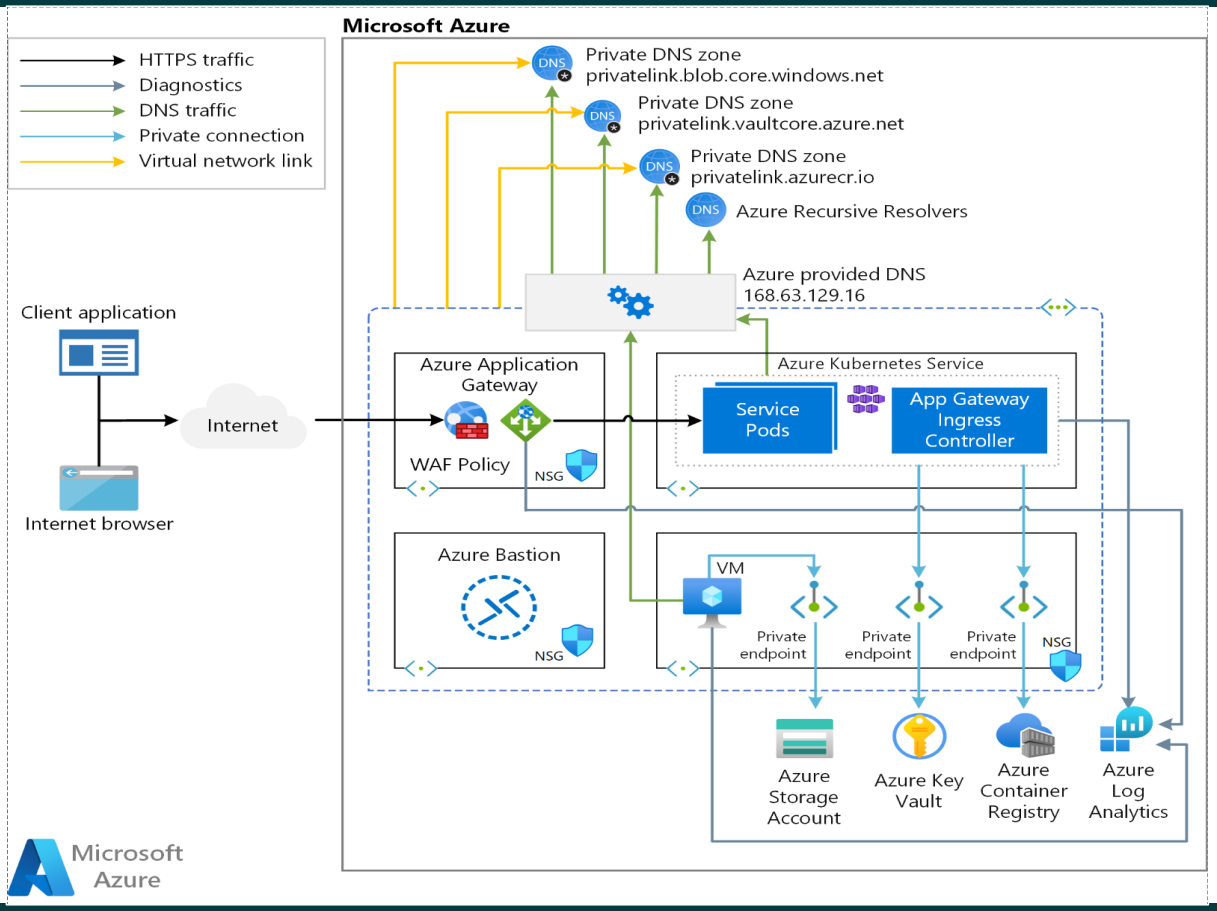
Bron



Wat biedt Microsoft Azure?

Automation (m.b.v. extra tools):

- Deployment via Infra as Code
- Automatisering DNS en Certificaat management;
- Application Deployment.



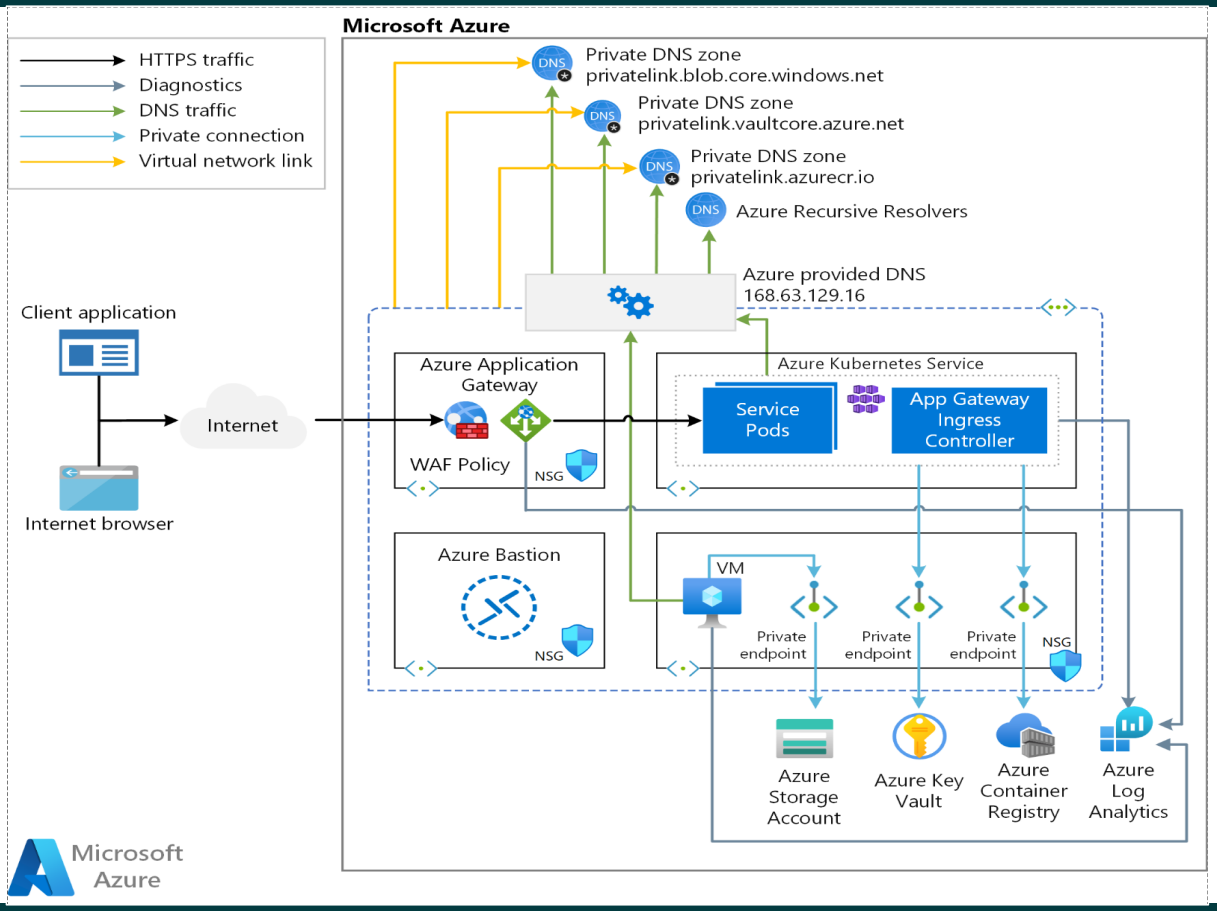
Wat biedt Microsoft Azure?

Application:

- APM via AppInsight;
- CI/CD tooling via AzureDevOps/GitHub;
- Code Repository via AzureDevOps/GitHub;
- Custom image build via VMScalesets;
- Geen backup.

Azure DevOps

GitHub Actions



Wat biedt Microsoft Azure?

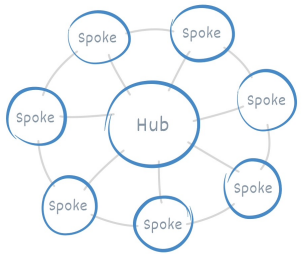
Security:

- System of User Managed Identities;
- Registry scanning via Defender for Cloud;
- CSI driver via KeyVault;
- Hardening Kubernetes via Azure Policy;
- Namespace isolatie via Calico Policies;
- AKS best practices;
- Azure Security Benchmark v3.

Open Policy Agent

Praktijk usecase

Strategie



- Azure Cloud Native, tenzij...;
- SAAS, boven PAAS, boven IAAS;
- Geen Tech-Preview features, alleen GA;
- Alles Infra as Code.

Greep van de Requirements

- Security en Compliancy (SCF, SMCF, DPIA, ISO/IEC 27001:2013);
- Workload Data verkeer scheiding workloads;
- Workload teams faciliteren;
- Ondersteuning DevOps Way of Working;
- Azure Best Practices;
- Fully-Automated Infrastructuur incl. Patch management;
- Kostmanagement.

Provisioning

- SharedAKS en SharedACR voor alle workloads;
- Namespace en serviceaccount;
- Netwerkverkeersstromen (base.json);
- Listeners AppGW's en Certificaat ingestion;
- KeyVault, DB, LAW en storage account koppeling;
- agentpool o.b.v. opgegeven vmsize.

Beheer

- Alle Deployment “Infra as Code”);
- Mogelijkheid deployment agentpool;
- Updates via KURED, ingesteld 1 x per week;
- Upgrade via separate Pipeline;

Workload team

- AzureDevOps (project, ado agent, SVC SharedAKS o.b.v. service account);
- Afstemming Kubernetes manifest files en pipeline.

Security

ACR

* Container image build via VMscaleset;
* Alleen VMScaleset MI ACR-Push permissies;
* Alleen SharedAKS MI ACR-Pull permissies;
* Baseline ACR;
* Azure policies.

AKS;

* Compliancy: SCF, SMCF, DPIA, ISO/IEC 27001:2013
* CIS Kubernetes benchmark;
* CIS Ubuntu;
* Azure Security Benchmark v3;
* Microsoft Cloud Security Benchmark;
* Baseline AKS;
* Azure Policies.

Security

ACR

* Container image build via VMscaleset;
* Alleen VMScaleset MI ACR-Push permissies;
* Alleen SharedAKS MI ACR-Pull permissies;
* Baseline ACR;
* Azure policies.

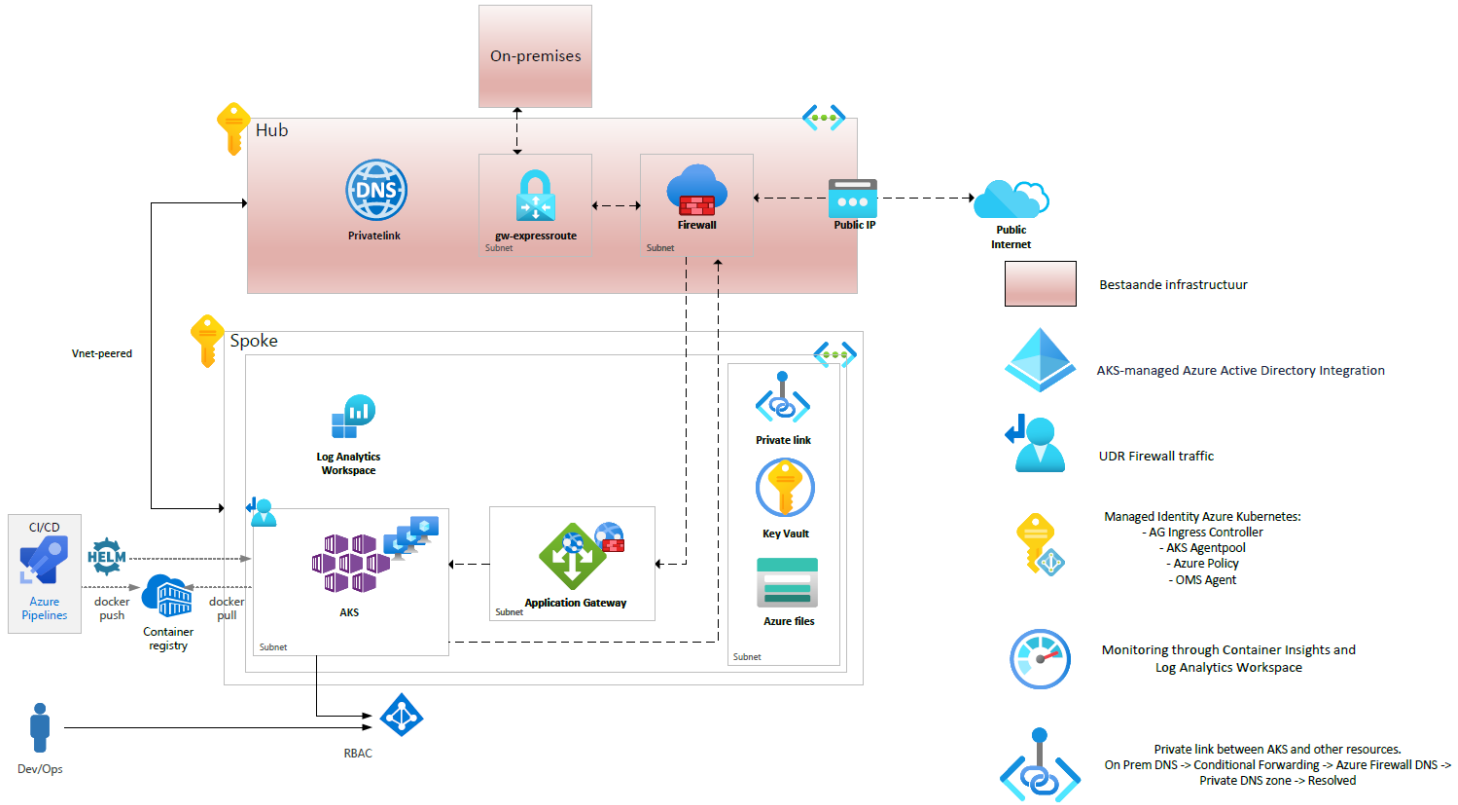
AKS;

* Compliancy: SCF, SMCF, DPIA, ISO/IEC 27001:2013
* CIS Kubernetes benchmark;
* CIS Ubuntu;
* Azure Security Benchmark v3;
* Microsoft Cloud Security Benchmark;
* Baseline AKS;
* Azure Policies.

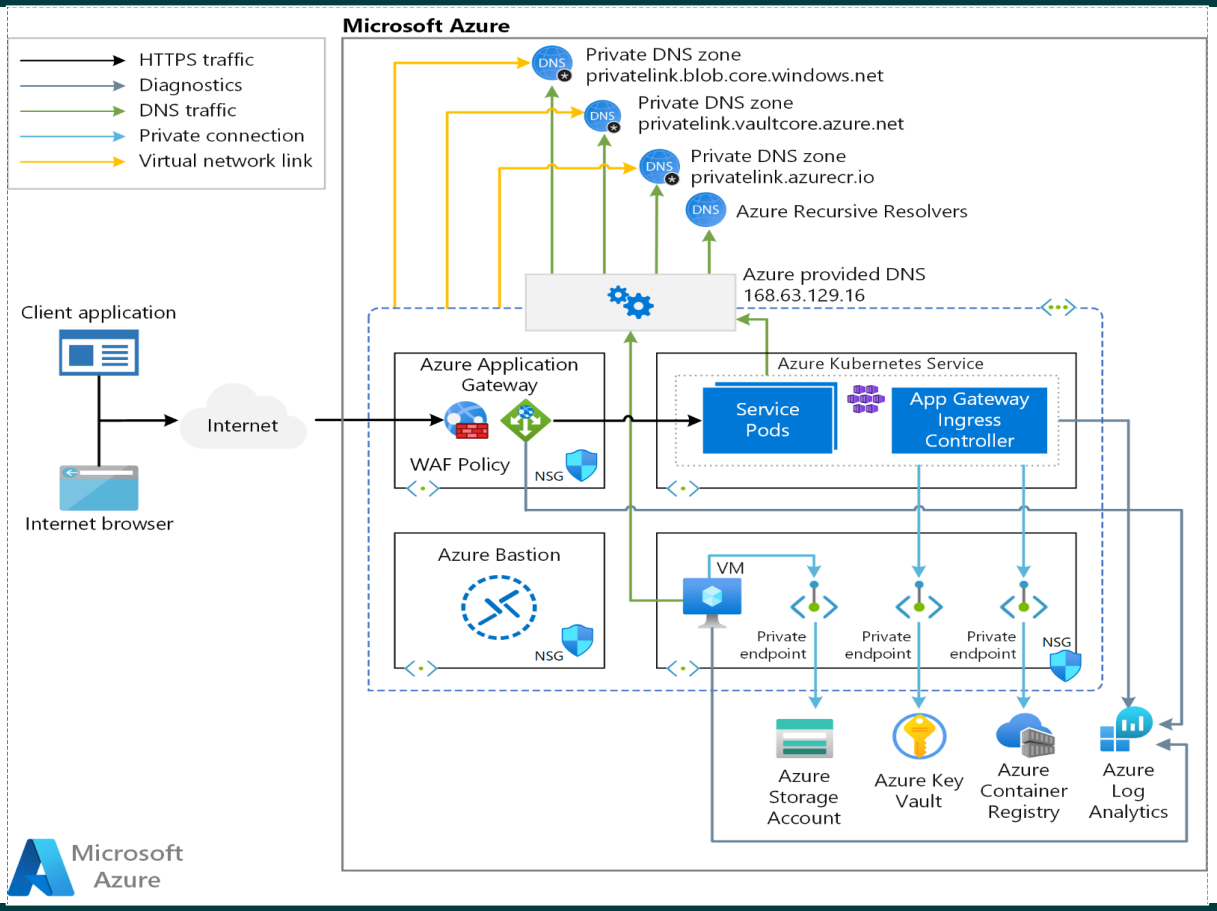
Domain	Control ID	Control title	Policy (Azure portal)
Network Security	NS-2	Establish network segmentation boundaries	Container registries should not allow unrestricted network access
Network security	NS-2	Secure cloud services with network controls	Container registries should use private link
Network security	NS-2	Secure cloud services with network controls	Configure Container registries to disable public network access
Network security	NS-2	Secure cloud services with network controls	Configure Container registries with private endpoints
Network security	NS-2	Secure cloud services with network controls	Container registries should have SKUs that support Private Links
Network security	NS-2	Secure cloud services with network controls	Public network access should be disabled for Container registries
Identity management	IM-1	Use centralized identity and authentication system	Configure container registries to disable anonymous authentication.
Identity management	IM-1	Use centralized identity and authentication system	Configure container registries to disable ARM audience token authentication.
Identity management	IM-1	Use centralized identity and authentication system	Configure container registries to disable local admin account.
Identity management	IM-1	Use centralized identity and authentication system	Configure container registries to disable repository scoped access token.
Identity management	IM-1	Use centralized identity and authentication system	Container registries should have anonymous authentication disabled.
Identity management	IM-1	Use centralized identity and authentication system	Container registries should have ARM audience token authentication disabled.
Data Protection	DP-5	Use customer-managed key option in data at rest encryption when required	Container registries should be encrypted with a customer-managed key
Data protection	DP-2	Monitor anomalies and threats targeting sensitive data	Container registries should have exports disabled
Privileged access	PA-1	Separate and limit highly privileged/administrative users	Container registries should have local admin account disabled.
Identity management	IM-1	Use centralized identity and authentication system	Container registries should have repository scoped access token disabled.
Logging and threat detection	LT-1	Enable threat detection capabilities	Container registry images should have vulnerability findings resolved

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Security	NS-2	Secure cloud services with network controls	Authorized IP ranges should be defined on Kubernetes Services	2.0.1
Privileged Access	PA-7	Follow just enough administration (least privilege) principle	Role-Based Access Control (RBAC) should be used on Kubernetes Services	1.0.2
Data Protection	DP-3	Encrypt sensitive data in transit	Kubernetes clusters should be accessible only over HTTPS	8.0.1
Logging and Threat Detection	LT-1	Enable threat detection capabilities	Azure Kubernetes Service clusters should have Defender profile enabled	2.0.0
Logging and Threat Detection	LT-2	Enable threat detection for identity and access management	Azure Kubernetes Service clusters should have Defender profile enabled	2.0.0
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	[Preview]: Kubernetes clusters should gate deployment of vulnerable images	2.0.1-preview
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Azure Policy Add-on for Kubernetes service (AKS) should be installed and enabled on your clusters	1.0.2
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers CPU and memory resource limits should not exceed the specified limits	9.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers should not share host process ID or host IPC namespace	5.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers should only use allowed AppArmor profiles	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers should only use allowed capabilities	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers should only use allowed images	9.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster containers should run with a read only root file system	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster pod hostPath volumes should only use allowed host paths	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster pods and containers should only run with approved user and group IDs	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster pods should only use approved host network and port range	6.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster services should listen only on allowed ports	8.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes cluster should not allow privileged containers	9.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes clusters should disable automounting API credentials	4.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes clusters should not allow container privilege escalation	7.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes clusters should not grant CAP_SYS_ADMIN security capabilities	5.0.1
Posture and Vulnerability Management	PV-2	Audit and enforce secure configurations	Kubernetes clusters should not use the default namespace	4.0.1
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	Running container images should have vulnerability findings resolved	1.0.1
DevOps Security	DS-6	Enforce security of workload throughout DevOps lifecycle	Running container images should have vulnerability findings resolved	1.0.1

Eindresultaat




Security Spectrum
Wat wordt er niet afgedekt?



Wat biedt AKS ons (nog) NIET?

Platform (still Tech Preview):

- AKS Image Cleaner;
- MarinerOS;
- AKS Fleet Manager;
- Confidential Compute Nodes;
- Bring your own CNI (Cilium);
- ASO database connection provisioning.

Security (still Tech Preview):

- CI/CD scanning en runtime policies;
- Workload Identities

Application:

- Logging / Monitoring.

Roadmap ACR
Roadmap AKS


Lessons learned
Multi-tenancy is mogelijk
Leercurve is stevig
Veel in Tech Preview
Veel controle mogelijk
Confidential is Toekomst
CI/CD scanning (missend)
Workload Identity (aad)
Controle in Kubernetes?


Break


Workshop

Bekijken Calico i.c.m. met AKS vanuit oogpunt:

- Netwerk;
- Security;
- Observability.

Benodigdheden:

- Computer met internet toegang;
- Actieve Azure Subscription.

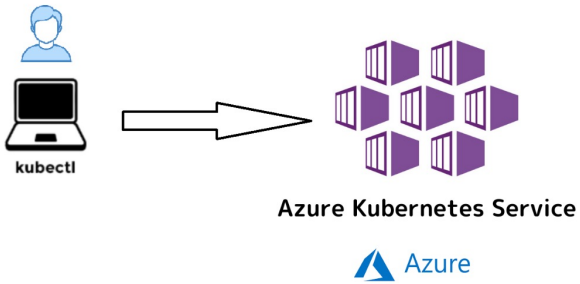
Github Repo:

https://github.com/erecica/Calico-AKS-BYOCNI
https://github.com/tigera-solutions/calicocloud-aks-workshop

Installatie stappen 1/3:

AKS installatie

1) Maak een Resource Group aan voor deze workshop;
2) Creëer een AKS Cluster zonder Kubernetes CNI pre-installed*;
3) Bekijk en gebruik de credentials om toegang te krijgen tot het AKS cluster met kubectl.



* Note: duurt ongeveer tussen de 6 a 9 minuten

Resultaat na stap 1 tot 3:

```
jurgen [ ~ ]$ kubectl get nodes -o wide
N De knop Terminal-container                STATUS     ROLES   AGE     VERSION    INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
aks-nodepool1-17182512-vmss000000   NotReady   agent   4m3s    v1.23.12   10.224.0.4    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
aks-nodepool1-17182512-vmss000001   NotReady   agent   3m55s   v1.23.12   10.224.0.5    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
aks-nodepool1-17182512-vmss000002   NotReady   agent   4m      v1.23.12   10.224.0.6    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
jurgen [ ~ ]$ kubectl cluster-info
Kubernetes control plane is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443
CoreDNS is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
jurgen [ ~ ]$ kubectl get ns
NAME              STATUS   AGE
default           Active   6m5s
kube-node-lease   Active   6m6s
kube-public       Active   6m6s
kube-system       Active   6m6s
jurgen [ ~ ]$ kubectl get pods -A
NAMESPACE     NAME                                  READY   STATUS    RESTARTS   AGE
kube-system   cloud-node-manager-6pqq6              1/1     Running   0          4m50s
kube-system   cloud-node-manager-7gpgf              1/1     Running   0          4m55s
kube-system   cloud-node-manager-fdnk4              1/1     Running   0          4m58s
kube-system   coredns-autoscaler-5589fb5654-9lzsv   0/1     Pending   0          6m6s
kube-system   coredns-b4854dd98-ttdzt               0/1     Pending   0          6m6s
kube-system   csi-azuredisk-node-mfrd9              3/3     Running   0          4m58s
kube-system   csi-azuredisk-node-ntj6v              3/3     Running   0          4m50s
kube-system   csi-azuredisk-node-wmcbd              3/3     Running   0          4m55s
kube-system   csi-azurefile-node-4ncpj              3/3     Running   0          4m55s
kube-system   csi-azurefile-node-jbwm8              3/3     Running   0          4m58s
kube-system   csi-azurefile-node-qz572              3/3     Running   0          4m50s
kube-system   konnectivity-agent-66f4b988c6-s464q   1/1     Running   0          6m5s
kube-system   konnectivity-agent-66f4b988c6-s99tm   1/1     Running   0          6m5s
kube-system   kube-proxy-2lslt                      1/1     Running   0          4m50s
kube-system   kube-proxy-4dpjq                      1/1     Running   0          4m55s
kube-system   kube-proxy-g6v9h                      1/1     Running   0          4m58s
kube-system   metrics-server-f77b4cd8-2k7m7         0/1     Pending   0          6m5s
kube-system   metrics-server-f77b4cd8-6svlc         0/1     Pending   0          6m5s
```

Installatie stappen 2/3:

Installatie en configuratie Calico (OSS)

4) Installatie van de Operator;
5) Configuratie van de Calico Installatie.

CNI verschillen:

Details	AKS + Azure CNI + Calico Network policy	AKS + Calico CNI + Calico Network Policy
Policy	Calico	Calico
IPAM	Azure	Calico
CNI	Azure	Calico
Overlay	No	VXLAN
Routing	VPC native	Calico
Datastore	Kubernetes	Kubernetes

Resultaat na stap 4 tot 5:

```
jurgen [ ~ ]$ kubectl get nodes -o wide
NAME                                STATUS     ROLES   AGE     VERSION    INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
aks-nodepool1-17182512-vmss000000   NotReady   agent   4m3s    v1.23.12   10.224.0.4    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
aks-nodepool1-17182512-vmss000001   NotReady   agent   3m55s   v1.23.12   10.224.0.5    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
aks-nodepool1-17182512-vmss000002   NotReady   agent   4m      v1.23.12   10.224.0.6    <none>        Ubuntu 18.04.6 LTS   5.4.0-1091-azure   containerd://1.5.11+azure-2
jurgen [ ~ ]$ kubectl cluster-info
Kubernetes control plane is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443
CoreDNS is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://calico-aks-calico-aks-resou-192e5c-fa5f8fb6.hcp.westeurope.azmk8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
jurgen [ ~ ]$ kubectl get ns
NAME              STATUS   AGE
default           Active   6m5s
kube-node-lease   Active   6m6s
kube-public       Active   6m6s
kube-system       Active   6m6s
jurgen [ ~ ]$ kubectl get pods -A
NAMESPACE     NAME                                  READY   STATUS    RESTARTS   AGE
kube-system   cloud-node-manager-6pqq6              1/1     Running   0          4m50s
kube-system   cloud-node-manager-7gpgf              1/1     Running   0          4m55s
kube-system   cloud-node-manager-fdnk4              1/1     Running   0          4m58s
kube-system   coredns-autoscaler-5589fb5654-9lzsv   0/1     Pending   0          6m6s
kube-system   coredns-b4854dd98-ttdzt               0/1     Pending   0          6m6s
kube-system   csi-azuredisk-node-mfrd9              3/3     Running   0          4m58s
kube-system   csi-azuredisk-node-ntj6v              3/3     Running   0          4m50s
kube-system   csi-azuredisk-node-wmcbd              3/3     Running   0          4m55s
kube-system   csi-azurefile-node-4ncpj              3/3     Running   0          4m55s
kube-system   csi-azurefile-node-jbwm8              3/3     Running   0          4m58s
kube-system   csi-azurefile-node-qz572              3/3     Running   0          4m50s
kube-system   konnectivity-agent-66f4b988c6-s464q   1/1     Running   0          6m5s
kube-system   konnectivity-agent-66f4b988c6-s99tm   1/1     Running   0          6m5s
kube-system   kube-proxy-2lslt                      1/1     Running   0          4m50s
kube-system   kube-proxy-4dpjq                      1/1     Running   0          4m55s
kube-system   kube-proxy-g6v9h                      1/1     Running   0          4m58s
kube-system   metrics-server-f77b4cd8-2k7m7         0/1     Pending   0          6m5s
kube-system   metrics-server-f77b4cd8-6svlc         0/1     Pending   0          6m5s
```

Installatie stappen 3/3:

Installatie van een applicatie

6) Uitrol van YAOBank;
7) Verifieren van onze deployment;
8) Deployment van een Loadbalancer;
9) Verifieren van de “Loadbalancer” deployment.

YAOBank app:

Voorbeeld app van Project Calico (Yust Another Online Bank).
Bestaat uit de volgende micro services:

- Customer (simpele Web GUI);
- Summery (middelware business logic);
- Database (persistent database).

Resultaat na stap 6 tot 9:

```
jurgen [ ~/Calico-AKS-BYOCNI ]$ kubectl get deployments -A
NAMESPACE          NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
calico-apiserver   calico-apiserver           2/2     2            2           7m34s
calico-system      calico-kube-controllers    1/1     1            1           8m17s
calico-system      calico-typha               2/2     2            2           8m17s
kube-system        coredns                    2/2     2            2           17m
kube-system        coredns-autoscaler         1/1     1            1           17m
kube-system        konnectivity-agent         2/2     2            2           17m
kube-system        metrics-server             2/2     2            2           17m
tigera-operator    tigera-operator            1/1     1            1           8m45s
yaobank-customer   customer                   1/1     1            1           4m58s
yaobank-database   database                   1/1     1            1           4m59s
yaobank-summary    summary                    2/2     2            2           4m58s
jurgen [ ~/Calico-AKS-BYOCNI ]$ kubectl get svc -A
NAMESPACE          NAME                              TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)          AGE
calico-apiserver   calico-api                        ClusterIP      10.0.205.85     <none>          443/TCP          7m49s
calico-system      calico-kube-controllers-metrics   ClusterIP      10.0.104.120    <none>          9094/TCP         7m54s
calico-system      calico-typha                      ClusterIP      10.0.138.84     <none>          5473/TCP         8m33s
default            kubernetes                        ClusterIP      10.0.0.1        <none>          443/TCP          17m
kube-system        kube-dns                          ClusterIP      10.0.0.10       <none>          53/UDP,53/TCP    17m
kube-system        metrics-server                    ClusterIP      10.0.99.77      <none>          443/TCP          17m
yaobank-customer   customer                          NodePort       10.0.215.5      <none>          80:30180/TCP     5m13s
yaobank-customer   yaobank-customer                  LoadBalancer   10.0.26.102     20.23.76.253    80:32281/TCP     4m4s
yaobank-database   database                          ClusterIP      10.0.207.12     <none>          2379/TCP         5m14s
yaobank-summary    summary                           ClusterIP      10.0.7.26       <none>          80/TCP           5m14s
jurgen [ ~/Calico-AKS-BYOCNI ]$
```

Activeren Global policy:

Installatie van een applicatie

1) Verificatie connectiviteit en Json vergelijk;
2) Toepassen default GlobalNetworkpolicy;
3) Toepassen applicatie networkpolicy;
4) Testen van de connectiviteit.

Github Repo:

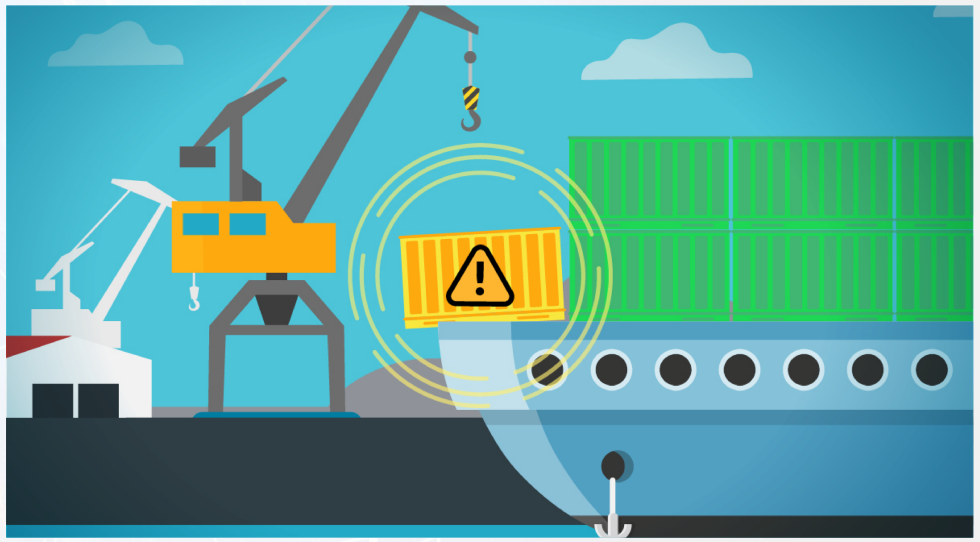
https://github.com/erecica/Calico-AKS-BYOCNI/tree/main/Exercises


Container Security

Types of Risico's

Type Risico's

- Container image;
- Container Registry;
- Kubernetes orchestration;
- Container (runtime);
- Operating System Kubernetes Nodes;



Container image

Defined riscs

- Vulnerabilities (CVE’s);
- Configuration defects;
- Embedded malware;
- Embedded clear text secrets;
- Untrusted images.

Mitigation measures

- Aqua can scan during build time (integration with Azure DevOps);
- Aqua can scan your Azure Container Registry;
- Aqua scans images on AKS hosts;
- Each image is scanned for vulnerabilities both in its OS packages and development language files.

Container Registry

Defined riscs

- Insecure connections;
- Stale images;
- Insufficient authentication and authorization restrictions.

Mitigation measures

- Only allow images from specific (trusted) container registries;
- Allows daily scans of images to alert on out-of-date vulnerable packages, base-images and versions;
- Allows the admin to define stale images via custom checks and block them from running;
- Can integrate automated scans into your CI processes to ensure only authorized images can be used.

Kubernetes orchestrator

Defined riscs

- Unbounded administrative access;
- Unauthorized access;
- Poorly inter-container connectivity;
- Mixing of workload sensitivity levels;
- Node trust.

Mitigation measures

- Audit logging;
- Set and enforce user access policies to container resources;
- Monitor user access, blocks, alerts unauthorized access attempts;
- Container Firewall limits network connectivity between workloads;
- Host integrity checks, including vulnerability scan, malware and CIS test to ensure nodes are secured.

Container

Defined riscs

- Vulnerabilities within runtime software;
- Unbounded network access from containers;
- Insecure container runtime configuration;
- Application vulnerabilities.

Mitigation measures

- Threat mitigation defenses detect and prevent port scanning;
- Threat mitigation defenses to detect and prevent connections to IP addresses with poor reputation;
- Real-time audit events on policy violations, report to SIEM tooling;
- Check for configuration drift;
- Block non-compliant images
- Block/allow certain executables;
- Prevent certain volumes to be mounted in a container;
- Manage and enforce seccomp profiles to unwanted syscalls;
- Log all container events.

Operating system

Defined riscs

- Attack surface;
- Shared kernel;
- Host OS component vulnerabilities;
- Improper user access rights;
- Host file system tampering.

Mitigation measures

- Scans host for vulnerabilities and malware against the Center for Internet Security (CIS) benchmarks (Docker, K8s);
- Logs user login and logout events on the host, including invocation of sudo programs;
- Scans hosts for configuration issues per the CIS Docker Benchmark;
- Restrict containers from specific mounting volumes or from writing into specific volumes or directories.

Container Security

aqua

Take aways

- Begrijp de omgeving
- Begin met een usecase
- Start klein
- DRY-Principe
- Repeat? Automatiseer
- Build & Test
- Just do it!
- Altijd Innoveren

Download Mentimeter op je smartphone!

- Vul de code ….
- Klik “Join”

IT-IMPRESSIVE

Detron

Dank voor jullie aanwezigheid