
"AKS Unlighted"
but what about Security, Cost and Multi-tenancy?
Devopsdays 2023

Introduction

Dinant Paardenkooper

Rol: Azure Cloud Native Architect | Consultant | IaC Speaker | Kubernetes | Automation | Security

Drive: Innovation, Business requirements transform to practical technical solutions

Hobby's: Play Gitar, Innovations, Running, Squash

E-mail: d.paardenkooper@it-impressive.nl

LinkedIn: www.linkedin.com/in/dinantpaardenkooper

Blog: https://dinantpaardenkooper.nl/posts/

Jurgen Allewijn

Rol: Cloud Architect| Speaker| Cloud Native | Azure | AWS | Kubernetes | Security

Drive: Innovation, IT transformations to achieve business needs

Hobby's: 3D Archery, gaming

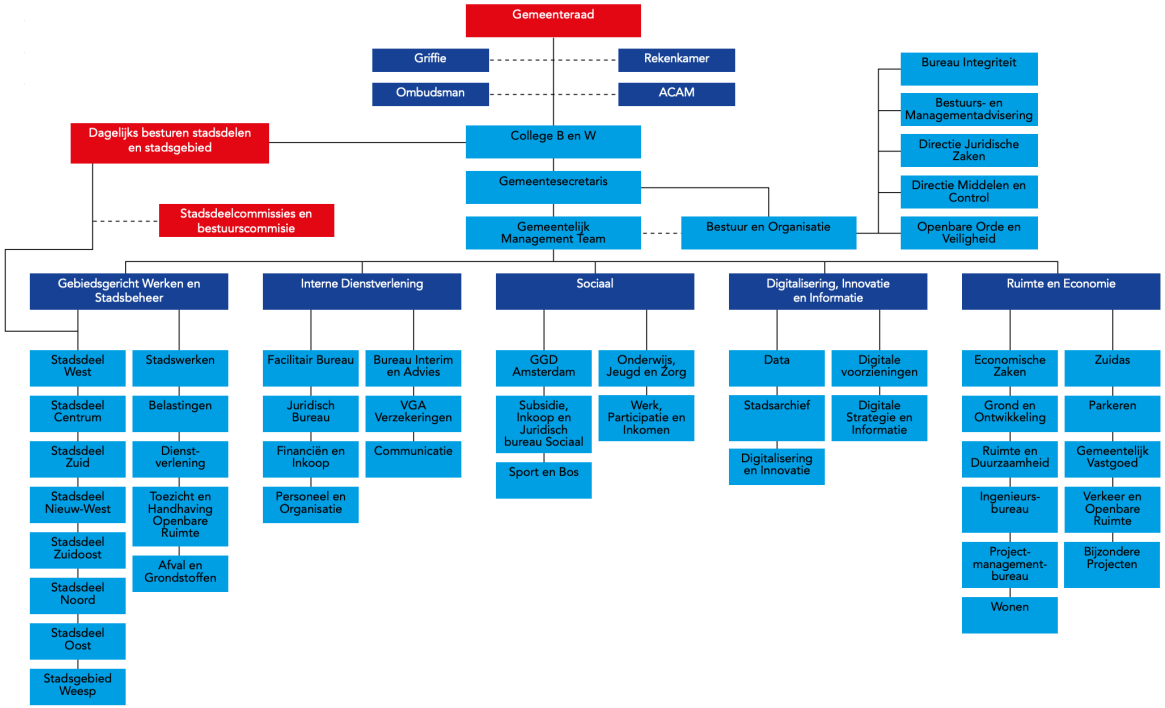
E-mail: jurgen.allewijn@luminis.eu

LinkedIn: https://www.linkedin.com/in/jallewijn/

Blog: https://jurgenallewijn.nl

Organisation

Who are we


Gemeenteraad
Griffie
Rekenkamer
Ombudsman
ACAM
Dagelijks besturen stadsdelen en stadsgebied
College B en W
Gemeentesecretaris
Stadsdeelcommissies en bestuurscommissie
Gemeentelijk Management Team
Bestuur en Organisatie
Bureau Integriteit
Bestuurs- en Managementadvisering
Directie Juridische Zaken
Directie Middelen en Control
Openbare Orde en Veiligheid
Gebiedsgericht Werken en Stadsbeheer
Interne Dienstverlening
Sociaal
Digitalisering, Innovatie en Informatie
Ruimte en Economie
Stadsdeel West
Stadsdeel Centrum
Stadsdeel Zuid
Stadsdeel Nieuw-West
Stadsdeel Zuidoost
Stadsdeel Noord
Stadsdeel Oost
Stadsgebied Weesp
Stadswerken
Belastingen
Dienst-verlening
Toezicht en Handhaving Openbare Ruimte
Afval en Grondstoffen
Facilitair Bureau
Juridisch Bureau
Financiën en Inkoop
Personeel en Organisatie
Bureau Interim en Advies
VGA Verzekeringen
Communicatie
GGD Amsterdam
Subsidie, Inkoop en Juridisch bureau Sociaal
Sport en Bos
Onderwijs, Jeugd en Zorg
Werk, Participatie en Inkomen
Data
Stadsarchief
Digitalisering en Innovatie
Digitale voorzieningen
Digitale Strategie en Informatie
Economische Zaken
Grond en Ontwikkeling
Ruimte en Duurzaamheid
Ingenieurs-bureau
Project-management-bureau
Wonen
Zuidas
Parkeren
Gemeentelijk Vastgoed
Verkeer en Openbare Ruimte
Bijzondere Projecten

Objectives

- **Efficient** use of available resources
- **Reliable and safe** (digital) services that are up to date.
- **Modern IT and data foundation**: reliable, secure, standardized, sustainable and agile.
- Future-proof **competencies**

IT Strategy

"Azure Cloud Native, unless..."

"SAAS, before PAAS, before IAAS"

"Only use Microsoft GA features"

"Infra as Code"

Principes and Compliancy

#	Principe
CP-B1	"Business is in the driver's seat": respond quickly and focussed to changing demand
CP-B2	Start small, evolve fast
CP-B3	Self Service, "5 minutes work" for DevOps teams
CP-B4	You Build It, You Run It (You Own it)
CP-P1	Adopt open internet and de facto industry standards
CP-P2	Available abstraction above DIY
CP-P3	Autonomous and self-managing DevOps teams
CP-P4	'Best of Platform' before 'Best of Breed'
CP-P5	Design based on Cloud Native concepts
CP-P6	'Security and Service Management by Design'
CP-P7	Automate everything

#	Compliancy standard
1	BIO
2	ISO/IEC 27001:2013
3	SCF
4	SMCF
5	DPIA
6	NORA

CP-B = Cloud Principe – Business, CP-P = Cloud Principe – Platform


Why SharedAKS?
Business Case
Callenge 1:
Facilitate Workload Teams
Challenge 2:
Cost, Sec and Compliancy
Challenge 3:
Technical Challenges

Facilitate Workload teams

Work with newest technology

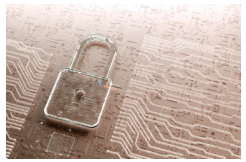
DevOps way of working

Centralize K8S expertise

Cost, Security and Compliancy

Minimize costs

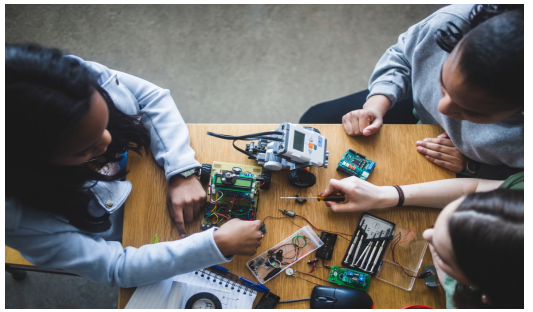
Secure & up to date

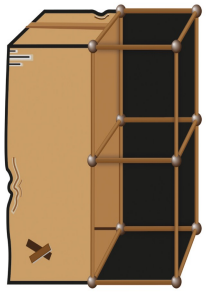


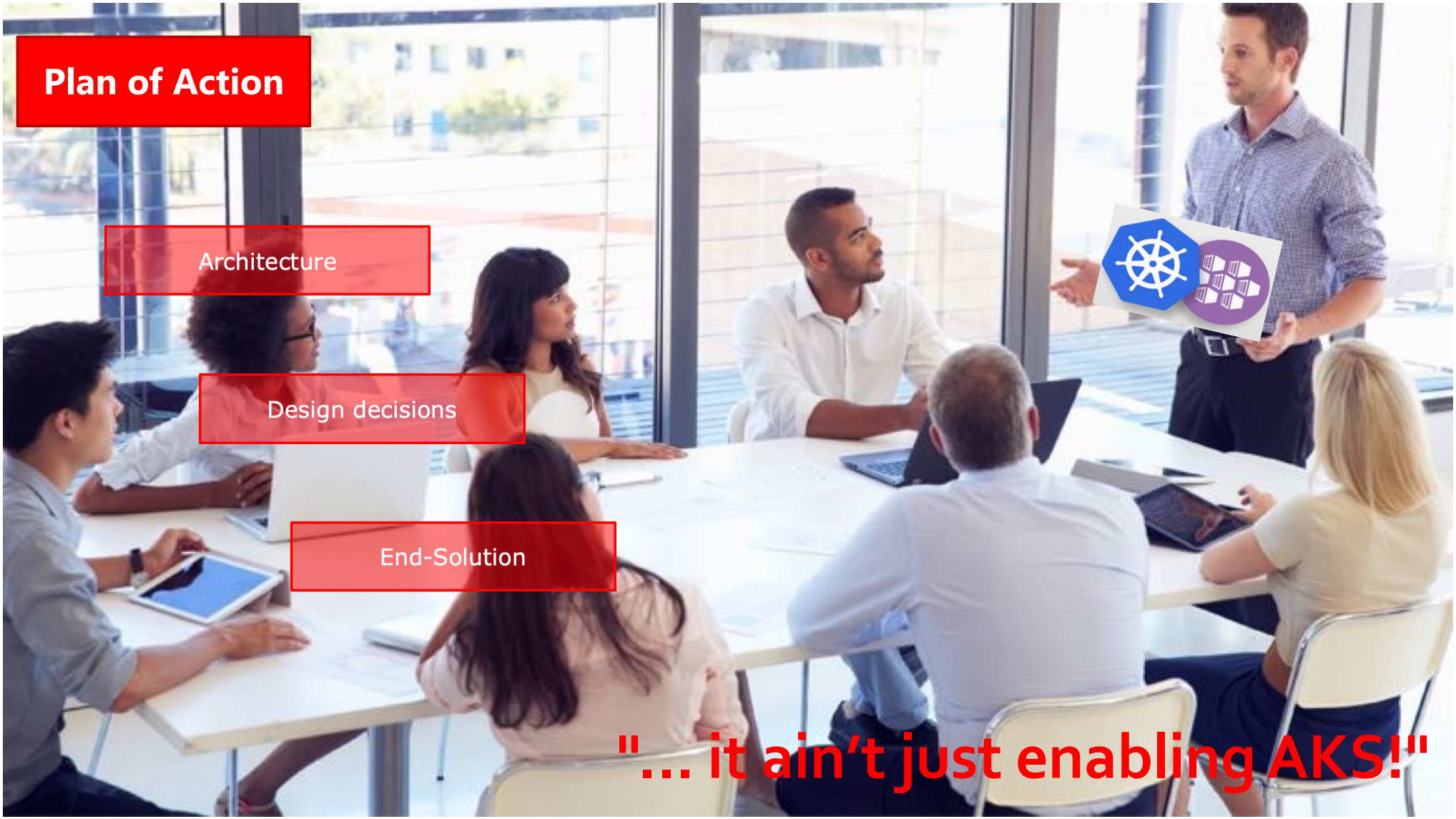
Compliancy

Automate

Technical Challenges



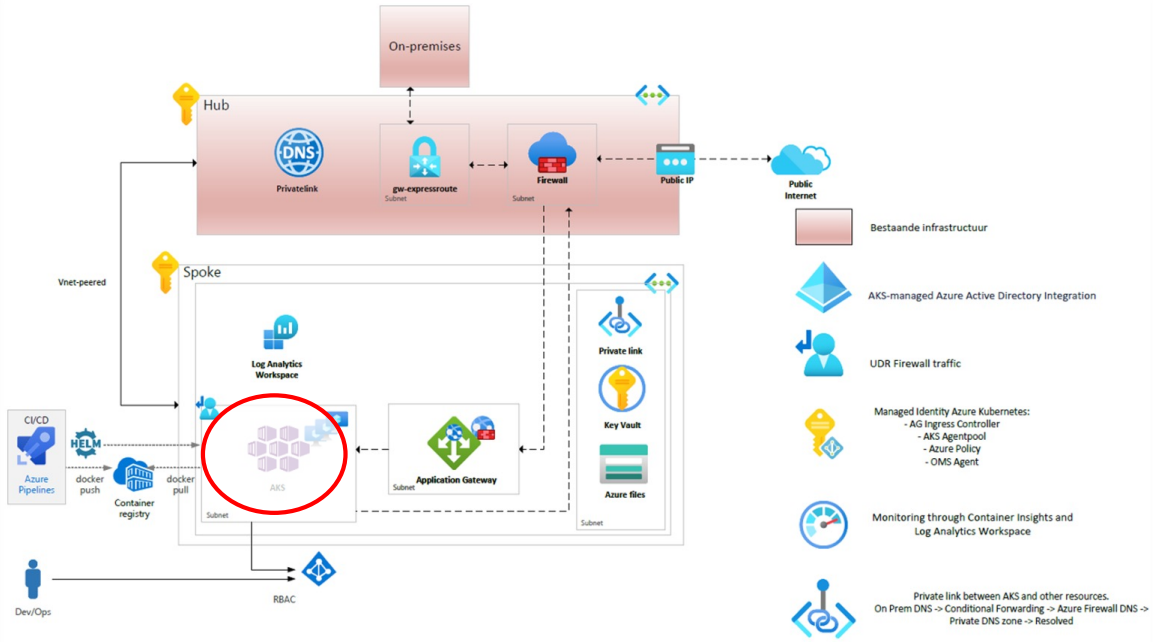



Plan of Action
Architecture
Design decisions
End-Solution
"... it ain't just enabling AKS!"

Architecture

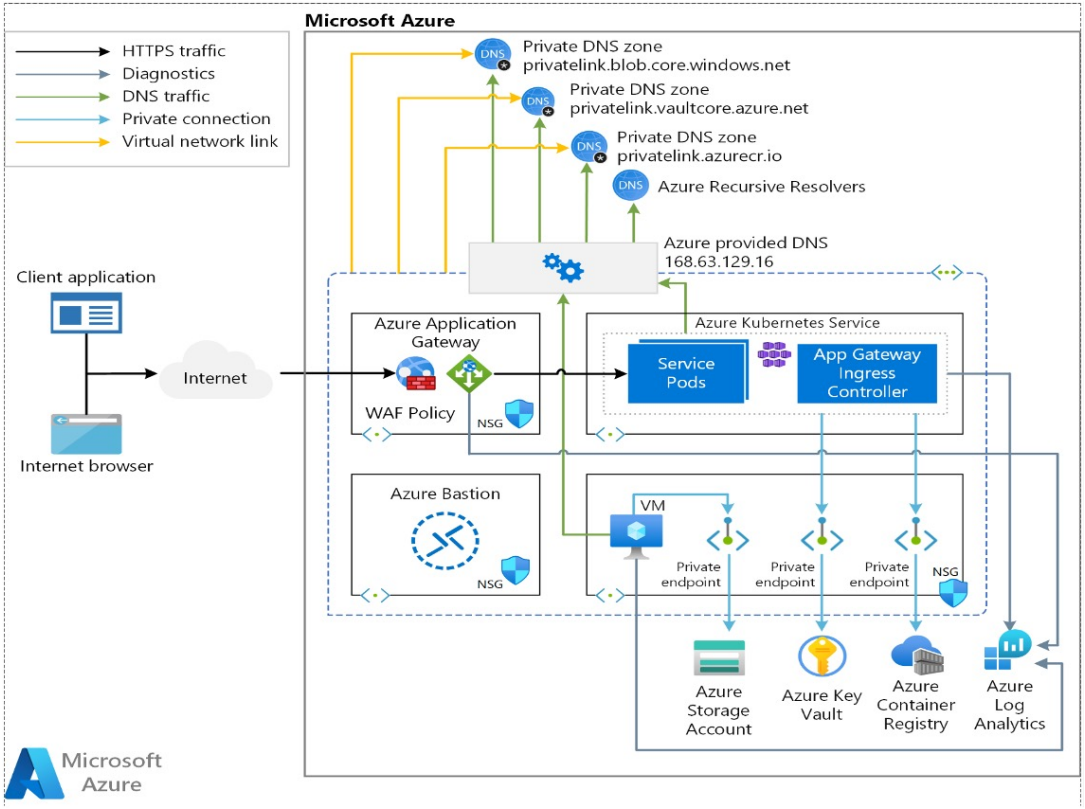
Base

- Cloud Adoption Framework;
- Well Architected Framework; [1]
- Haven compliant; [2]
- BBN2 compliant;
- Part of hub-spoke model;
- Private (AKS en ACR).



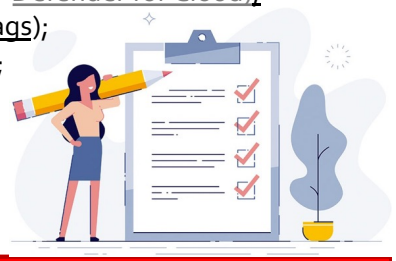
1. Well Architected Review Security
2. https://haven.commonground.nl/

Design decisions



Platform choices

-AKS Private;
- Workload Separation based on Agent pools;
-Add-ons (Azure Policy, CSI);
-Hardend OS (CIS benchmark);
-Ingress Controller (Nginx);
-Network, Application Gateway and Load Balancer;
-CNI (Azure CNI);
-Registry (ACR);
-Persistent storage (CSI driver);
-Secret management (Azure KeyVault);
-Logging (LogAnalytics);
-Monitoring (ContainerInsight);
-Authentication and authorization (AzureAD);
-Patching (KURED) and ClusterAutoScale enabled;
-SLA (Standard);
-Container Security (**Aqua >>** Defender for Cloud);
-Cost management (Azure Tags);
- Azure FW (AKS service tag);
- Workload Identity.



Design decisions

ACR;

* Container image build via VM Scaleset;
* MI ACR-Push permissions on VM Scaleset;
* MI ACR-Pull permissions on Shared AKS;
* Baseline ACR;
* Azure policies.

AKS;

* CIS Kubernetes benchmark;
* CIS Ubuntu;
* Azure Security Benchmark v3;
* Microsoft Cloud Security Benchmark;
* Baseline AKS;
* Azure Policies.

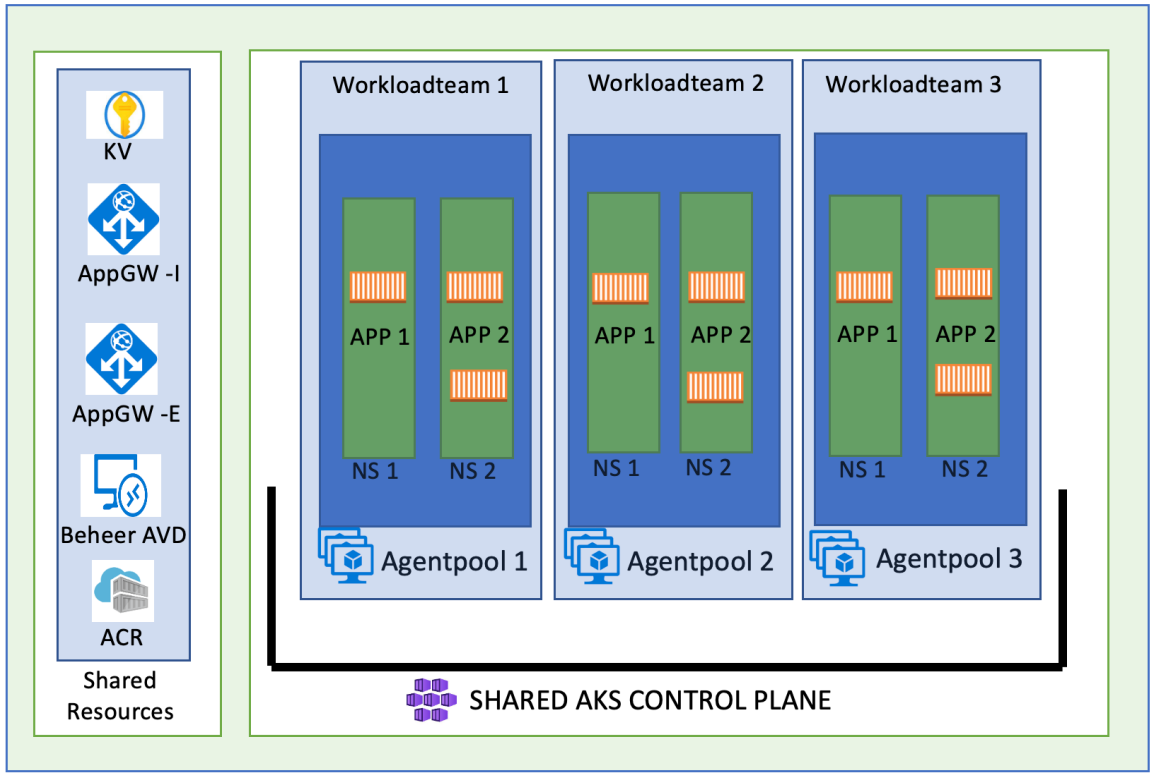
Both;

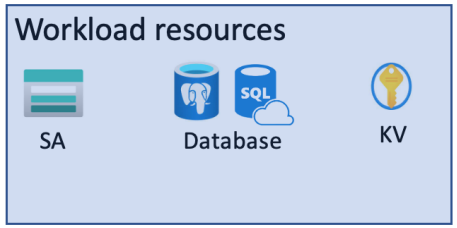
* Compliancy: BIO, SCF, SMCF, DPIA, NORA, ISO/IEC 27001:2013

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

End Solution



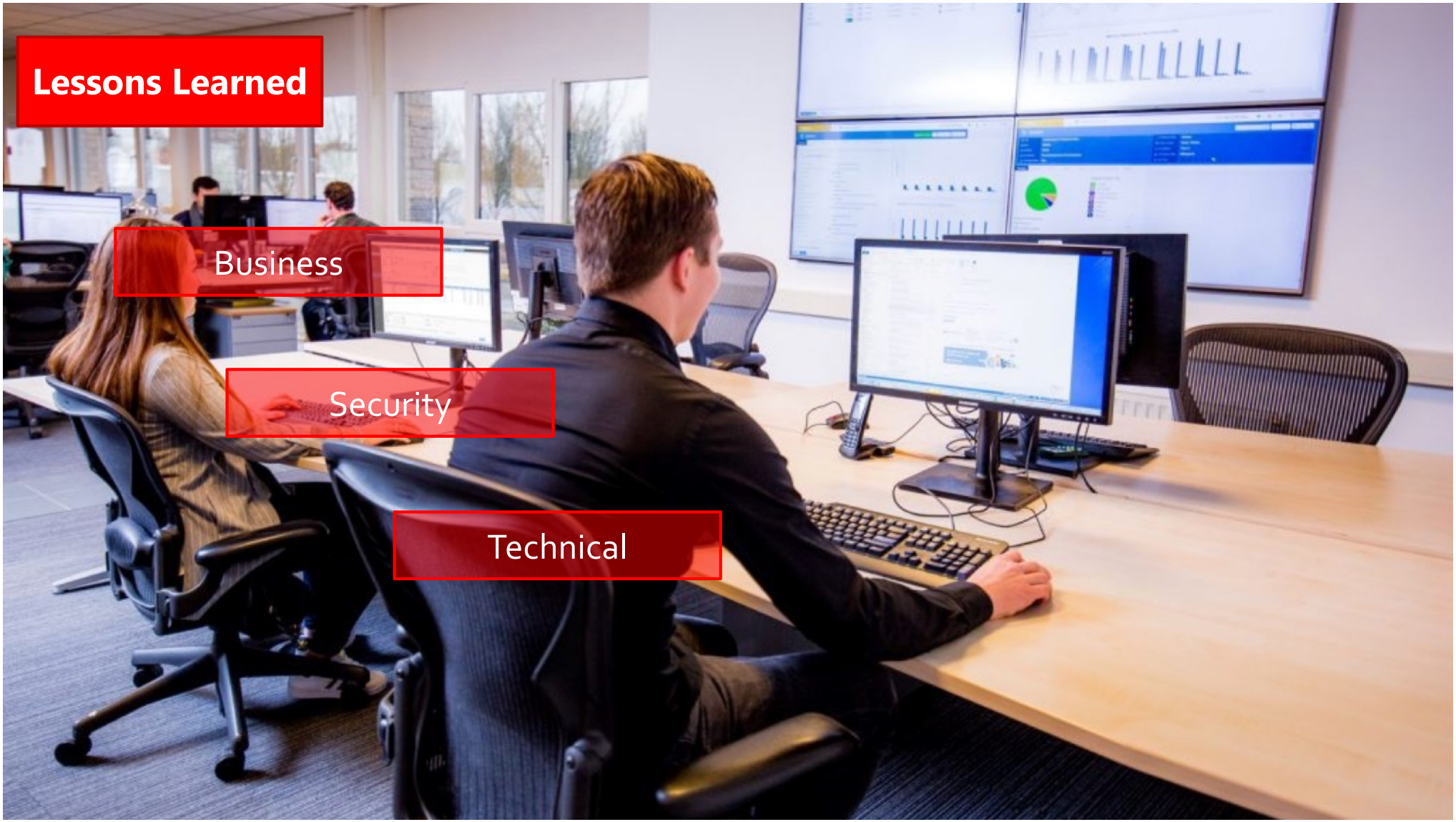


Provisioning

- AKS agent pool with custom VM size;
- Namespace and service account;
- Traffic flows;
- Listeners Application Gateways (AppGW) & Certificate ingestion;
- Workload resource connections;

Workload team

- Azure DevOps project;
- Kubernetes Service Connection;
- Private Agent connection to AKS;
- Pipeline for deployment.


Lessons Learned
Business
Security
Technical

Lessons learned

Business

- Feature Expectations;
- Costs Insights;
- DevOps Way of Work;
- Thinking "Cloud Native".

Security

- Lots of frameworks;
- Lots of Best Practices;
- Continuous renewal.

Lessons learned

Technical

- Default not Multi-Tenant;
- Private = Time consuming;
- Segmentation via AgentPools;
- Automate everything;
- Kubernetes is difficult.

Roadmap AKS: https://github.com/Azure/AKS/projects/1
Roadmap ACR: https://github.com/Azure/acr/projects/1
AKS blogs: https://dinantpaardenkooper.nl/posts/


Takeways
Understand environment
Use agentpool to divide workloads
Private AKS introduces complexity
Always Innovate
K8S Multi-tenancy is not a given
Reduce cost by sharing
Automate as much as possible