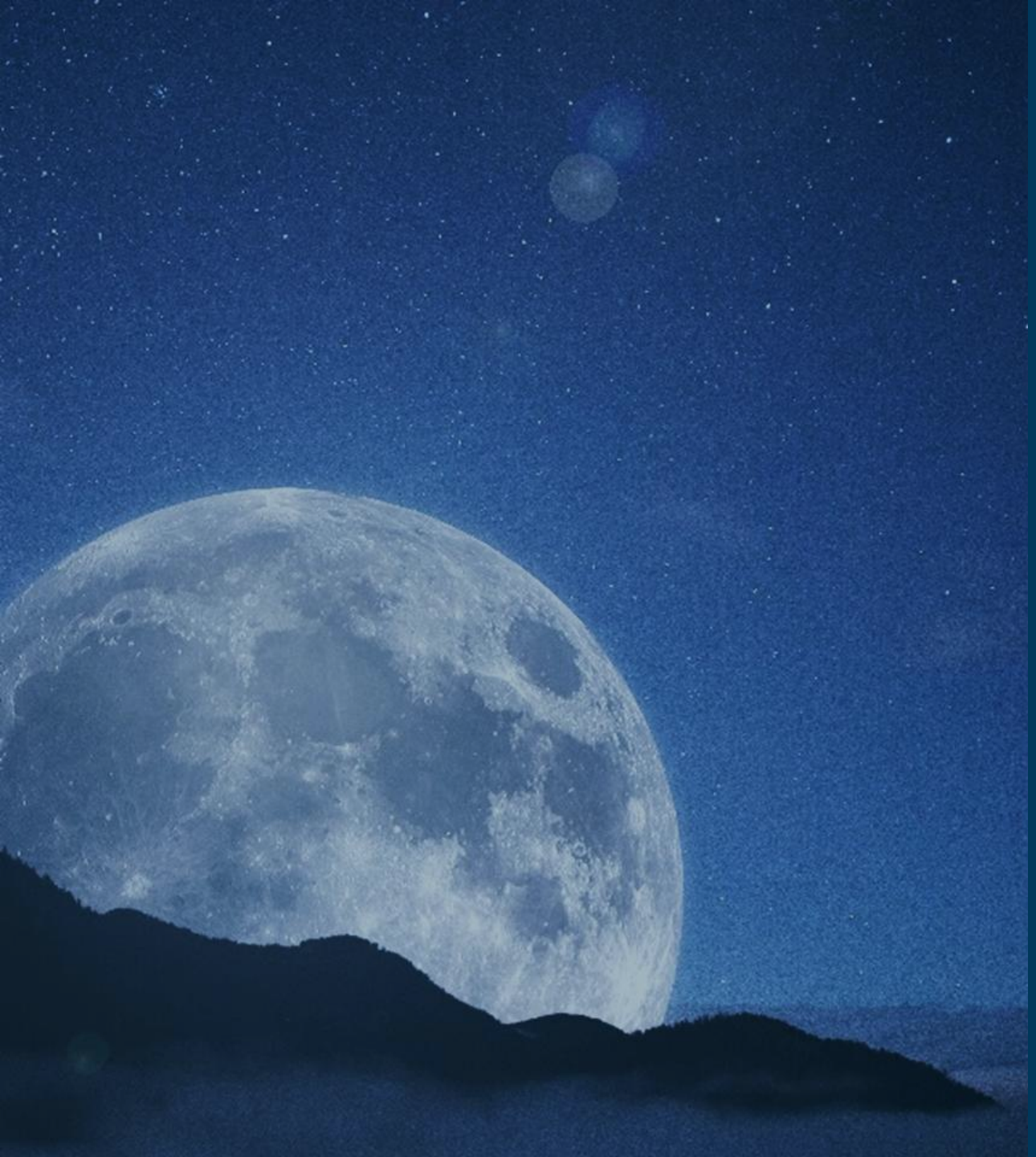
LUNARIS
Digital Solutions

SECURITY BY DESIGN IN AZURE

Sicherheit als integraler Bestandteil

Florian Bader



MIT WEM HABT IHR ES ZU TUN?

Florian Bader

Solution Architect | CTO

Florian.Bader@lunaris.digital

https://lunaris.digital
https://github.com/florianbader

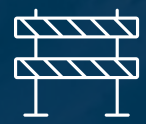
LUNARIS
Digital Solutions

KEY TAKE AWAYS

Wie bewerte ich Sicherheit in der Cloud?

Wie wird Sicherheit zum Standard?

Wie kann ich kontinuierlich auf Sicherheit prüfen?



Wie vermeide ich ernsthafte Probleme?

WARUM SECURITY BY DEFAULT

Der schnellste Weg sollte auch der Sicherste sein

Cloud ermöglicht schnelle Entwicklung, aber auch schnelle Fehlkonfiguration

Meiste Sicherheitsvorfälle entstehen durch:

- Überpriviligierte Konten
- Hartkodierte oder geleakte Secrets
- Fehlkonfiguration oder vergessene Einstellungen

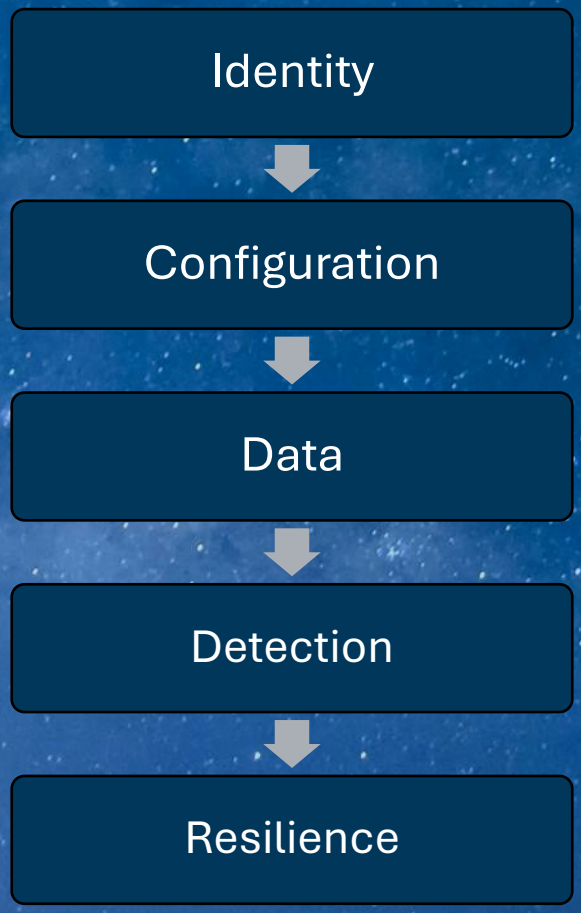
Security by Default bedeutet: Sichere Einstellungen sind automatisch der Standard und bilden die Grundlage

- Azure bietet starke Sicherheitsbausteine
- Teams müssen diese konsequent im Entwicklungsprozess verankern

Warum Security by Default?

- Standards (ISO27001), Regulatorien (CRA, NIS2, GDPR)

DIE FÜNF BAUSTEINE EINES SICHEREN AZURE-SYSTEMS



OWASP TOP 10

A01:2025 - Broken Access Control

- RBAC, Admin Logins, CORS, Elevation of Privilege

A02:2025 - Security Misconfiguration

- Default Credentials, Log Configuration, Missing Security Defaults

A03:2025 - Software Supply Chain Failures

- Vulnerabilities in Containers

A04:2025 - Cryptographic Failures

- Fehlende Verschlüsselung, Self Signed Certificates

A05:2025 – Injection

- SQL, XML, Command Line, LDAP

A06:2025 - Insecure Design

- Threat Modeling

A07:2025 - Authentication Failures

- Weak Admin Password, Missing MFA

A08:2025 - Software or Data Integrity Failures

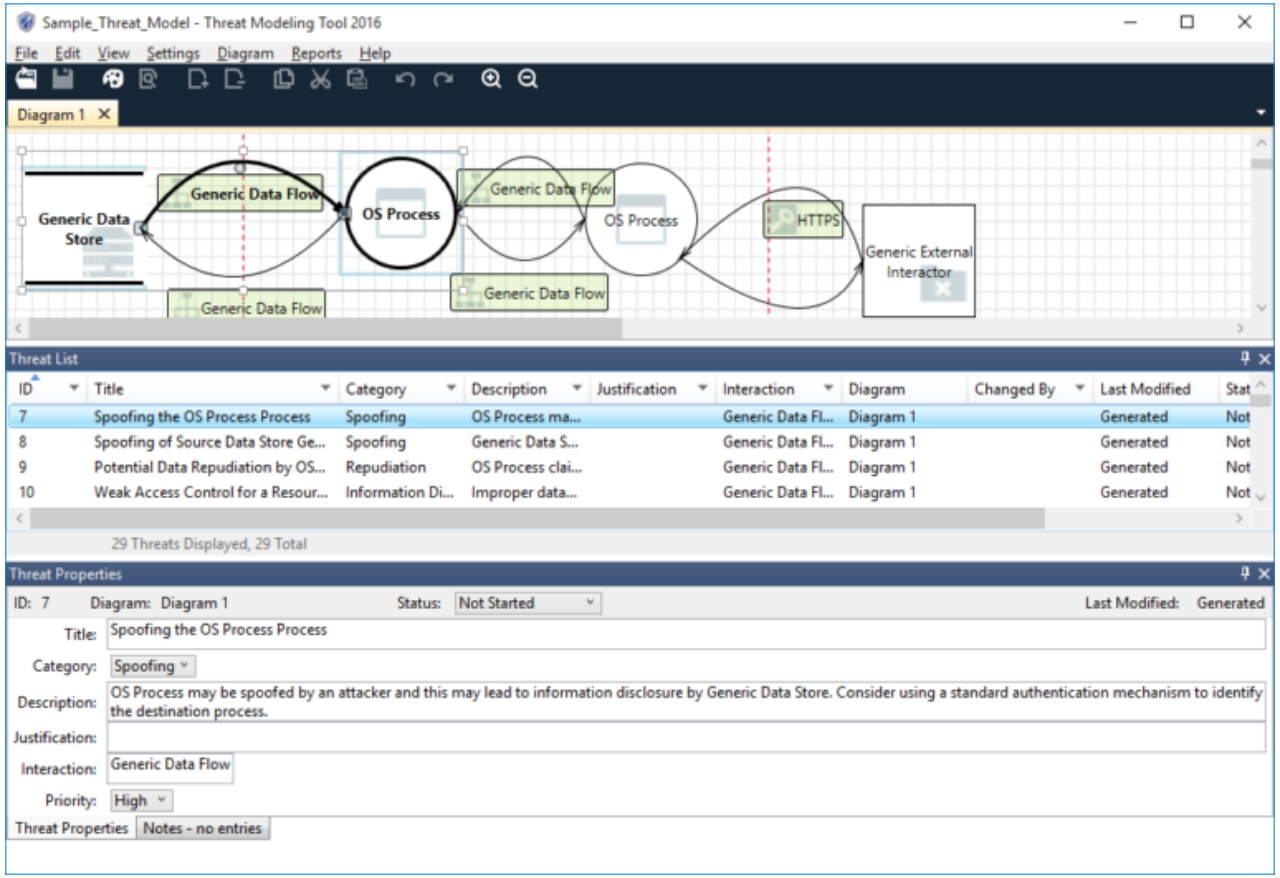
- Trusted Repos für IaC, CI/CD Environment Separation

A09:2025 - Logging & Alerting Failures

- Sensitive Daten in Logs, Keine Logs für verdächtige Aktivitäten

A10:2025 - Mishandling of Exceptional Conditions

- Sensitive Daten in Exceptions, Denial of Service



Microsoft Threat Modeling Tool overview - Azure | Microsoft Learn

THREAT MODELLING

Identifizieren, wo Schwachstellen sind

Threat Modeling

PlantUML/Markdown, Microsoft Threat Modeling Tool, OWASP Threat Dragon, IriusRisk

Non-Functional Requirements

AuthN /AuthZ, Verschlüsselung, Logging/Monitoring, Datenschutz, Verfügbarkeit, ...

STRIDE

Spoofing (Identitätsbetrug)

Tampering (Manipulation)

Repudiation (Abstreitbarkeit)

Information Disclosure (Informations-Leaks)

Denial of Service (Unverfügbarkeit)

Elevation of Priviledge (Rechteausweitung)

Alternativen: PASTA, AttackTrees, FMEA

FOUNDATION

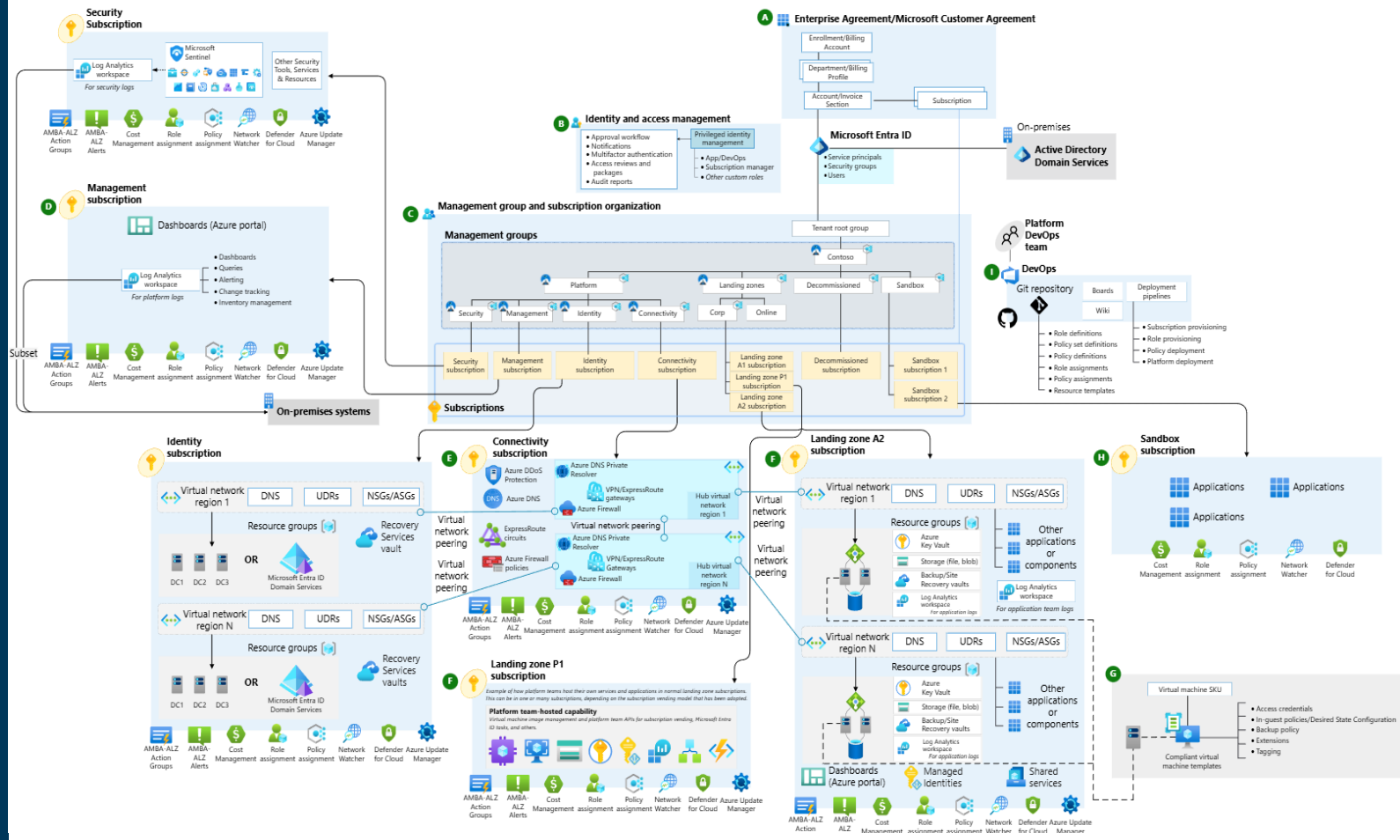
Wie strukturiere ich meinen Azure Tenant sicher?

Eine gute Tenant-Struktur ist die Basis für Sicherheit & Governance

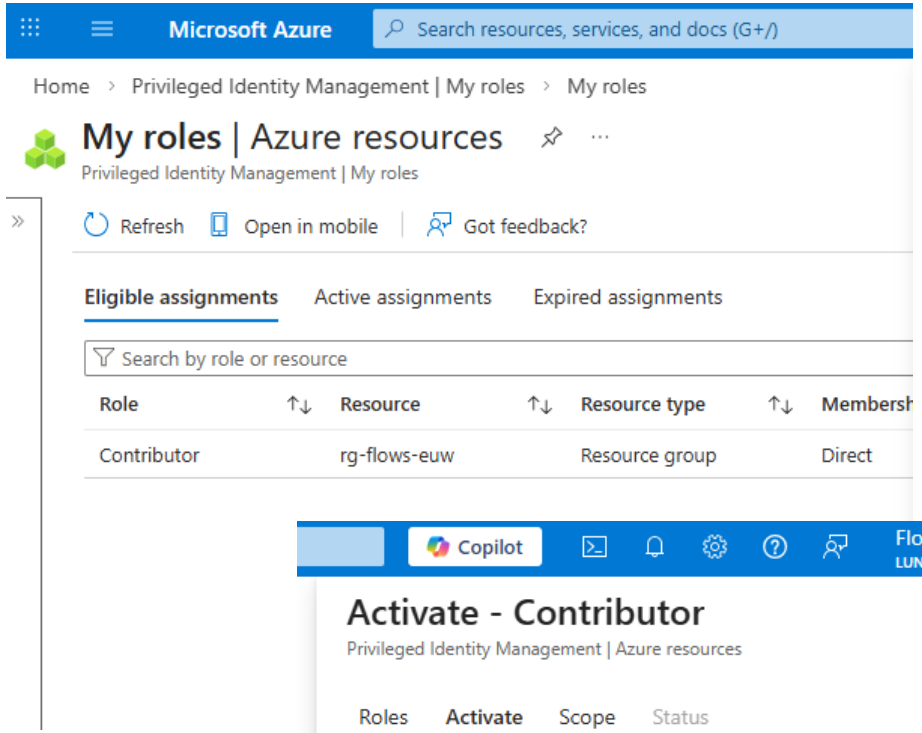
- Management Groups, Subscriptions, Resource Groups
- Subscriptions trennen Verantwortlichkeiten und Risiken
- Resource Groups strukturieren Workloads

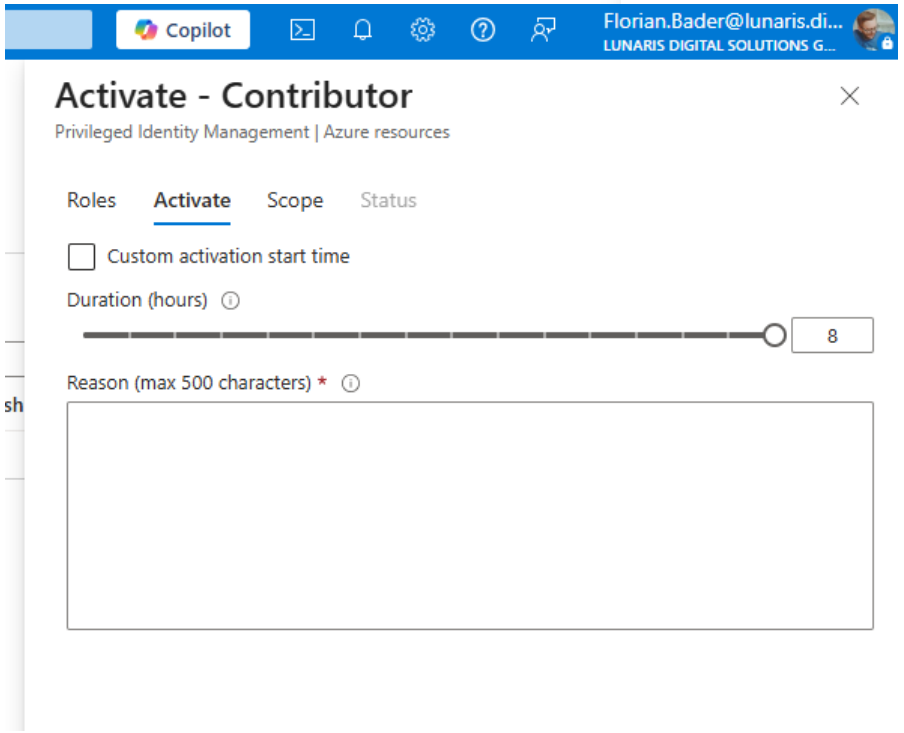
Klein starten, aber skalierbar denken

- Je nach Unternehmensgröße und Anzahl der Teams
- Je nach Billing (Enterprise Agremeent, Pay as you go)



What is an Azure landing zone? - Cloud Adoption Framework | Microsoft Learn





IDENTITY & ACCESS

Wer bekommt welche Berechtigung?

Entra für Berechtigung

- Gruppen definieren, statt Berechtigung für Einzelpersonen
- Keine Admin Logins oder Access Keys nutzen
- Custom Roles statt Contributor by Default

Privilege Identity Management (PIM)

- User können sich zusätzliche Rechte beschaffen nach Freigabe
- Nicht jeder hat direkt Zugriff auf Production

Adminzugang

- Plattform und App Admins trennen
- Conditional Access nutzen
- Notfallzugang klären (Breaking Glass Accounts)

ENTWICKLUNG

Passwortloser Zugriff als Standard

Secrets gehören nicht in den Code

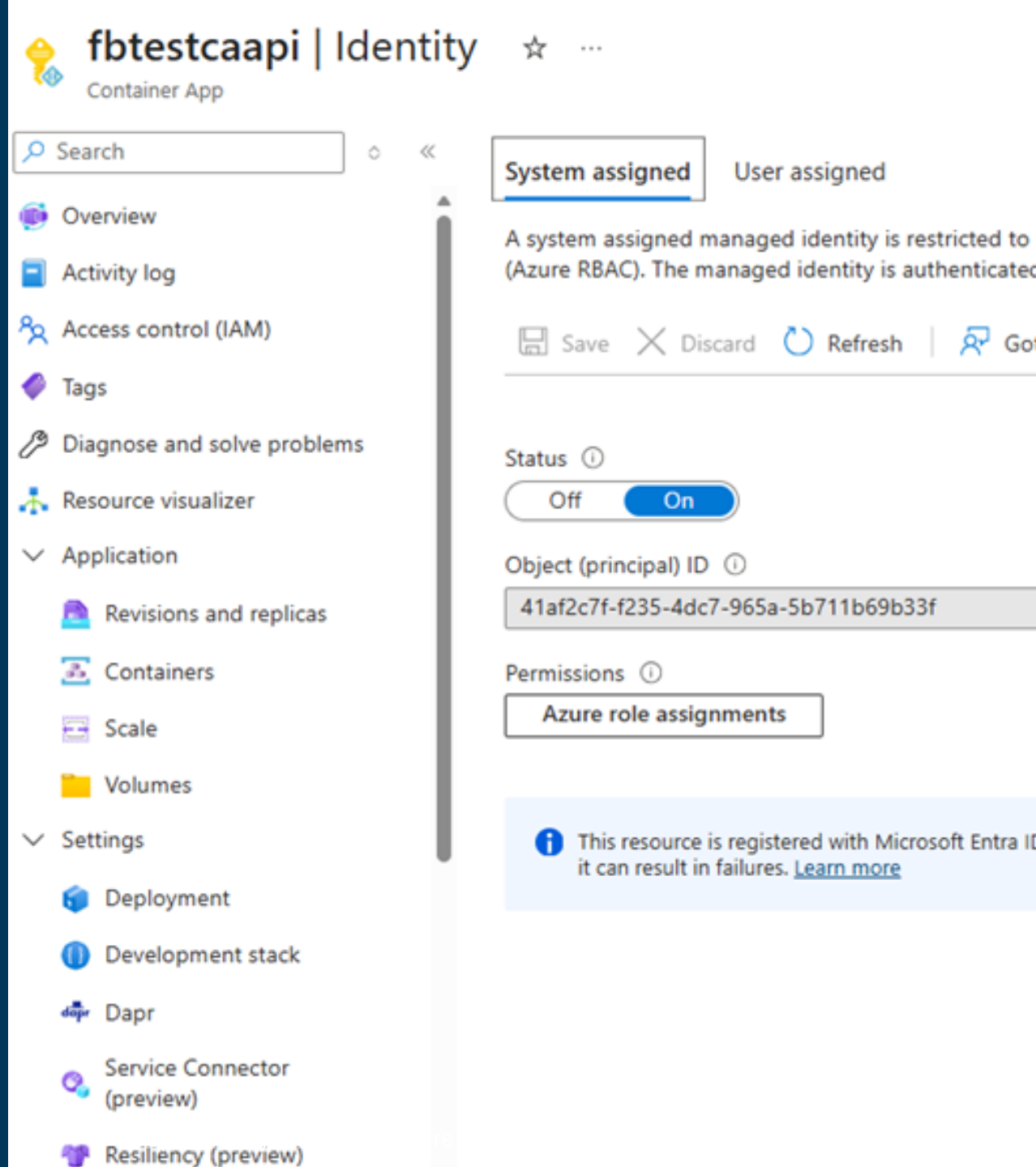
- Key Vault schützt Secrets, die es doch noch braucht

Managed Identities und Entra Authentifizierung nutzen

- Secretless mit Entra basierter Authentifizierung
- Client Credentials oder mTLS (Zertifikate) statt API Tokens

Service Principle und Workload Identities für Pipelines

- Ein Service Principal pro App pro Environment für saubere Isolation



```
Template: C:\dev\openspaceplanner\projects\Common\src\Infrastructure\containerAppEnvironment.bicep
        Root Template: C:\dev\openspaceplanner\projects\OpenSpacePlanner\src\Infrastructure\main.bicep
        AZR-000363: Disable public access
                Severity: High
                Recommendation: Consider disabling public network access.
                More information: https://azure.github.io/PSRule.Rules.Azure/en/rules/Azure.ContainerAp
p.PublicAccess/
                Result: Failed
                Line: 23


Template: C:\dev\openspaceplanner\projects\Common\src\Infrastructure\containerRegistry.bicep
        Root Template: C:\dev\openspaceplanner\projects\OpenSpacePlanner\src\Infrastructure\main.bicep
        AZR-000005: Disable ACR admin user
                Severity: High
                Recommendation: Consider disabling the admin user account and only use identity-based a
uthentication for registry operations.
                More information: https://azure.github.io/PSRule.Rules.Azure/en/rules/Azure.ACR.AdminUs
er/
                Result: Failed
                Line: 30


Template: C:\dev\openspaceplanner\projects\Common\src\Infrastructure\sqlServer.bicep
        Root Template: C:\dev\openspaceplanner\projects\OpenSpacePlanner\src\Infrastructure\main.bicep
        AZR-000186: Use Advanced Threat Protection
                Severity: High
                Recommendation: Consider enabling Advanced Data Security and configuring Microsoft Defe
nder for SQL logical servers.
                More information: https://azure.github.io/PSRule.Rules.Azure/en/rules/Azure.SQL.Defende
rCloud/
                Result: Failed
                Line: 40
        AZR-000187: Enable auditing for Azure SQL DB server
                Severity: High
                Recommendation: Consider enabling auditing for each SQL Database logical server and rev
iew reports on a regular basis.
                More information: https://azure.github.io/PSRule.Rules.Azure/en/rules/Azure.SQL.Auditin
g/
                Result: Failed
                Line: 40
        Rules passed: 0
```

TESTING & VALIDATION

Sicherheit während des Deployment prüfen

Infrastructure as Code für Security Defaults

- Don’t reinvent the wheel: Azure Verified Modules
- Zentral für alle Teams bereitstellen

SAST für Infrastructure as Code

- Checkov, Trivy, Azure Template Analyzer, Bandit, ...

Pre-Deployment Approval

- Wer darf Änderungen freigeben?
- Was wird gerade deployed? (Dependencies, Secrets, ...)

Azure Policy

- Security Defaults erzwingen
- Alerts bei Nichteinhaltung von Standards

BETRIEB

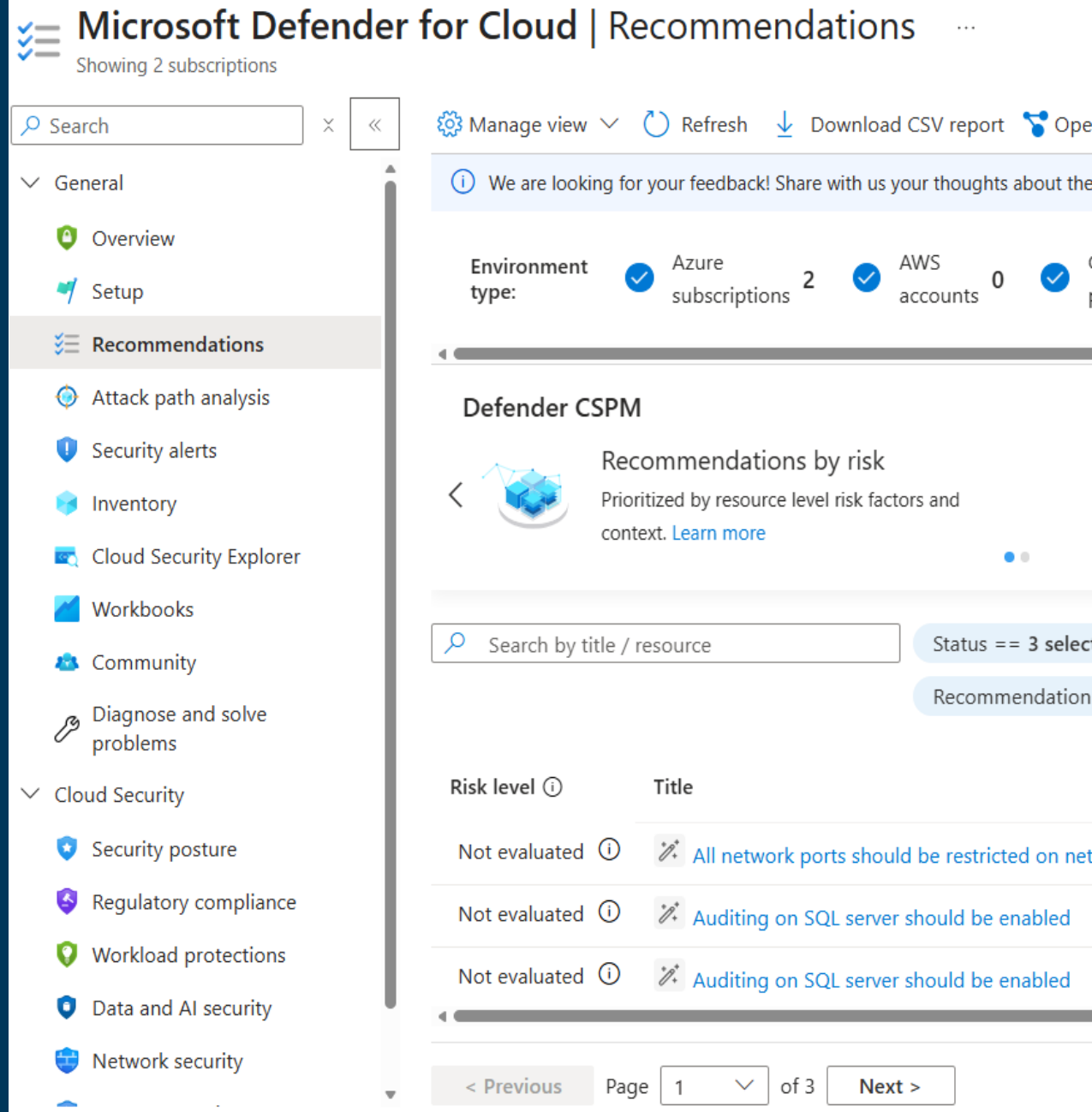
Sicherheit nach dem Deployment kontinuierlich prüfen

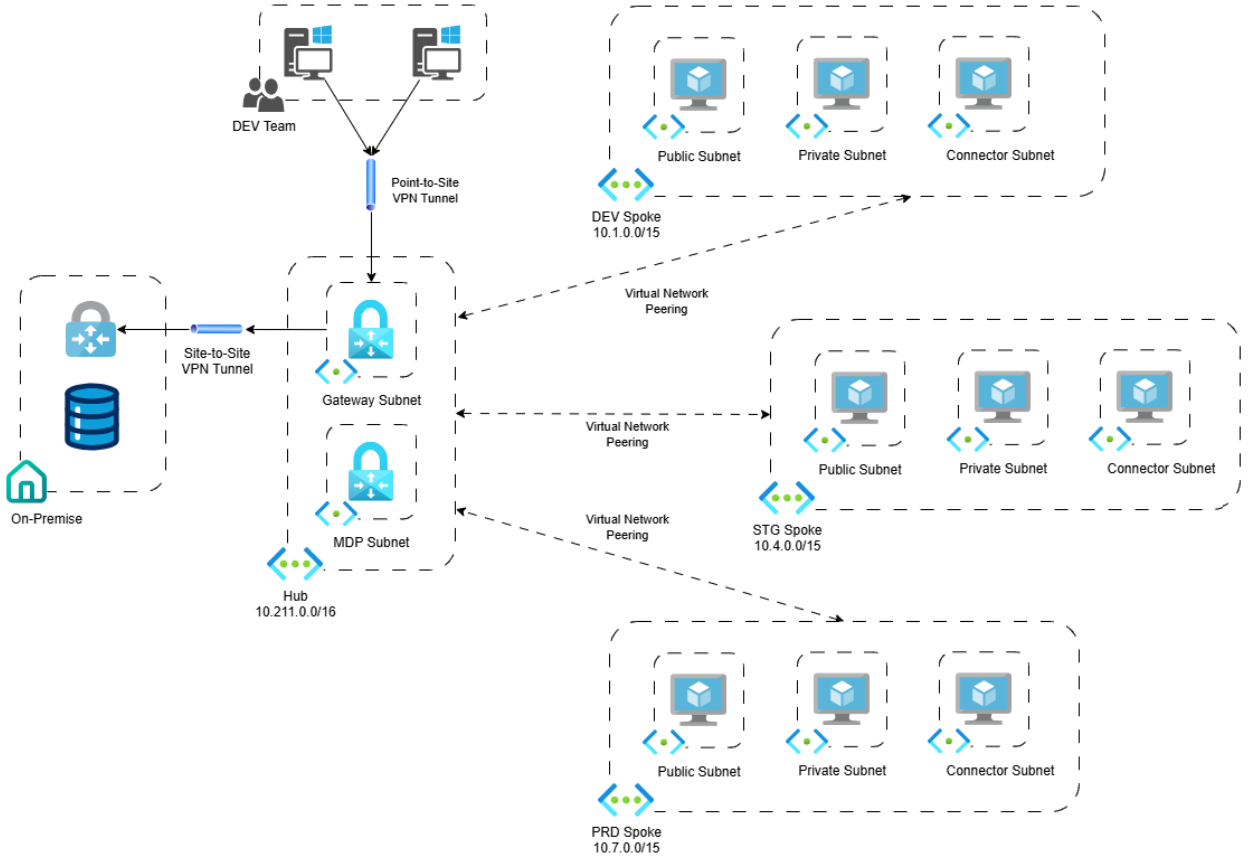
Monitoring & Threat Protection

- Microsoft Defender for Cloud
- Microsoft Sentinel

Compliance & Governance

- Azure Log Analytics
- Activity Logs
- Diagnostic Logs
- Azure Policy





NETWORK SECURITY

Für alles, was nicht Öffentlich sein darf

Virtual Networks

- Segmentierung nach Workload / Umgebung
- Klare Subnet-Trennung (App, Data, Shared Services)
- Grundlage für isolierte, private Kommunikation

Network Security Groups (NSG)

- Layer-4-Firewall pro Subnet oder NIC
- Least Privilege Ports: Nur explizit erlaubte Flows
- Logging & Flow Logs für Analyse

Private Endpoints

- Private IPs für Public Services

VPN Gateway

- Sicherer Dev-Zugriff auf interne Ressourcen
- Point-to-Site mit Zertifikat oder Entra ID Auth

RESILIENCY

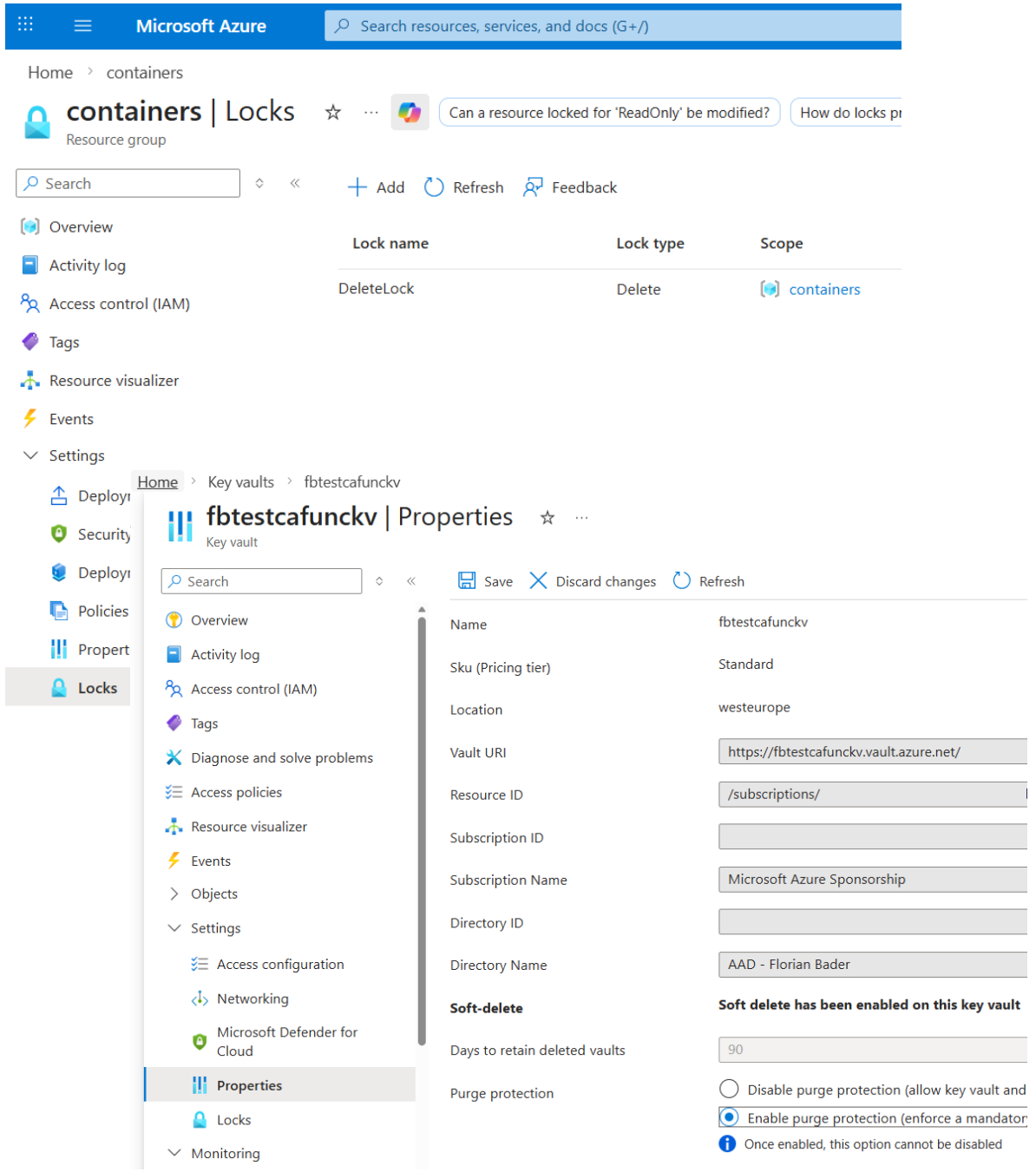
Schutz vor fälschlicher Modifikation

Anti Tampering

- Resource Locks (Delete, Read-Only)
- Soft-Delete & Purge Protection
- Azure Policy
- Immutable Storage

Backup & Restore

- Disaster Recovery Strategy
- Azure Backup
- Backup & Restore in Resourcen (Azure SQL, Storage, ...)

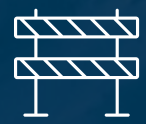


RECAP

Sicherheit entsteht im Design

Infrastructure as Code hilft Sicherheitsstandards zu etablieren

Kontinuerliches und automatisiertes Prüfen auf Sicherheit ist notwendig



Resilienz ist Teil der Sicherheit


LUNARIS
Digital Solutions