Azure Bicep – from manual to magical

- Bernhard Flür

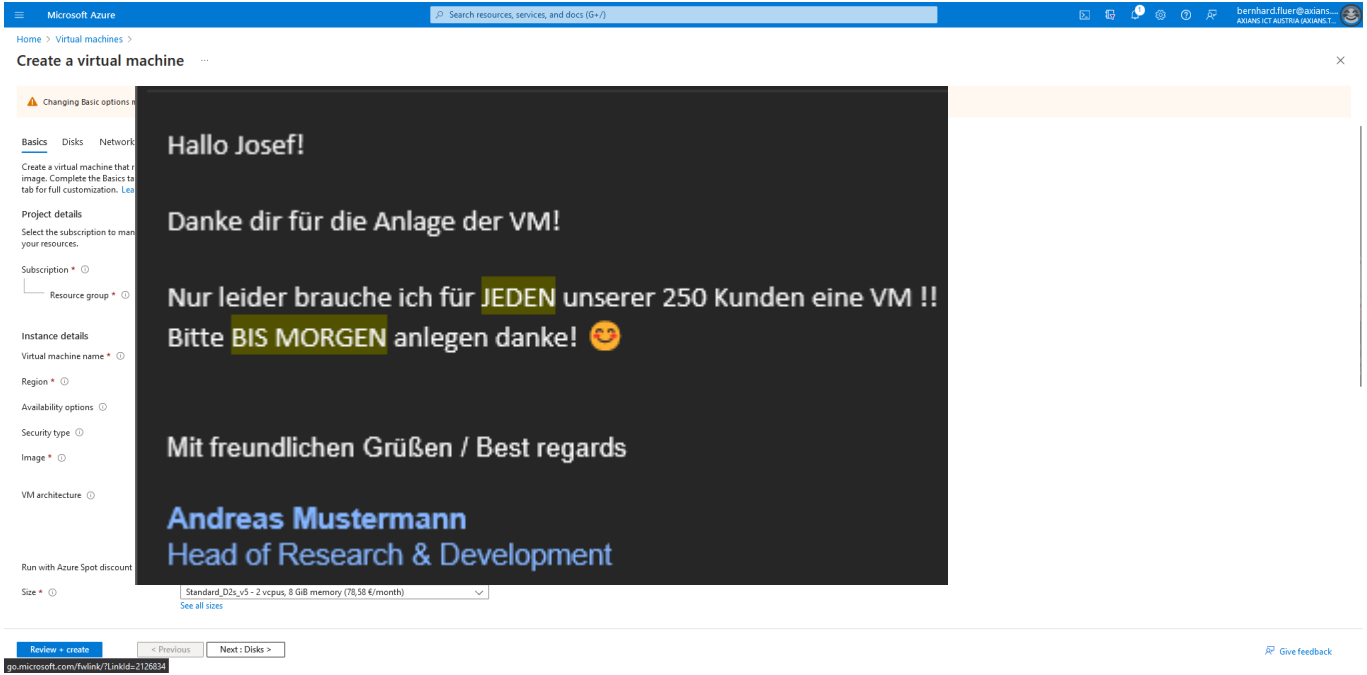
Experts Live Austria

Bernhard Flür

- Innsbruck, Austria
- Seit 2022 bei Axians (Cloud Transformation Services)
- Schwerpunkte: Azure, DevOps, IaC

Deploying Azure

(in den meisten Fällen)



Nachteile

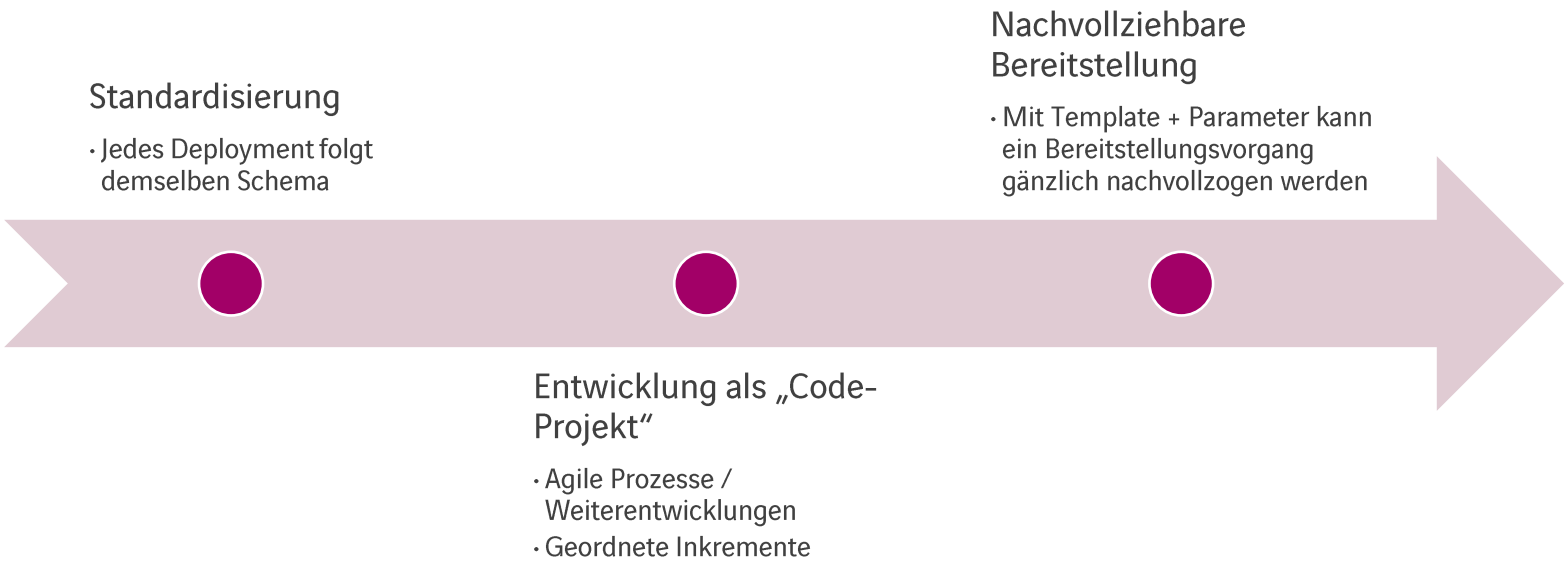
„Manueller“ Cloud Betrieb

- Aufwand für wiederholende Tasks
- Dokumentation
- Neue Ressourcen sind nicht konsistent

axians

Infrastructure as Code

Vorteile


Standardisierung
• Jedes Deployment folgt demselben Schema
Entwicklung als „Code-Projekt“
• Agile Prozesse / Weiterentwicklungen
• Geordnete Inkremente
Nachvollziehbare Bereitstellung
• Mit Template + Parameter kann ein Bereitstellungsvorgang gänzlich nachvollzogen werden

Meist genutzte IaC – Sprachen in Azure

Azure Bicep

ARM

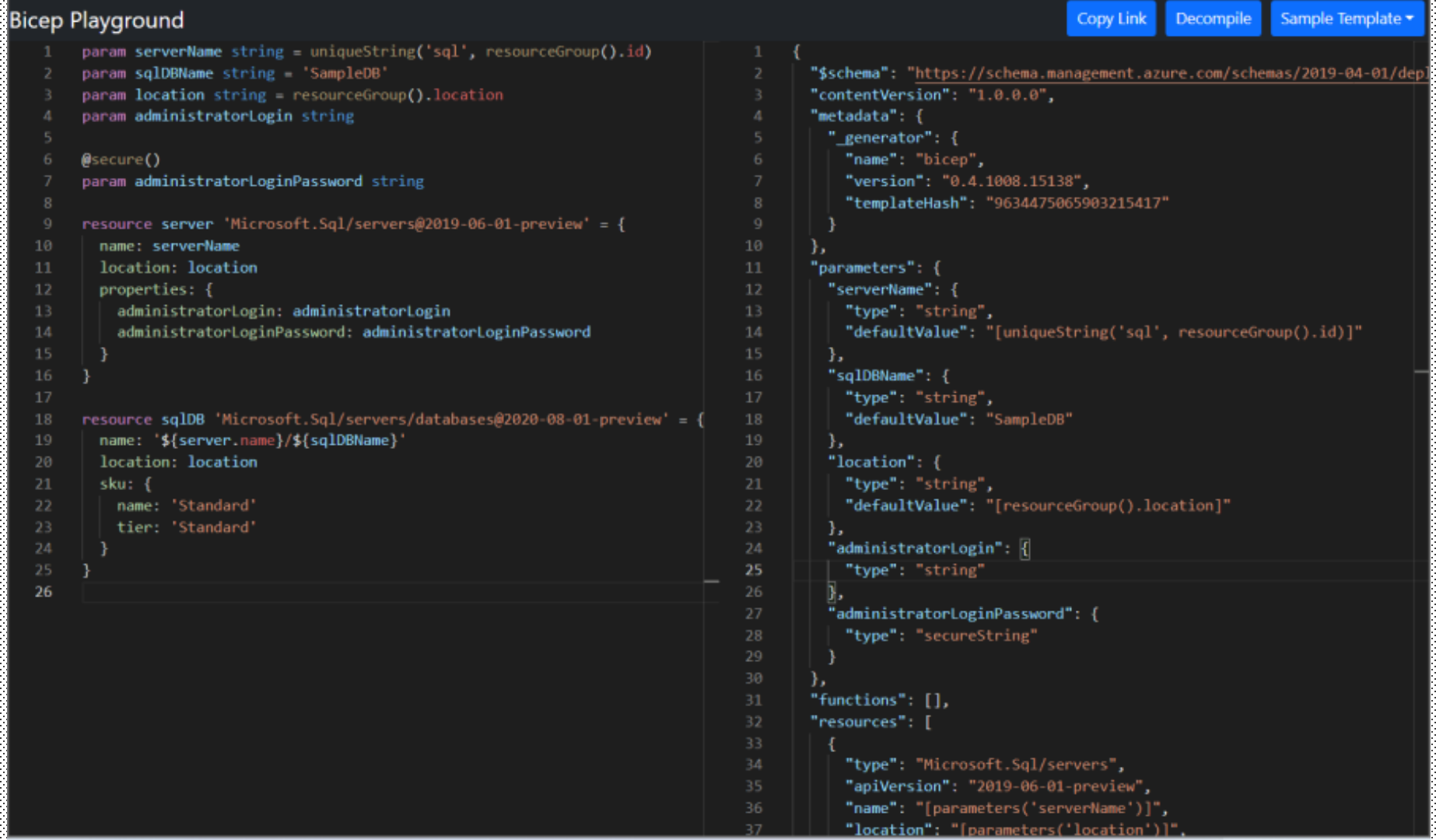
HashiCorp
Terraform

Azure Bicep

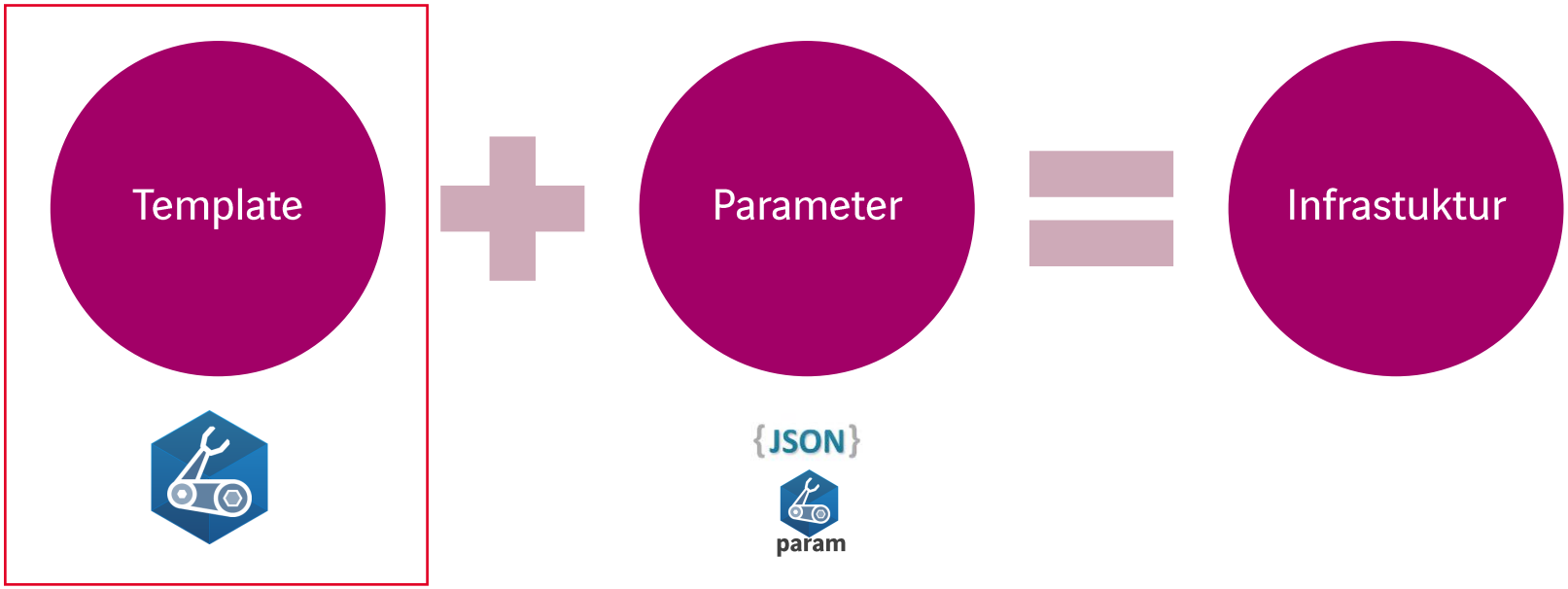
The evolution of ARM

- Since 2020
- Motto: „Make ARM easier to read“
- Deklarative DSL*
  - = Beschreibung des gewünschten Endzustands

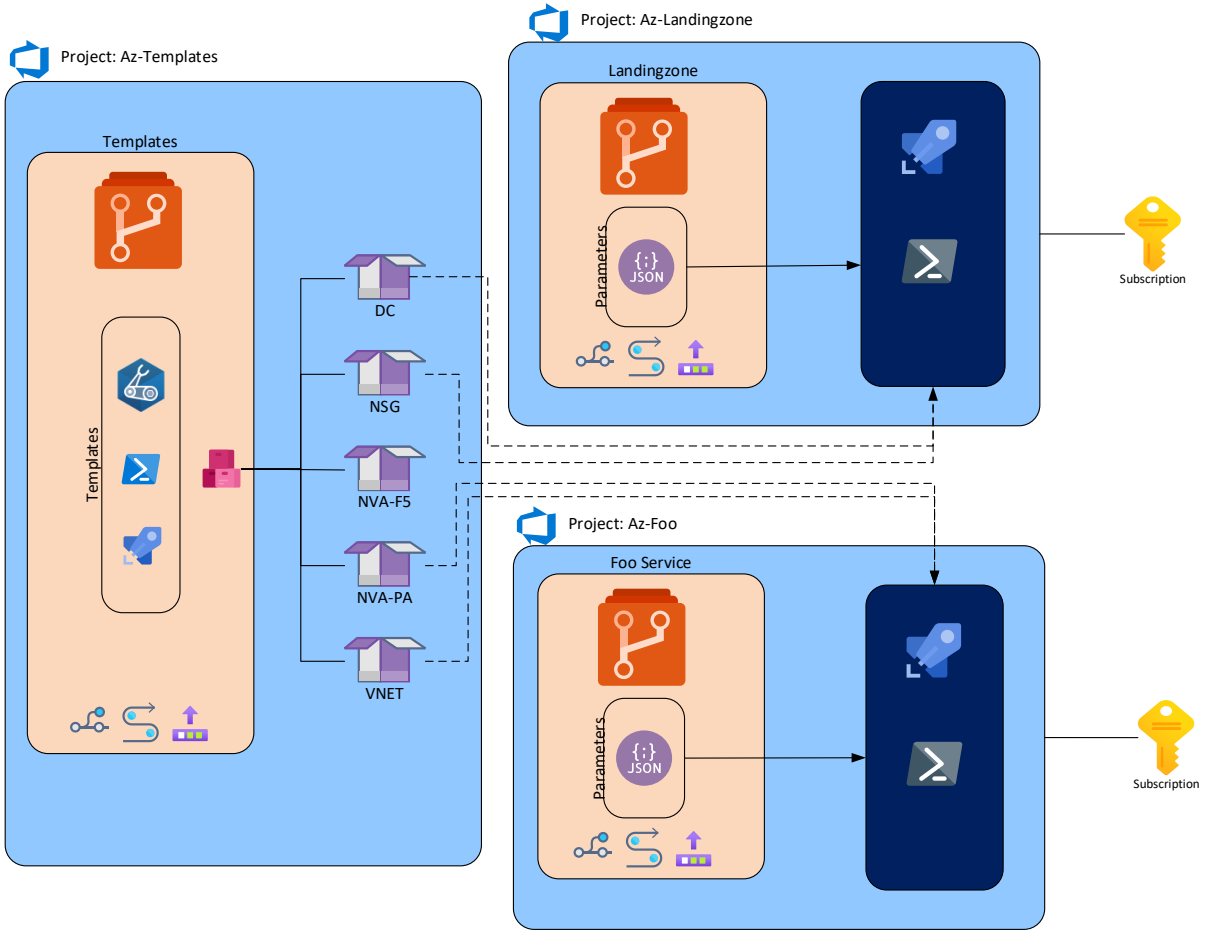
* = Domain Specific Language

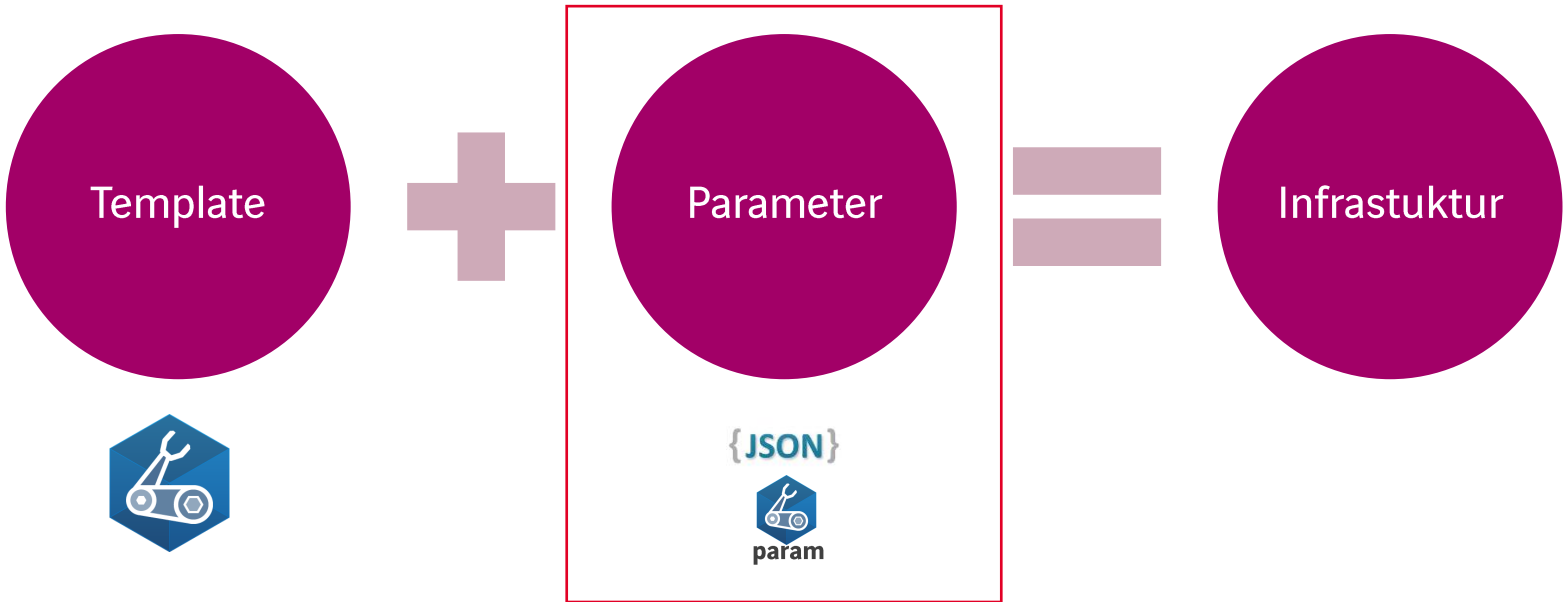

Bicep Playground
Copy Link
Decompile
Sample Template
param serverName string = uniqueString('sql', resourceGroup().id)
param sqlDBName string = 'SampleDB'
param location string = resourceGroup().location
param administratorLogin string
@secure()
param administratorLoginPassword string
resource server 'Microsoft.Sql/servers@2019-06-01-preview' = {
name: serverName
location: location
properties: {
administratorLogin: administratorLogin
administratorLoginPassword: administratorLoginPassword
}
}
resource sqlDB 'Microsoft.Sql/servers/databases@2020-08-01-preview' = {
name: '${server.name}/${sqlDBName}'
location: location
sku: {
name: 'Standard'
tier: 'Standard'
}
}
{
"$schema": "https://schema.management.azure.com/schemas/2019-04-01/dep
"contentVersion": "1.0.0.0",
"metadata": {
"_generator": {
"name": "bicep",
"version": "0.4.1008.15138",
"templateHash": "9634475065903215417"
}
},
"parameters": {
"serverName": {
"type": "string",
"defaultValue": "[uniqueString('sql', resourceGroup().id)]"
},
"sqlDBName": {
"type": "string",
"defaultValue": "SampleDB"
},
"location": {
"type": "string",
"defaultValue": "[resourceGroup().location]"
},
"administratorLogin": {
"type": "string"
},
"administratorLoginPassword": {
"type": "secureString"
}
},
"functions": [],
"resources": [
{
"type": "Microsoft.Sql/servers",
"apiVersion": "2019-06-01-preview",
"name": "[parameters('serverName')]",
"location": "[parameters('location')]",

Konzepte


Template
Parameter
Infrastuktur
{JSON}
param

Templates


Project: Az-Templates
Templates
Templates
DC
NSG
NVA-F5
NVA-PA
VNET
Project: Az-Landingzone
Landingzone
Parameters
JSON
Subscription
Project: Az-Foo
Foo Service
Parameters
JSON
Subscription


Template
Parameter
{JSON}
param
Infrastuktur

Unterteilung von IaC Deployments

~ meine „Persönliche“

Parameter

STATEFUL

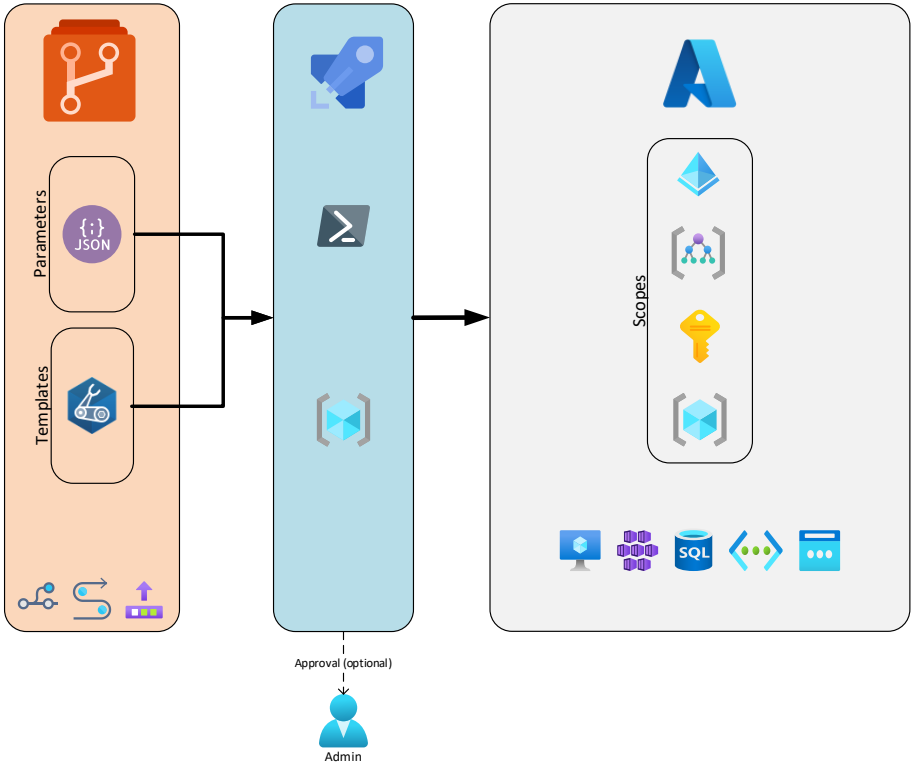
- Laufende Evolution der bereitgestellten Infrastruktur (Template/Parameter)
- Gute Nachvollziehbarkeit / Dokumentation
  - Landingzone
  - Services

STATELESS

- Einmalige Bereitstellung
- Keine Notwendigkeit eines späteren Abrufs oder Änderung der Parameter
- „Requestable Items“
  - Virtual Machine
  - Spoke vNet

axians

Parameter

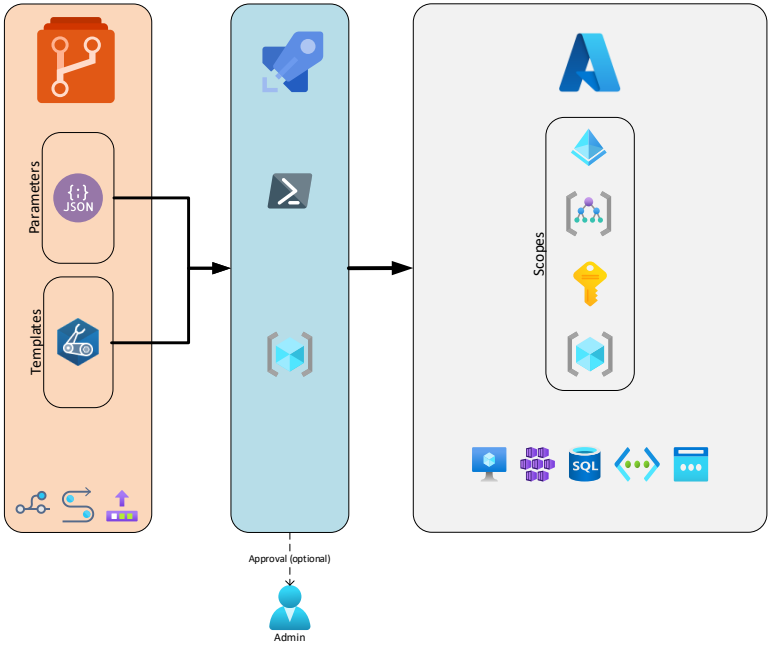


Parameter by Repository

Parameter

- Werte werden im Parameters-JSON im Repository gespeichert
- Änderungen erfolgen per Push ins Repostitory
- Tracking der Änderungen über Git-Versionierung
- 1 Parameter-File = 1 Deployment

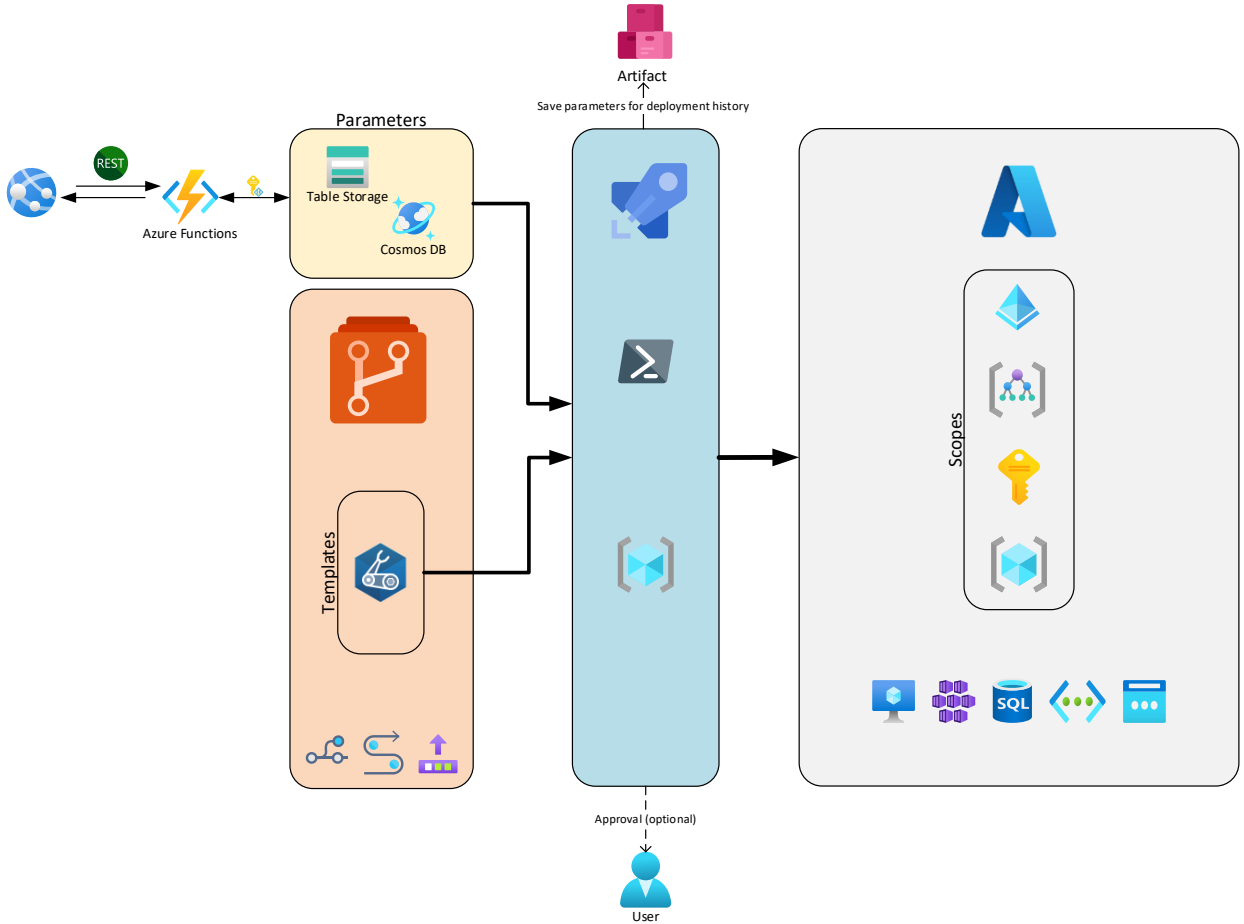
- Kein Automatismus vorhanden ☹



DATABASE-PROVIDED

axians

Parameter

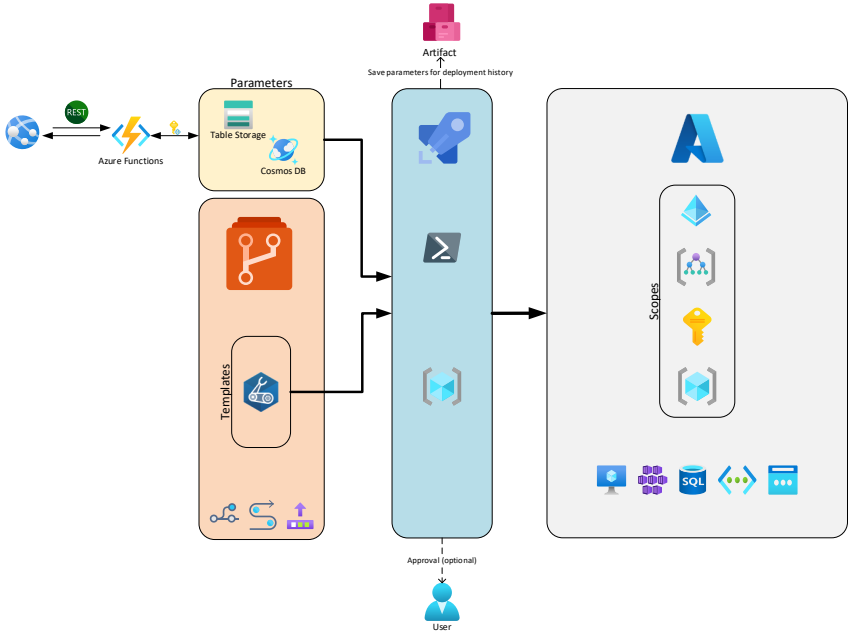


Parameters by Database

Parameter

- Werte kommen von einem Table Storage / Cosmos DB als NoSQL Database Entry
- Externes Handling erfolgt per Azure Function
- Abruf von Parameter über PowerShell Skript Task

- Automatismus und Deployment Status vorhanden
- Allerdings: keine native Rückverfolgbarkeit von Änderungen



axians

Parameter

Unterteilung von IaC Deployments

~ meine „Persönliche“

STATEFUL

- Laufende Evolution der bereitgestellten Infrastruktur (Template/Parameter)
- Gute Nachvollziehbarkeit / Dokumentation
  - Landingzone
  - Services

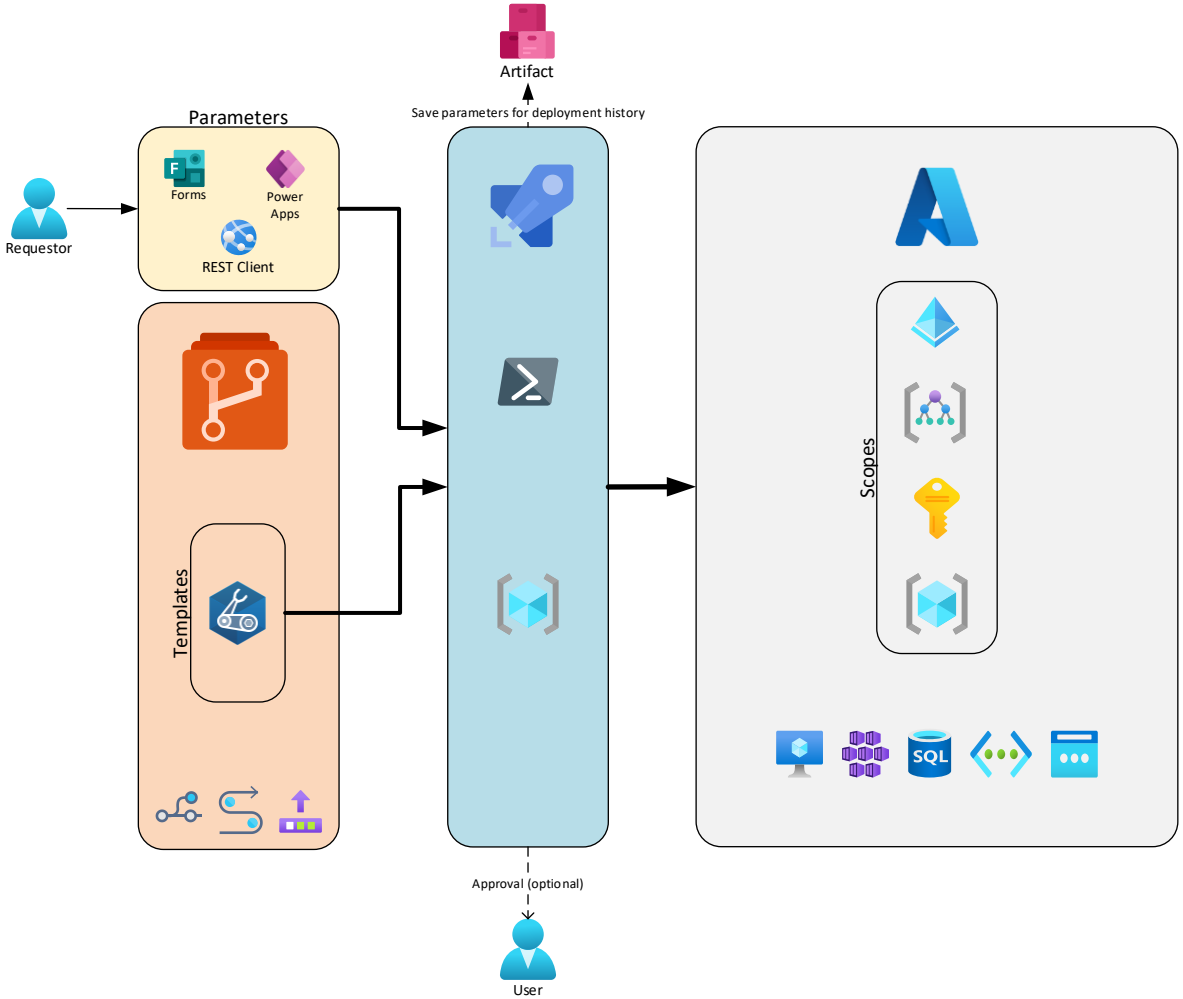
STATELESS

- Einmalige Bereitstellung
- Keine Notwendigkeit eines späteren Abrufs oder Änderung der Parameter
- „Requestable Items“
  - Virtual Machine
  - Spoke vNet

PROVIDED BY REQUEST

axians

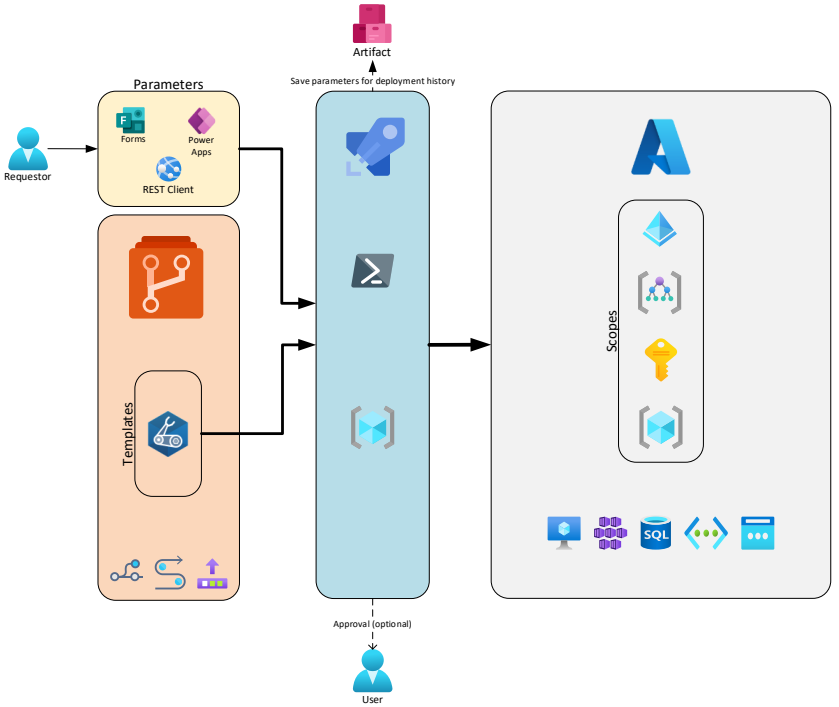
Parameter



Parameters by Request

Parameter

- Deployment wird über Power Platform oder 3rd Party API angefragt
- Übergabe aller relevanten Parameter über Eingabemaske
- Parameter werden in Artifact gespeichert
  - -> Nachvollziehbarkeit!



axians

Parameter

Zusammenfassung

Strategien zur Parameterbereitstellung

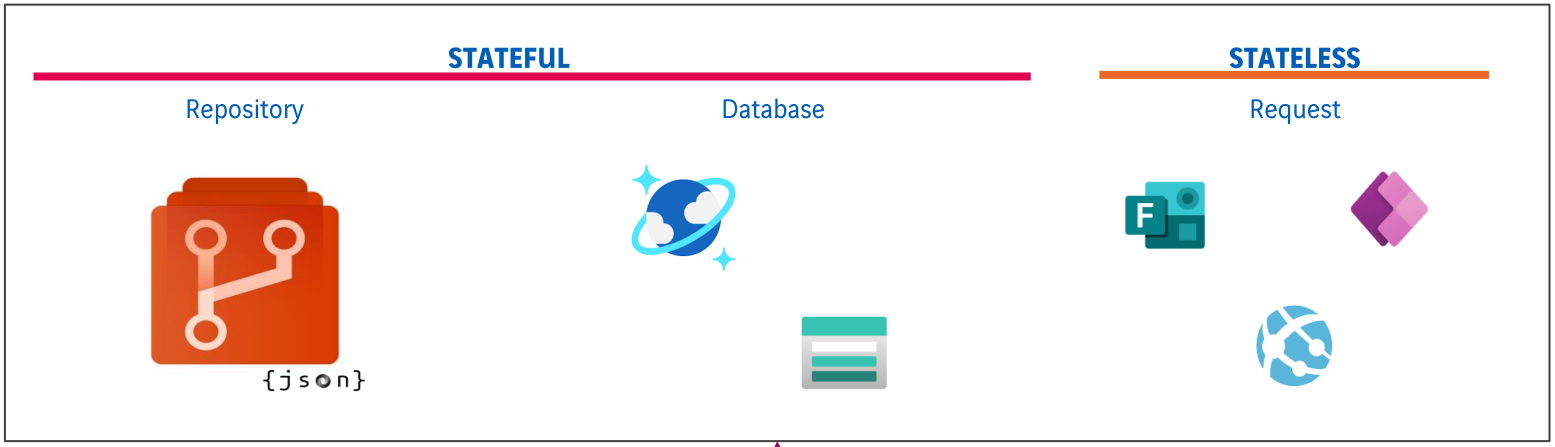
STATEFUL

Repository

Database

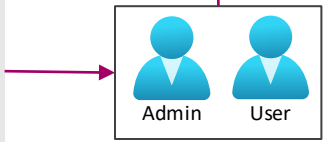
STATELESS

Request



Parameter:

- Name of VM
- Location
- Vnet/SNet
- Image Source
- Size
- ...



Automatismus?

Automatic Parameter Sources

Azure Naming Tool

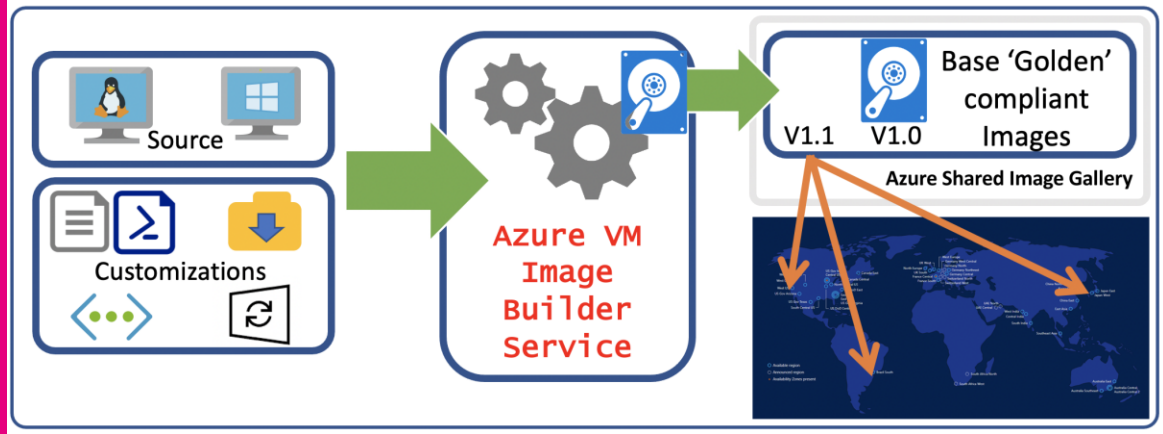
- Naming API
- standardisierte Ressourcennamen aus Naming Convention
- REST API kompatibel

Azure IPAM

- Automatisiertes IPAM für Azure
- Bereitstellung von IP Adressbereiche für Deployments
- REST API

Azure Image Builder

- Basierend auf HashiCorp Packer
- “Image as Code“
- Erstellung von Images für verschiedene Anwendungen:
  - AVD
  - Enterprise Servers



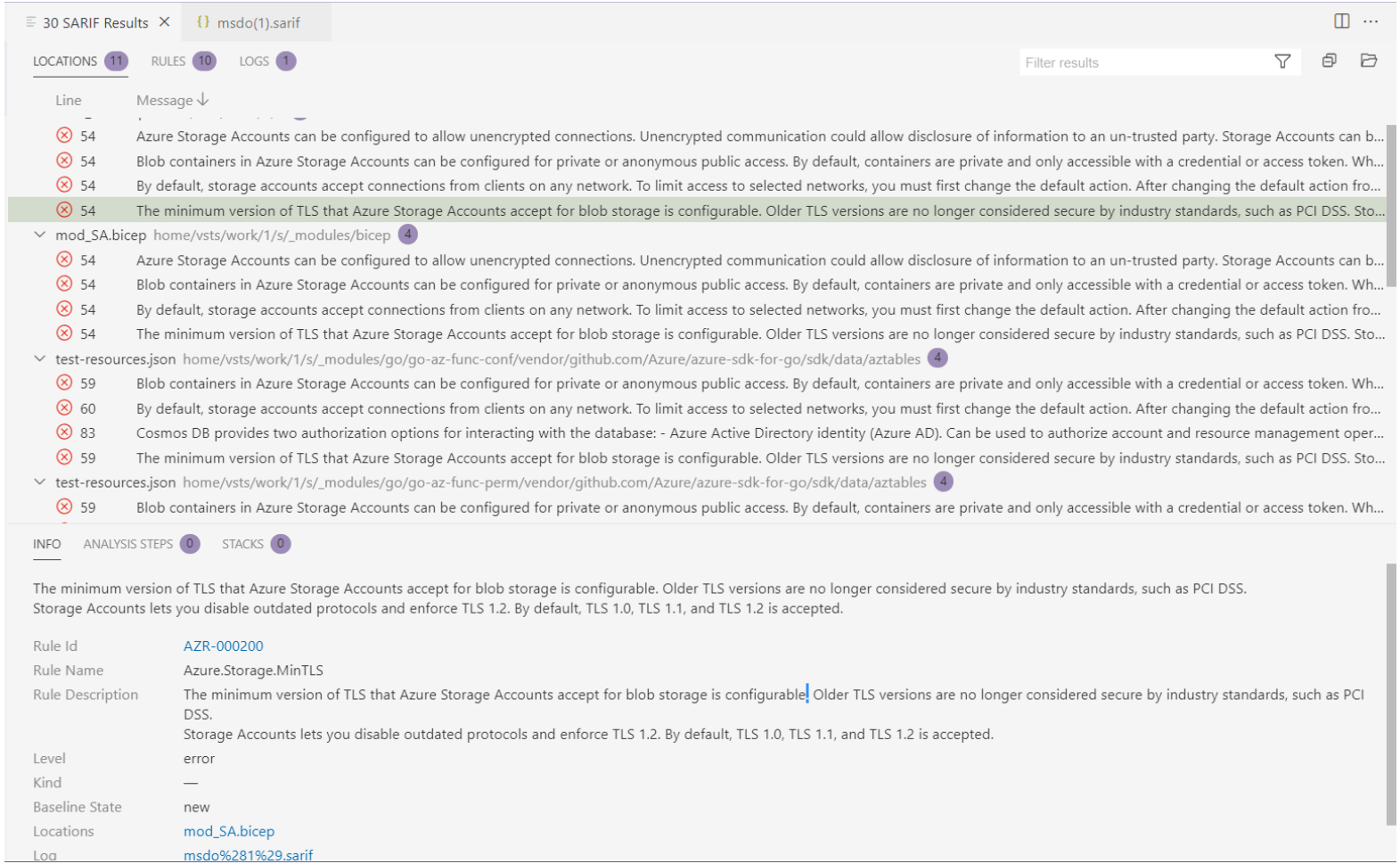
axians

Code-Security

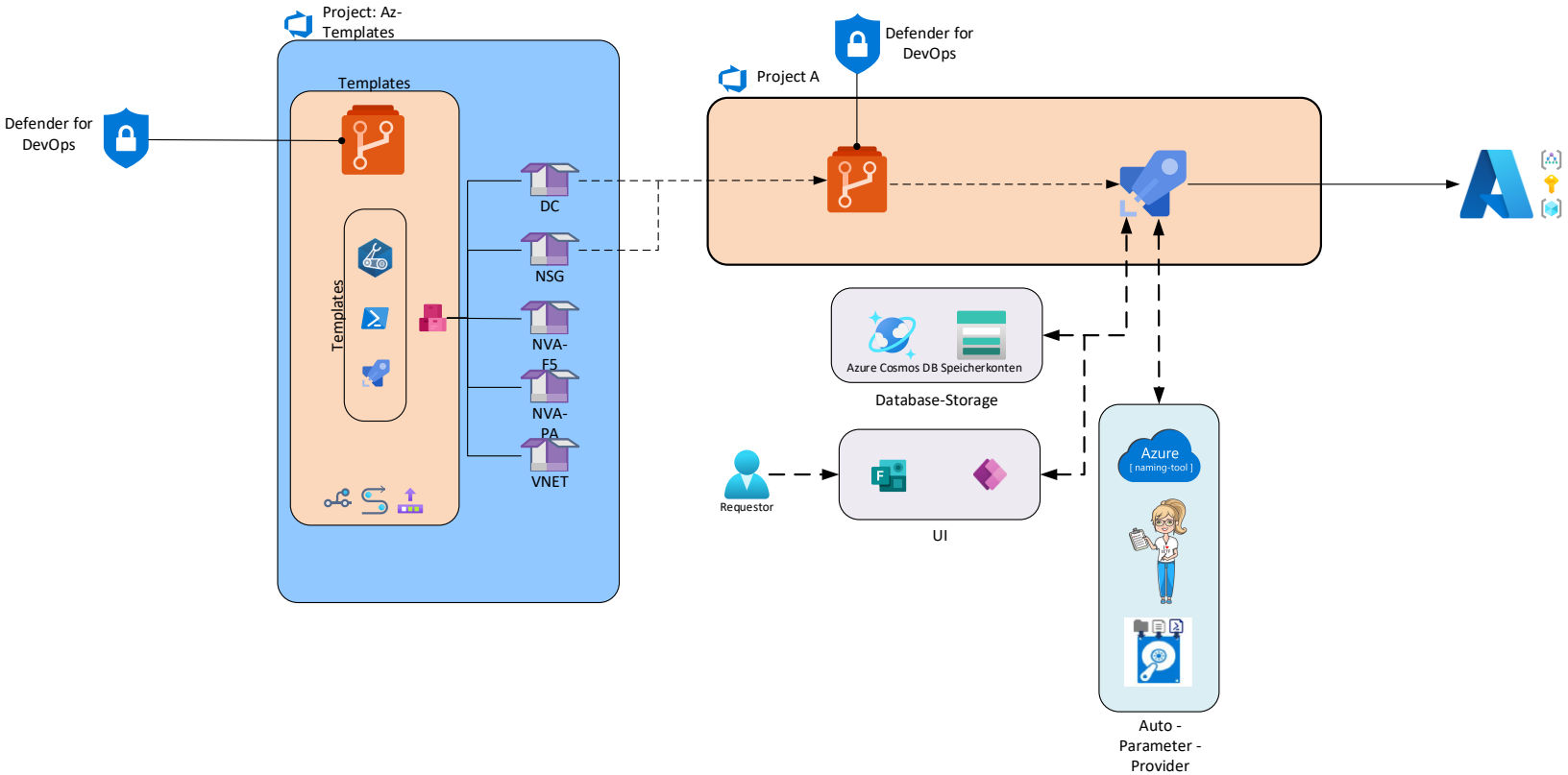
Defender for Cloud

Better say: „Defender for DevOps“

- Kann IaC-Code nach aktuellsten Microsoft-Sicherheitsrichtlinien bewerten
- Ausführung per Azure Pipeline möglich
- Output: SARIF-File


30 SARIF Results
msdo(1).sarif
LOCATIONS 11
RULES 10
LOGS 1
Filter results
Line
Message
54 Azure Storage Accounts can be configured to allow unencrypted connections. Unencrypted communication could allow disclosure of information to an un-trusted party. Storage Accounts can b...
54 Blob containers in Azure Storage Accounts can be configured for private or anonymous public access. By default, containers are private and only accessible with a credential or access token. Wh...
54 By default, storage accounts accept connections from clients on any network. To limit access to selected networks, you must first change the default action. After changing the default action fro...
54 The minimum version of TLS that Azure Storage Accounts accept for blob storage is configurable. Older TLS versions are no longer considered secure by industry standards, such as PCI DSS. Sto...
mod_SA.bicep home/vsts/work/1/s/_modules/bicep 4
54 Azure Storage Accounts can be configured to allow unencrypted connections. Unencrypted communication could allow disclosure of information to an un-trusted party. Storage Accounts can b...
54 Blob containers in Azure Storage Accounts can be configured for private or anonymous public access. By default, containers are private and only accessible with a credential or access token. Wh...
54 By default, storage accounts accept connections from clients on any network. To limit access to selected networks, you must first change the default action. After changing the default action fro...
54 The minimum version of TLS that Azure Storage Accounts accept for blob storage is configurable. Older TLS versions are no longer considered secure by industry standards, such as PCI DSS. Sto...
test-resources.json home/vsts/work/1/s/_modules/go/go-az-func-conf/vendor/github.com/Azure/azure-sdk-for-go/sdk/data/aztables 4
59 Blob containers in Azure Storage Accounts can be configured for private or anonymous public access. By default, containers are private and only accessible with a credential or access token. Wh...
60 By default, storage accounts accept connections from clients on any network. To limit access to selected networks, you must first change the default action. After changing the default action fro...
83 Cosmos DB provides two authorization options for interacting with the database: - Azure Active Directory identity (Azure AD). Can be used to authorize account and resource management oper...
59 The minimum version of TLS that Azure Storage Accounts accept for blob storage is configurable. Older TLS versions are no longer considered secure by industry standards, such as PCI DSS. Sto...
test-resources.json home/vsts/work/1/s/_modules/go/go-az-func-perm/vendor/github.com/Azure/azure-sdk-for-go/sdk/data/aztables 4
59 Blob containers in Azure Storage Accounts can be configured for private or anonymous public access. By default, containers are private and only accessible with a credential or access token. Wh...
INFO
ANALYSIS STEPS 0
STACKS 0
The minimum version of TLS that Azure Storage Accounts accept for blob storage is configurable. Older TLS versions are no longer considered secure by industry standards, such as PCI DSS.
Storage Accounts lets you disable outdated protocols and enforce TLS 1.2. By default, TLS 1.0, TLS 1.1, and TLS 1.2 is accepted.
Rule Id AZR-000200
Rule Name Azure.Storage.MinTLS
Rule Description The minimum version of TLS that Azure Storage Accounts accept for blob storage is configurable. Older TLS versions are no longer considered secure by industry standards, such as PCI DSS.
Storage Accounts lets you disable outdated protocols and enforce TLS 1.2. By default, TLS 1.0, TLS 1.1, and TLS 1.2 is accepted.
Level error
Kind —
Baseline State new
Locations mod_SA.bicep
Log msdo%281%29.sarif

Zusammenfassung



Demo

Thanks a lot!

BERNHARD FLÜR

bernhard.fluer@axians.at

- 16:15 – 16:55 – Hybrid Cloud Track

Eagle Eye - Flying a Drone using Unity 3D Across the Mixed Reality Spectrum

Experts Live Austria

Bicep Playground

```bicep
param serverName string = uniqueString('sql', resourceGroup().id)
param sqlDBName string = 'SampleDB'
param location string = resourceGroup().location
param administratorLogin string

@secure()
param administratorLoginPassword string

resource server 'Microsoft.Sql/servers@2019-06-01-preview' = {
  name: serverName
  location: location
  properties: {
    administratorLogin: administratorLogin
    administratorLoginPassword: administratorLoginPassword
  }
}

resource sqlDB 'Microsoft.Sql/servers/databases@2020-08-01-preview' = {
  name: '${server.name}/${sqlDBName}'
  location: location
  sku: {
    name: 'Standard'
    tier: 'Standard'
  }
}
```

```json
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "serverName": {
      "value": ""
    },
    "sqlDBName": {
      "value": ""
    },
    "location": {
      "value": ""
    },
    "administratorLogin": {
      "value": ""
    },
    "administratorLoginPassword": {
      "value": ""
    }
  }
}
```

main.bicepparam > ...

```bicep
using './main.bicep'

param serverName = ''
param sqlDBName = ''
param location = ''
param administratorLogin = ''
param administratorLoginPassword = ''
```