

# DEVOPS kickstarter

van idee naar productie in Azure!

# Welkom...

... bij de 2<sup>de</sup>  TechUpdate van 2018 !

Sessie:

- Informeel
- Hands-on (dus niet alleen sheets)
- Vragen staat/is vrij
- Mogelijkheid voor vragen na de sessie of op later tijdstip



# Scope

Focus op de DevOps 'enablers':

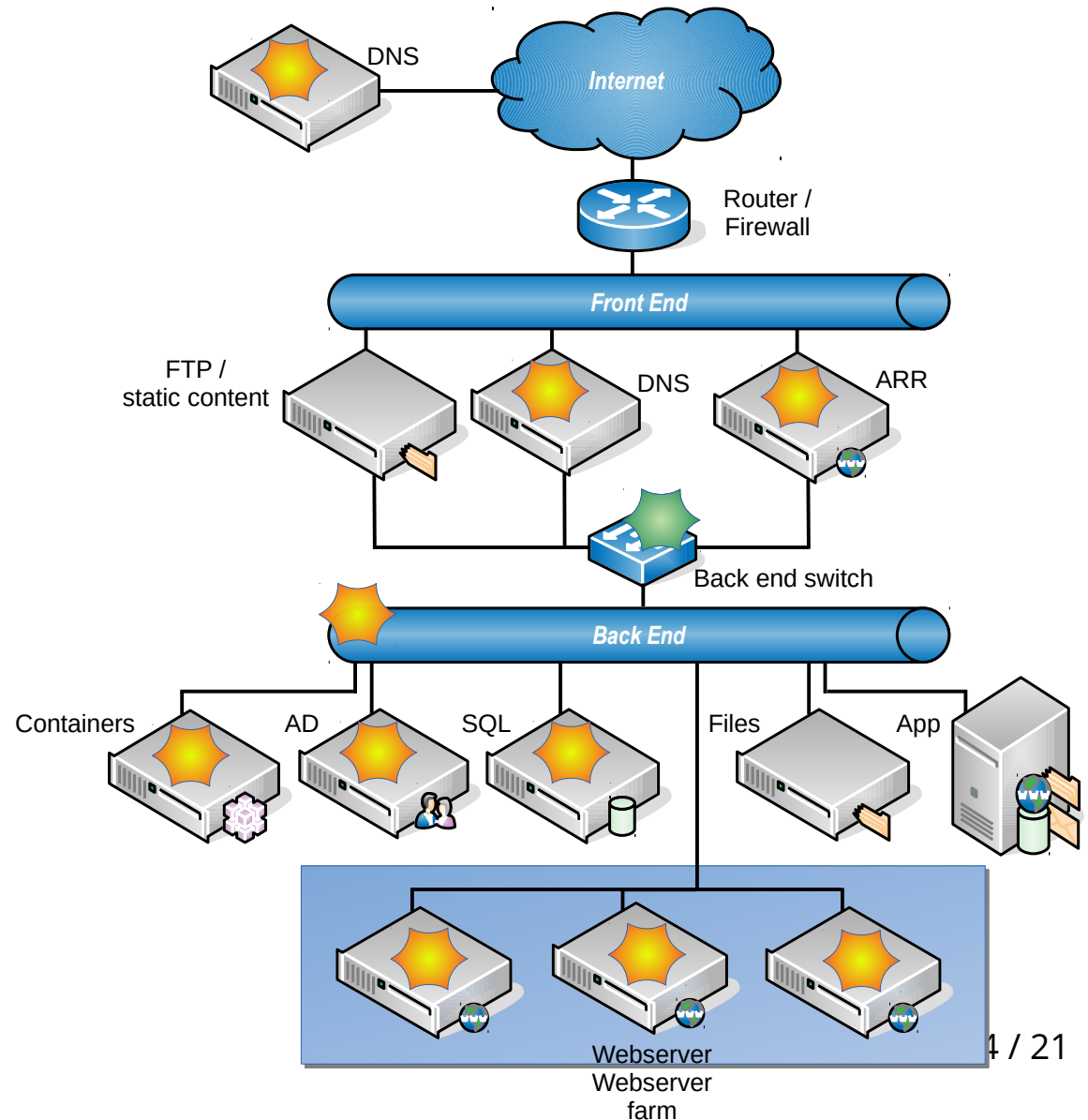
- CI (Continuous Integration)
- CD (Continuous Delivery)
- IaC (Infrastructure-as-Code)

In context van online diensten:

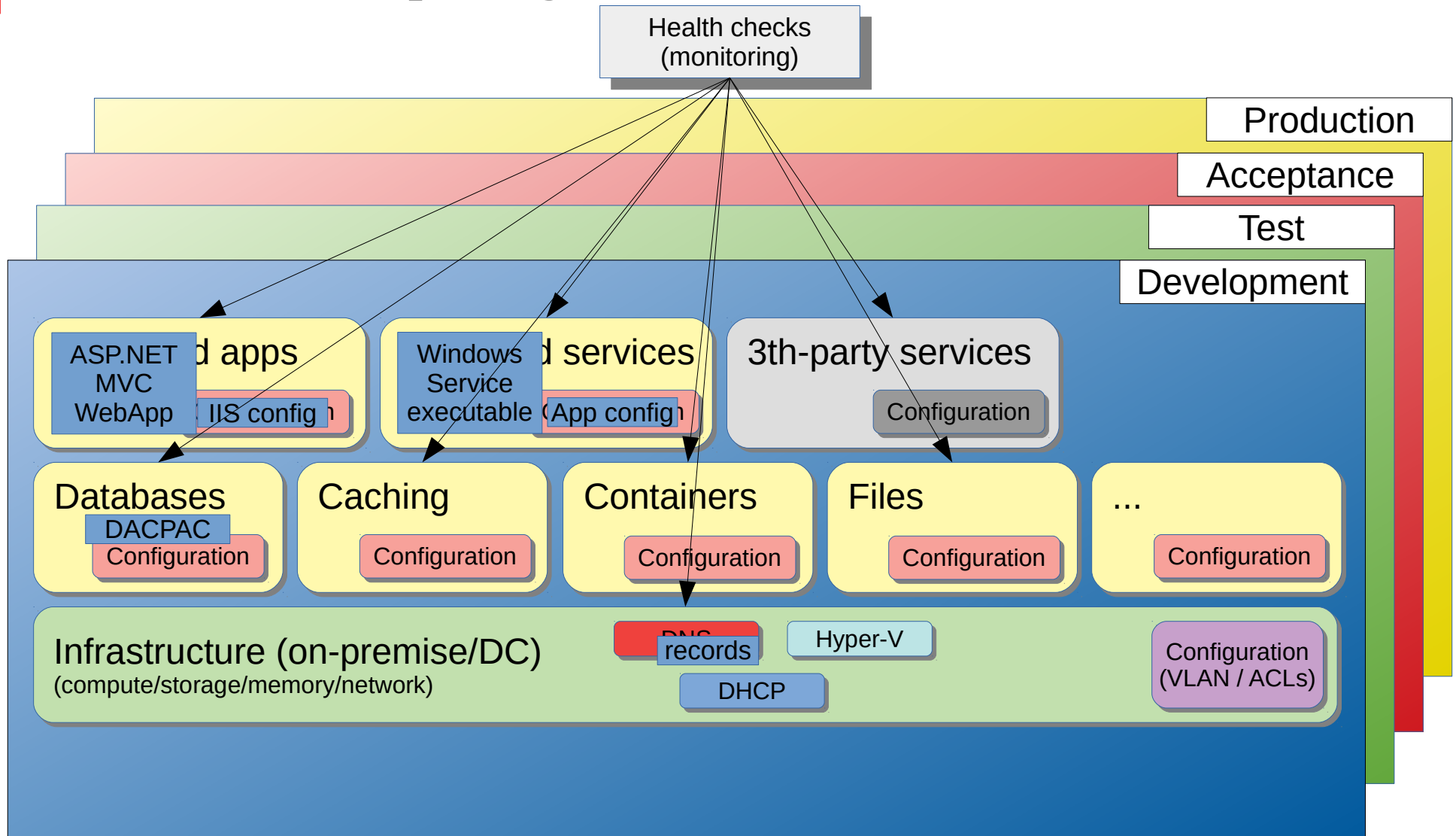
- VSTS (Visual Studio Team Services)
- Microsoft Azure

# Landscape - App Lifecycle

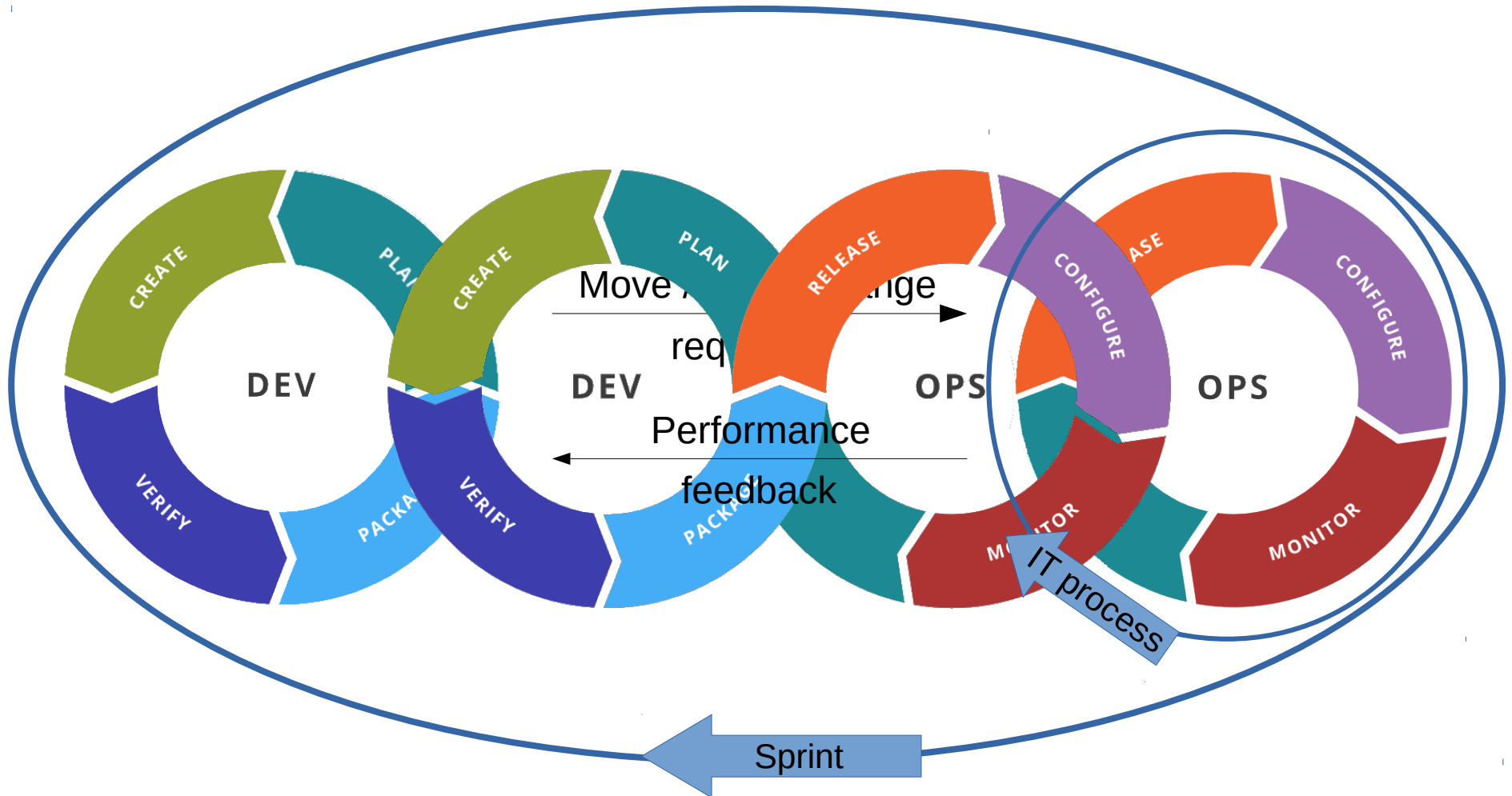
- Self hosted (on-premise/DC)
- Provider hosted ('traditional' IaaS)



# Dev / Deploy / Monitor



# Development & Operations (waarom)



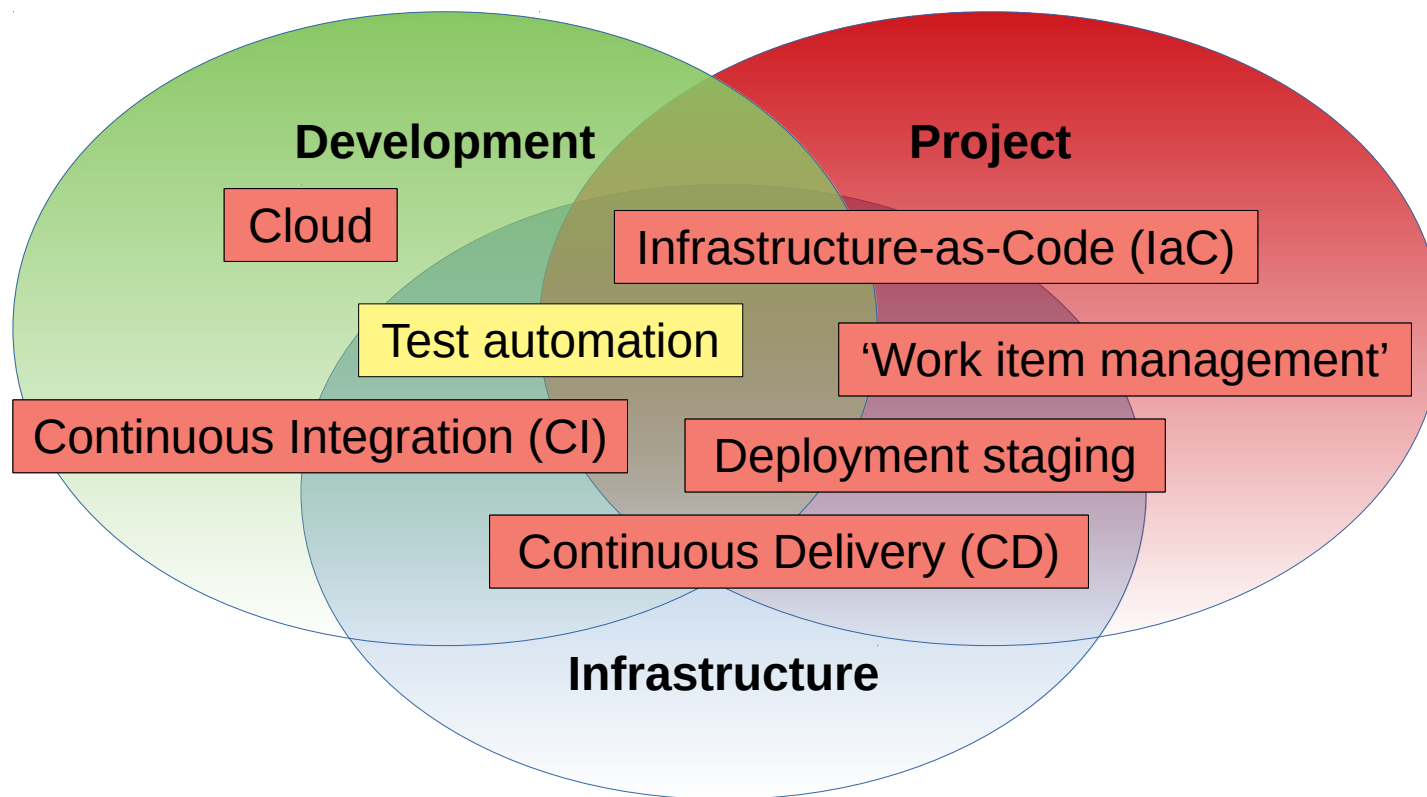
## Functioneel (TTM)

- toevoegingen
- aanpassingen
- bug-fixes (inc. MTTR)

## Niet functioneel

- schaalbaarheid
- beschikbaarheid
- performance

# Development & Operations (hoe)





# Scenario's

## 1) 'van idee naar productie in Azure!'

- End-to-end demo CI/CD
- Gebaseerd op Azure DevOps project (WebApp)
- Azure PaaS

## 2) IaC; ARM templates en CI/CD (ACC/PROD stages)

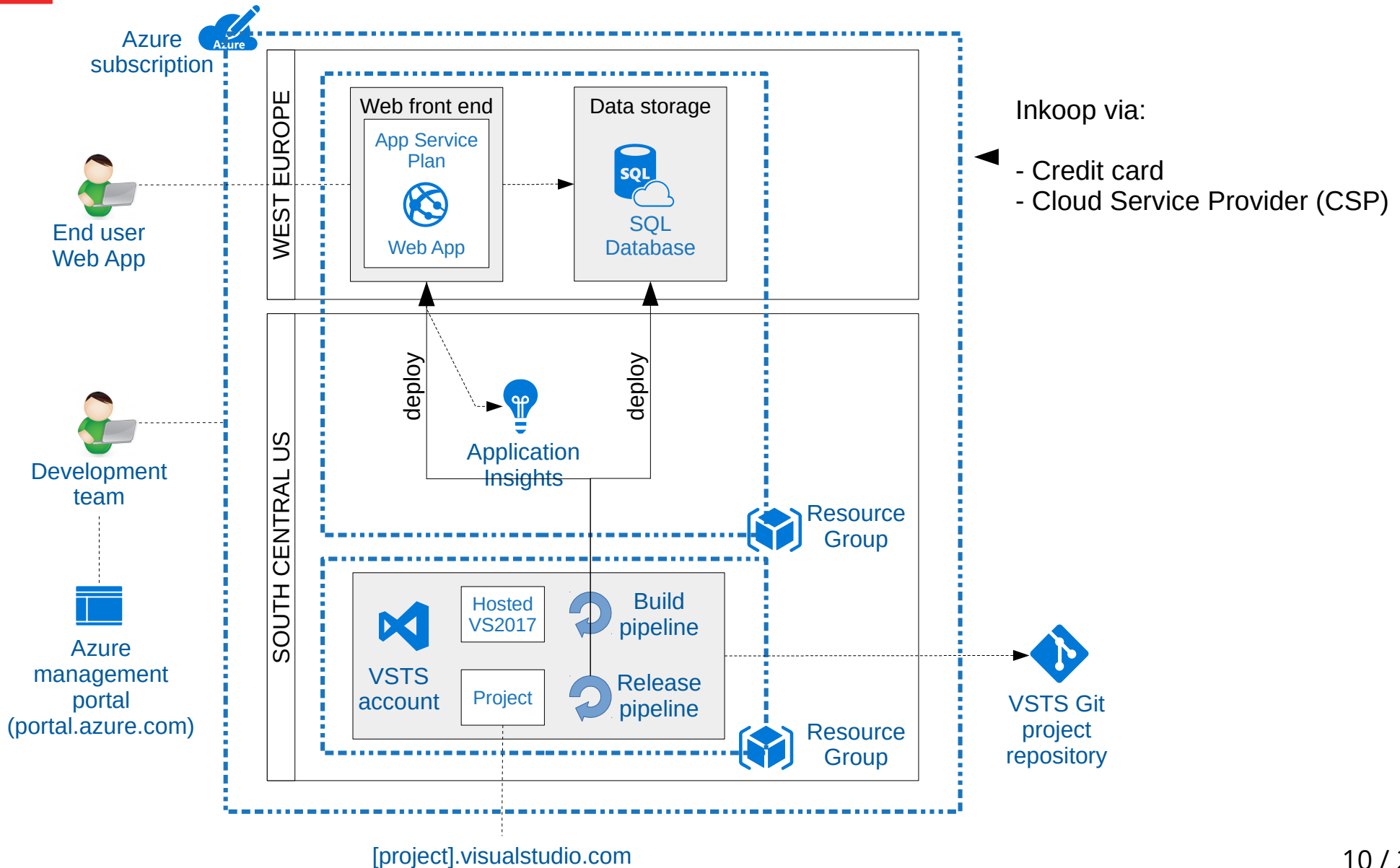
- Azure IaaS/PaaS provisioning/configuration
- Staging environments



# Scenario

# 1

# Azure DevOps project - internals





# #Tips

- Proef abonnement beschikbaar (€170 of 12 mnd)
- Hands-on labs (gratis):  
<https://www.microsoft.com/handsonlabs>
- Azure binnen eigen DC (housing)/on-premise ?  
Azure Stack
- Pricing calculator:  
<https://azure.microsoft.com/en-us/pricing/calculator/>



Scenario 1

# Resultaat

# Azure DevOps – wat krijg je

- Project homepage en dashboard – samenwerken
- Proces ondersteuning (backlog e.d.)
- Versiebeheer (TFVC / Git)
- Test automatisering (unit, load, ...)
- Hosted build (environment) pipelines (ook self hosted)
- Release pipelines
- Monitoring/Health checks – applicatie laag, services en kosten
- RBAC voor Azure en VSTS

# Azure DevOps – wat kan je

- Schaalbaarheid applicatie garanderen
- Beschikbaarheid applicatie garanderen
- Snel inspelen op functionele en kwaliteits verbeteringen (application insights, services monitoring, health checks)
- DevOps 'template' gebruiken als startpunt (kickstarter 😊) en eventueel verder uitbouwen

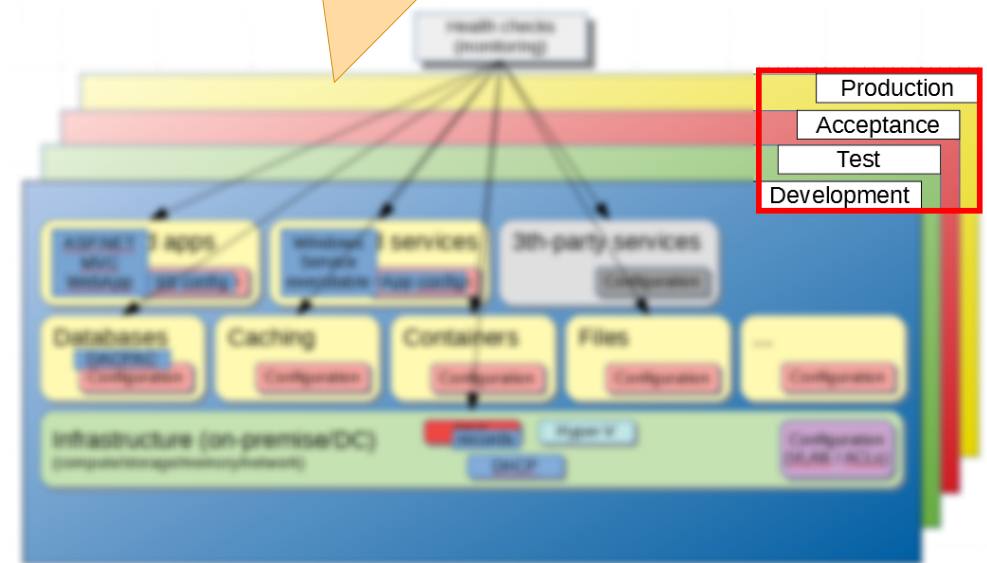
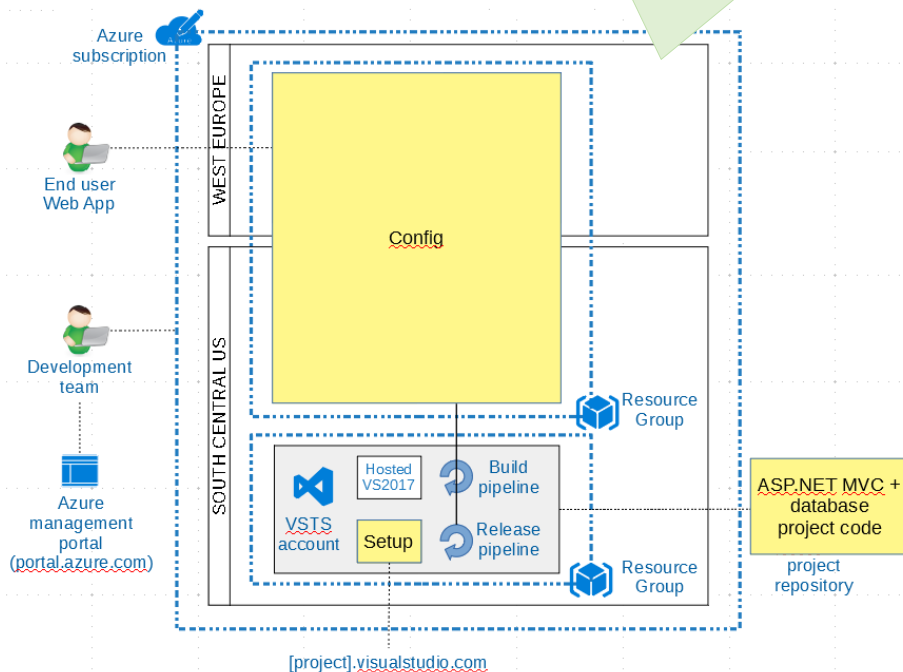
# Next step...

Reproduceerbaarheid ?  
Traceerbaarheid ?

Resources met 'de hand' aanmaken  
via het Azure portal

Haalbaarheid ?

Meerdere omgevingen



# ARM templates (IaC)

- In DevOps voorbeeld; complete provisioning services (web app, infra e.d.) door Azure
- IaC voorbeeld; provisioning services middels ARM templates naar acceptatie en productie omgeving
- ARM templates; declaratieve syntax om infrastructuur te beschrijven
- Op basis van dependencies, linked templates en parameters (json, resources en variables)
- Van toepassing op alle services (IaaS, PaaS)

# ARM template - indeling/syntax

```
"resources": [  
  {  
    "name": "[p4c.uniqueName(parameters('storageNamePrefix'))]",  
    "type": "Microsoft.Storage/storageAccounts",  
    "apiVersion": "2016-01-01",  
    "sku": {  
      "name": "Standard_LRS"  
    },  
    "kind": "Storage",  
    "location": "South Central US",  
  },  
],  
"outputs": {
```

```
New-AzureRmResourceGroupDeployment -Mode Complete -Name ExampleDeployment `
  -ResourceGroupName ExampleResourceGroup -TemplateFile c:\example.json `
  -TemplateParameterFile c:\example.parameters.json `
  -storageNamePrefix overriddenName
```

Dependencies (dependsOn)

Linked templates (nesting)



Scenario

2



# Scenario 3 ?

- Combinatie scenario 1 (app) en scenario 2 (ARM) mogelijk
- Applicatie code en ARM templates in VS .sln

# Azure 'interactivity'

- Grafische front-end
  - Azure portal ([portal.azure.com](https://portal.azure.com))
- Windows Powershell (met modules)
- Azure CLI (cross platform)
- Azure Resource Manager (ARM) templates
- Via code (bibliotheeken)
- Maken allemaal gebruik van de Azure REST-API



# Als laatste...

- Vragen ?
- Opmerkingen ?
- Suggesties ?
- Wensen ?