Infrastructure as Code: Scaling deployments across customers

Dutch DevOps & GitHub Community presents:
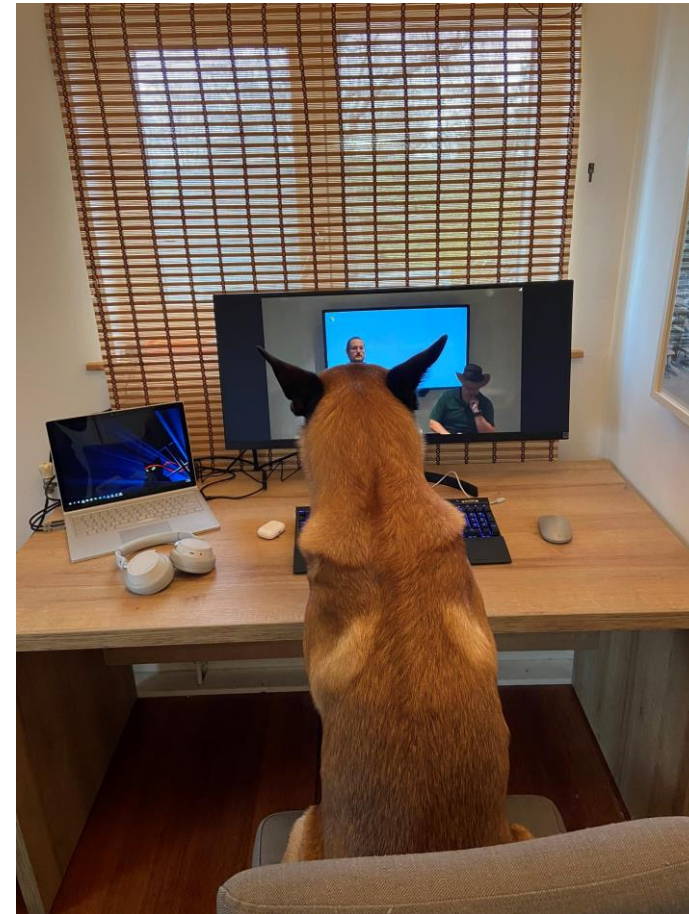
D-DOG

August 13

Virtual meetup @ 19h CEST
Get Tickets: https://ddog.nl

Speakers:
Martin Woodward
Barbara Forbes
Wesley Haakman

Wesley Haakman

Lead Azure Architect @ Intercept
Microsoft Azure MVP

@whaakman

www.linkedin.com/in/wesleyhaakman

whaakman@intercept.nl

Agenda

Infrastructure as Code

ARM Templates or scripting?

Re-using deployments for multiple customers

Unpopular opinion:

Infrastructure as code should be more than just ARM templates

Infrastructure as Code

- Complexity versus usability
- It's about creating a balance
- Start simple → baby steps

Who are you writing code for?

The thing with ARM templates..

- "I really like them"
- The learning curve is pretty steep
- Chances are that your target audience has no experience with ARM Templates
- You don't really have to stuff everything into a template
- It's a bit too much "Dev"

The thing with ARM templates..

- More often than not we see devs developing arm templates
- Sometimes Ops people don't have the background making that learning curve even steeper
- Scripting feels more natural

Common Use Case

- A software company has set their mind of hosting their solution on Azure

- They provide services globally

- They are refactoring and rebuilding to ensure their solution runs on WebApps and Azure SQL

- The technical team consists of Developers and System Administrators (Ops)

- Customer Data comes from a CRM solution

- You are tasked with supporting the technical team in automating deployments

- Time is of the essence; the customer already cancelled the contract with the current hosting provider

The challenges

- They're relatively new to Azure

- Throwing ARM templates at them might not be the best solution

- Automating deployments goes far beyond just ARM Templates

- Writing logic in ARM Templates steepens the learning curve even further
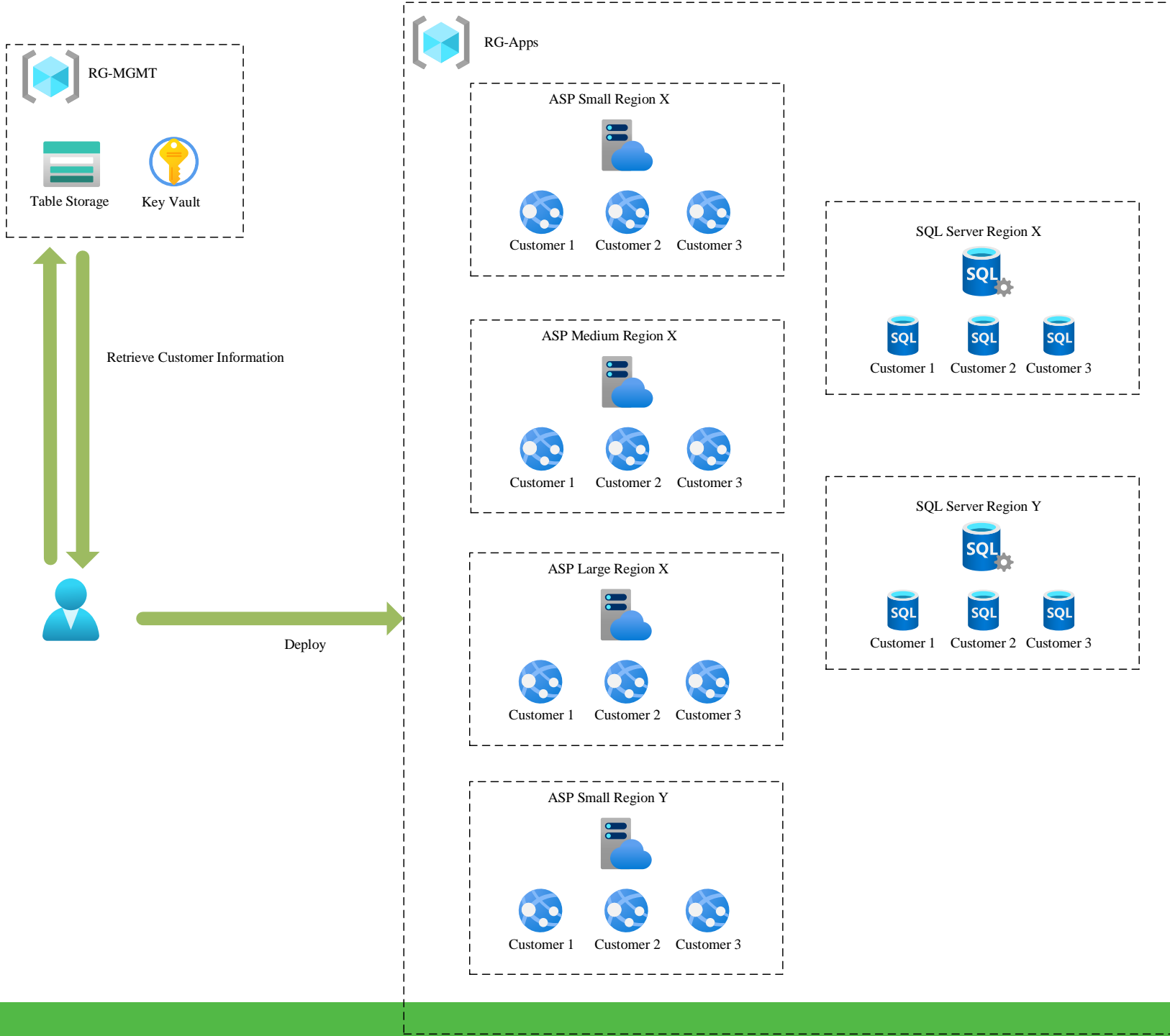
Demo time!

The Result

- Devs can use the ARM Templates for multiple scenario's (think dev/test)

- They can modify and share new templates with the people responsible for deployments

- The scripting can be executed manually, hosted in a pipeline or in Azure Automation

- If you're creative, turn it into a couple of Azure Functions

- There's a balance between complexity and usability

Thank you!

https://github.com/whaakman/iac-scale-customers