Lessons learned from moving legacy application to Azure

Why this topic?

Me

- Heikki-Jussi Niemi
- Software developer @ Protacon
- 20% Internal improvements / internal Azure consulting, etc
- 80% Outsourced software development

Protacon

Background

Project

Business

- ERP system
- Mission critical
- Used mainly 07:00 - 18:00
- Used for billing
  - End of month usage spikes
  - Handles money

Technical

- Legacy
- ASP.NET MVC
- SQL Server
- Windows services
- Multitenant
- 3rd party integrations
- On-premise service
  - Printers, payment terminals...

Why cloud?

- Reduced costs
  - Wasted resources (scaling)
  - Hardware management
  - etc.
- Good time for moving
  - New region - Testing without migrating
  - Old region will be updated/migrated later
- Automated environment creation

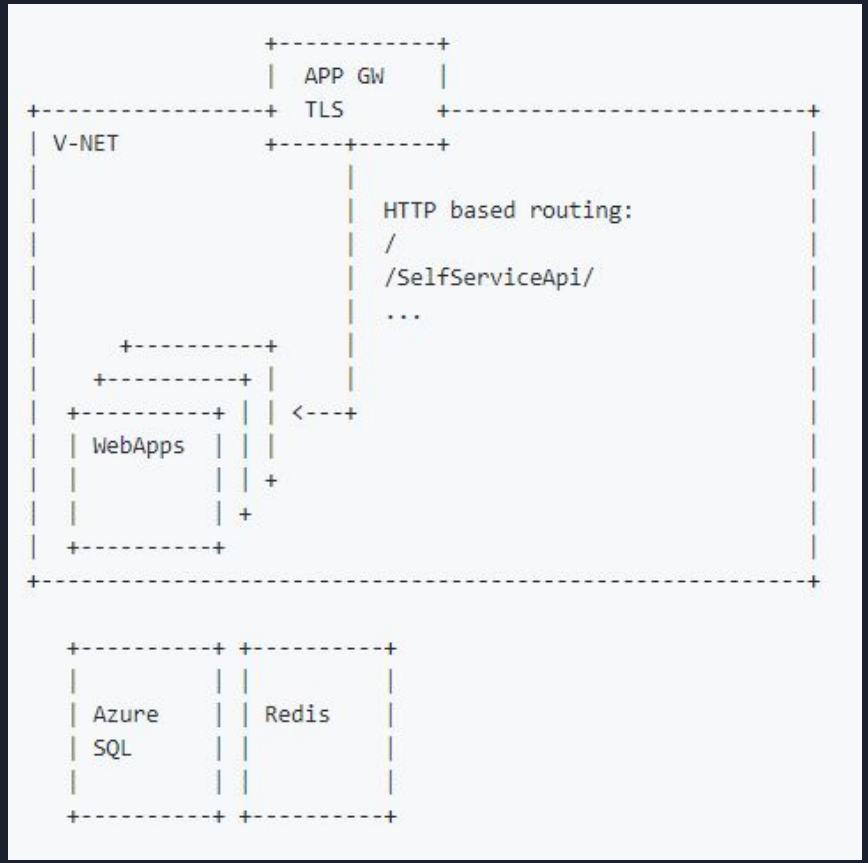
Changes / Considerations

- Windows services -> Web Jobs
- Static IP -> VM with HTTP Proxy
- App Gateway routes to subdomains
- Old file logging -> Application Insights
- ASP.NET SQL Session State -> Redis
- Hangfire Database -> Webjobs, App service
- SLA
- Old project format -> New project format (just to make things easier)
- IT monitoring solutions is still undecided

Azure Architecture

Shared resources (not shown in picture)

- Storage for devops
- Proxy VM
- PDF storage
- ...



Continuous Integration

Old

- Teamcity
- Configurations done by hand

New

- Azure Dev Ops
  - Jenkins was considered because it's in in-house use, but Azure Devops has better out-of-the-box support for Azure
- Configuration as code
  - YAML
  - Release-pipeline doesn't support configuration as code yet :(
- Blob storage is used as artifact storage
  - Pekka had some performance issues with artifact storage provided by Pipelines

General observations

- Deployment times varies
  - Redis cache was consistently slow
  - Makes building from scratch painful
  - Makes debugging ARM templates very painful
- App Service VNet tools are not there yet
- Calculating service prices is hard
  - We have 3rd party provider for Azure Services (CSP?)
  - Old servers hosted in-house
  - Developers don’t have visibility to prices
- Az cli vs. AzureRm module vs. Az module

Lessons

Lesson: ARM template debugging

Pekka Savolainen 09:51

myisin kyllä vaikka mummoni että azuren arm deployment antais parempia virheitä

Message=The template deployment 'appgateway.json' is not valid according to the validation procedure.

tämä selvä

Pekka Savolainen 10:13

ilmeisesti vika oli se että app gateway viittasi basic tierin IP osoitteeseen vaikka vaatii standardin tierin

10:13 voi kun olis sanonut että "Expected public ip address koaksoad to be attleast tier standard, got basic."

It’s hard.

Lesson: Application gateway deployment

- Certificate auto renewal from Key Vault is not supported yet
  - This should be coming
- Referencing KeyVault secret can’t be done inside template
- Creating with ARM template was pain in the ass
  - Bad error messages
  - Slow

Lesson: Huge template vs. Smaller templates

Huge

- Everything is visible in same template
- Inner references

Small (deployed separately)

- Easier to debug
- Faster to deploy segments parts
- Might remove data if segments are deployed in wrong order

Lesson: IP addresses

App Service needs static outgoing IP address

- App Service Environment (isolated) was too expensive
- App Service VNet tooling lacking

This was solved with VM hosting HTTP proxy (Squid) with static IP

Lesson: App Service with Containers

Pekka Savolainen 11:17

https://www.azuredevopslabs.com/labs/vstsextend/docker/ eli toi käytännössä, tahtotilana oli että sais imettyä GCR:ästä imaget koska CI tuottaa niitä sinne (jaettu osa, käytetään muualla) -> authentikointi tuntu tekemättömältä paikalta

> azuredevopslabs.com
> **Deploying a Docker based web application to Azure App Service**
> Check out the tutorial 'Deploying a Docker based web application to Azure App Service' and many more at https://aka.ms/devopslabs

- Deployment from private GCR caused problems
  - Authentication
  - Debugging was hard
- Deployment from private Azure Repository worked

Lesson: Term mixups

- Permission management in Key Vault
  - Object ID - There was a different ID in portal
- Service Principal vs. App registration vs. Enterprise applications

Lesson: Communication between IT & Developers

IT people and developers have different perspectives

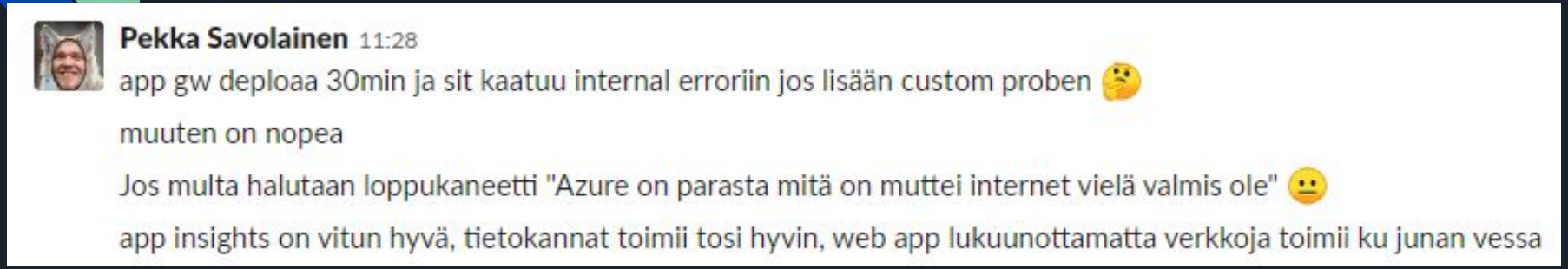
- Security
- Monitoring
  - Integrations to current systems
- Ease of deployment
  - Current systems vs. new systems
- Costs

Lesson: Powershell Azure Module version

Powershell Azure Module was updating during this

- AzureRm module vs. Az module
- Version(s) supported by Pipeline might be different than the one installed in developer computer

Conclusion



- There is room for improvement, but things are better than before.
- Conversion process in itself isn't cheap, consider consultants.
- About 467h done (24.5.2019), about 33% still to go
  - Estimating developer effort has been hard, but it was a known risk
  - This should break even in about 1 year.
- In our case these skills will be useful because we have similar projects coming.