Options for Migrating On-Premises Web Apps to Azure

Scott Snyder
Solutions Architect
RBA
Scott Snyder
Solutions Architect

Recent Experience

International Logistics
Assisted in enterprise wide migration of an on-premises Identity Provider to a cloud based IdP spanning hundreds of applications including both APIs and UIs.

Consumer Packaged Goods
Contributed to the flagship website suite which serviced over 650MM page views per year.

Financial Services – Portfolio Risk Management
Provided technical leadership and mentoring focusing on code modernization and software architecture.

Financial Services – Travel and Expense Management
Strategized with CEO and Product Manager on long term business goals and technological need. Lead a team of developers covering the entire Microsoft stack. Drove CI/CD adoption and code base modernization.

Skills & Expertise

Software Tools/ Technologies
- Microsoft Azure
- C#/VB.NET
- ASP.NET
- Web Services
- SQL/T-SQL
- MSTest/NUnit
- Moq
- Javascript/JQuery
- Jasmine
- XML
- HTML/CSS
- Java/Groovy

Education

Bachelor of Science - Computer Science
University of Wisconsin - Madison
Motivations

- What problem(s) are you trying to solve?
- How is success measured?
- Cloud Adoption Framework from Microsoft
Most common cloud motivation?

Cost Savings
# Modernization

<table>
<thead>
<tr>
<th>Processes &amp; Techniques</th>
<th>Design Patterns</th>
<th>Technologies</th>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile</td>
<td>Microservices</td>
<td>Containers</td>
<td>Automatic Scaling</td>
</tr>
<tr>
<td>DevOps</td>
<td>Inversion of Control (IoC)</td>
<td>Automation Servers</td>
<td>Platform-as-a-Service (PaaS)</td>
</tr>
<tr>
<td>Automated Testing</td>
<td>Circuit Breaker</td>
<td>Jenkins</td>
<td>Serverless</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Azure DevOps Pipelines</td>
<td>Infrastructure-as-Code (IaC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Terraform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Azure Resource Manager (ARM) Templates</td>
</tr>
</tbody>
</table>
Modernization

What does that get us?
- It allows IT to be more nimble to business needs and strategy which increases our value to the business.

For the business
- Modernization means reducing time to market, and enabling an organization to be more effective, efficient and agile.
Where does Azure Fit?

- Physical Infrastructure
- Modernization Features
- Security

Return on Investment
Business Value
Scaling

- Improves Usability
- Helps Manage Cost
- Autoscaling
### Cloud Responsibility Model

<table>
<thead>
<tr>
<th>On-Premises</th>
<th>IaaS</th>
<th>PaaS</th>
<th>SaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
</tr>
<tr>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
</tr>
<tr>
<td>OS</td>
<td>OS</td>
<td>OS</td>
<td>OS</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
</tr>
<tr>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
</tr>
<tr>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
</tr>
</tbody>
</table>

- **User responsibility**
- **Cloud provider responsibility**
Infrastructure-as-a-Service

Virtual Machine
- Straight up lift and shift
- Easiest/Quickest option
- Urgency

Virtual Machine Scale Set
- Autoscaling
- More dramatic change to deployment process
Reasons to choose IaaS

- Urgency (i.e. end of life of owned hardware)
- Full Access to VM or specific OS customizations
- Installed 3rd party software
- More configuration flexibility - especially much larger
- Isolation
## Cloud Responsibility Model

<table>
<thead>
<tr>
<th>On-Premises</th>
<th>IaaS</th>
<th>PaaS</th>
<th>SaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
</tr>
<tr>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
</tr>
<tr>
<td>OS</td>
<td>OS</td>
<td>OS</td>
<td>OS</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
</tr>
<tr>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
</tr>
<tr>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
</tr>
</tbody>
</table>

**User responsibility**

**Cloud provider responsibility**
PaaS - App Services

- Additional layer of abstraction over the VM
- Microsoft managed OS and runtime
- Autoscaling
- Basic Load Balancing
- Deployment Slots (Blue/Green deployments)
# PaaS - App Services

## Limitations
- No access to GAC
- Can’t save files to disk
- No email from within the app
- All apps deployed to an App Service run under a single app pool

## Workarounds
- Package assemblies with app
- Send to blob storage
- Send to a queue to get picked up/processed/sent
PaaS - Containers

- Containers

  - Already building containers?
    - Y: Yes
      - Can deploy to App Services?
        - Y: Yes
          - App Services
        - N: No
          - VM (IaaS)
    - N: No
      - Can deploy to App Services?
        - Y: Yes
          - App Services
        - N: No
          - VM (IaaS)
PaaS - Containers - Deployment

Service Fabric
- Dead (not really)
- Container orchestration moving away from Service Fabric

Azure Container Instances (ACI)
- Not for always on containers.
- Best for:
  - Container Image Development
  - Proof-of-Concept (POC)
  - Batch Jobs
  - Elastic Bursting (i.e. CI/CD)
PaaS - Containers - Deployment

App Services for Containers
- Same benefits as deploying directly to App Services
  - Microsoft managed OS and runtime
  - Autoscaling
  - Basic load balancing
  - Etc.

Azure Kubernetes Service (AKS)
- Complex
- You’ll end up here eventually
- All of Kubernetes benefits
- Microsoft manages Kubernetes
- Don’t pay for cluster management
Compute Summary

Infrastructure-as-a-Service
- Virtual Machines
- Virtual Machine Scale Sets

Platform-as-a-Service
- App Services
- Containers
  - App Services
  - AKS
Databases

- Infrastructure-as-a-Service
  - SQL Server on Azure VM

- Platform-as-a-Service
  - Managed Instances
  - Single Database
  - Elastic Pools

- CosmosDB
IaaS - SQL Server an Azure VM

- Lift and Shift
- Complete control of OS and SQL Server
- If you need SQL Agent
PaaS - Azure SQL

- Microsoft maintains OS and SQL Server
- High Availability (99.99%)
- Built in backups (RA-GRS)
<table>
<thead>
<tr>
<th><strong>Managed Instances</strong></th>
<th><strong>Single Database</strong></th>
<th><strong>Elastic Pool</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Near 100% compatibility</td>
<td>For predictable usage</td>
<td>Collection of databases with varying and unpredictable usage</td>
</tr>
<tr>
<td>Full SQL Server access</td>
<td>Dedicated resources</td>
<td>Autoscale within the pool’s resources</td>
</tr>
<tr>
<td>Expensive, complex, and hard to set up</td>
<td>Scale manually or programmatically</td>
<td>DBs can be moved in and out of a pool</td>
</tr>
</tbody>
</table>
Cost Management

- Azure Cost Calculator
Cost Management

- Azure Cost Calculator
- Azure Cost Management (portal)
- Save over pay-as-you-go
  - Enterprise Agreements
  - Azure Hybrid Benefit
  - Reserved Instances
- It’s a process, not a one-time exercise
Cost Management

- Cost optimization, not cost minimization
- ROI, not absolute cost
- Right Sizing
- Autoscaling
The architecture is the cost and the cost is the architecture. If you don’t like the cost, then you don’t like the architecture.