

NIC

2019

Artificial Edition

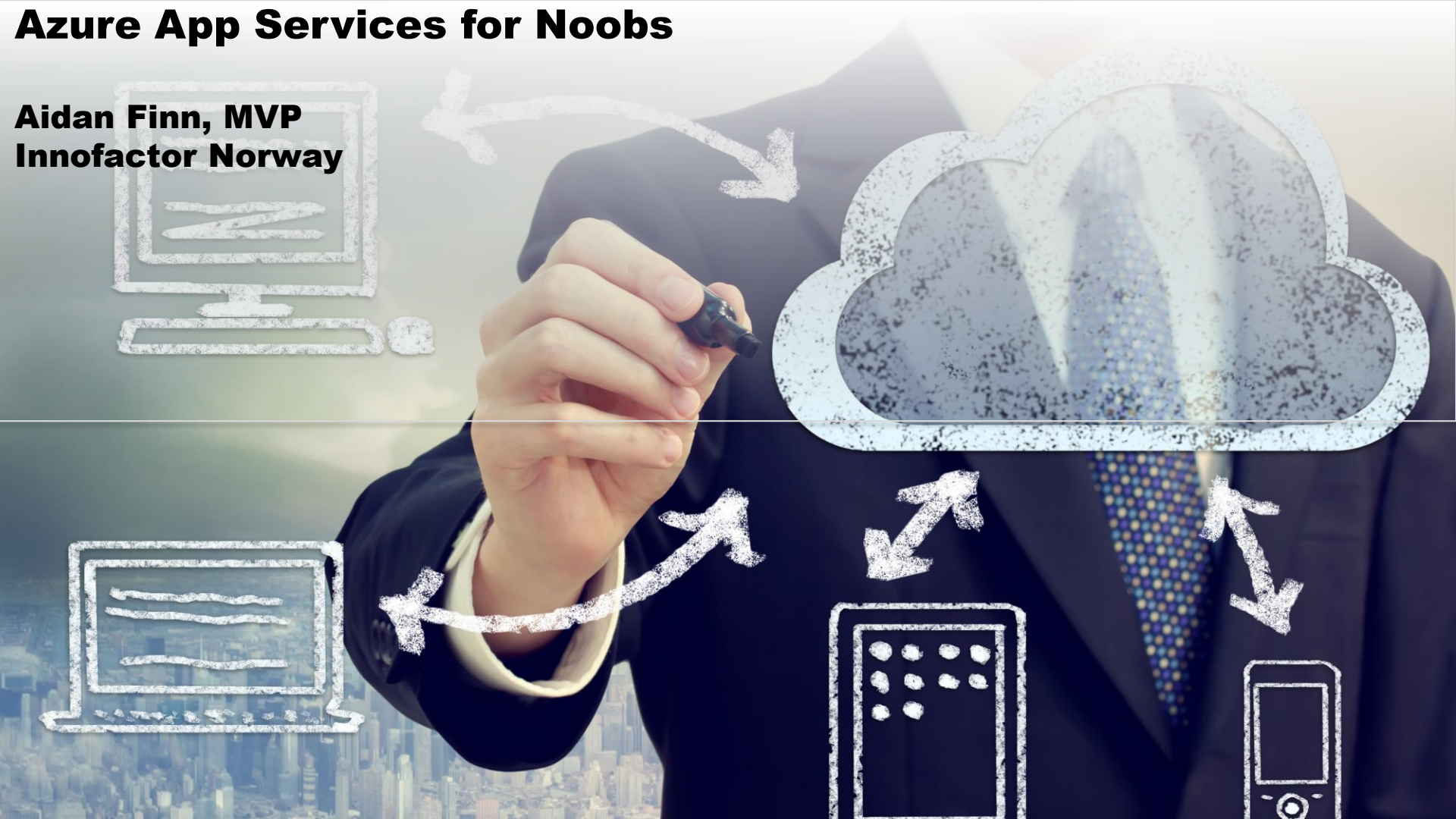
6-8 February





Azure App Services for Noobs

Aidan Finn, MVP
Innofactor Norway





About Aidan Finn

- 11 year MVP – currently Microsoft Azure (2)
 - Previously Hyper-V (9) and SCCM (1)
- Principal Consultant for Innofactor Norway
- Working as consultant/sys admin since 1996
 - Windows Server, Hyper-V, System Center, desktop management, and Azure
- <http://aidanfinn.com>
- <http://petri.com>
- @joe_elway

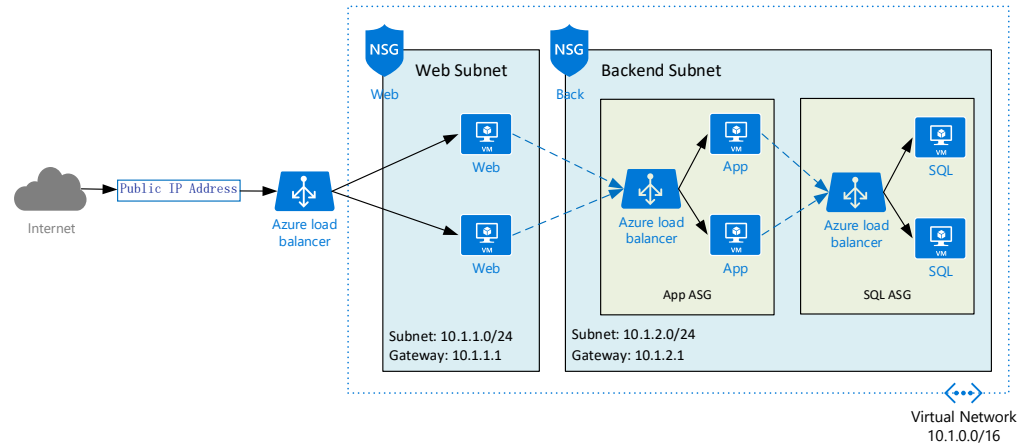
Server Struggles

Decision Time

- Imagine you want a new LOB app developed
- Go old school?
 - Database server
 - Application server
 - Thick client on a PC
- Go modern?
 - Database
 - Application layer
 - Present the client app via HTTP/HTTPS

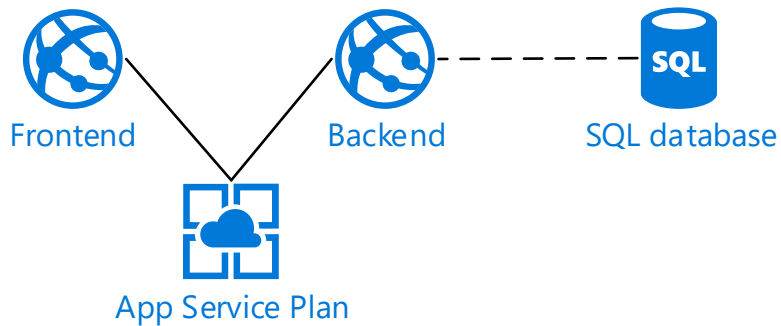
Pros and Cons

- Pros:
 - Familiarity for server admins
 - Devs know this stuff too
- Cons:
 - Costs
 - Complexity
 - Scaling
 - Time to deploy
 - Inflexibility



Azure App Services

- What if I said?
 - You could have all of the above service
 - But solve all the above problems
 - Never have to patch/upgrade an OS or database server again
- Sound good?

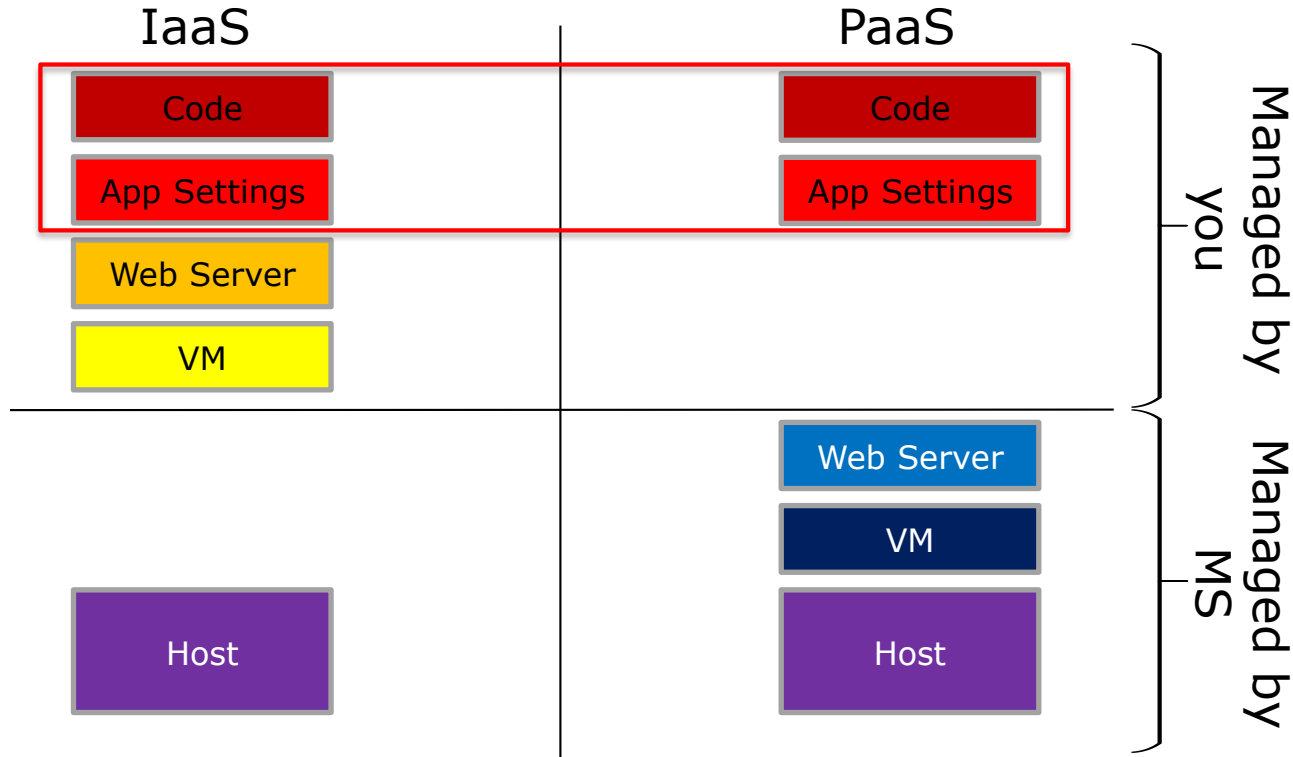


PaaS

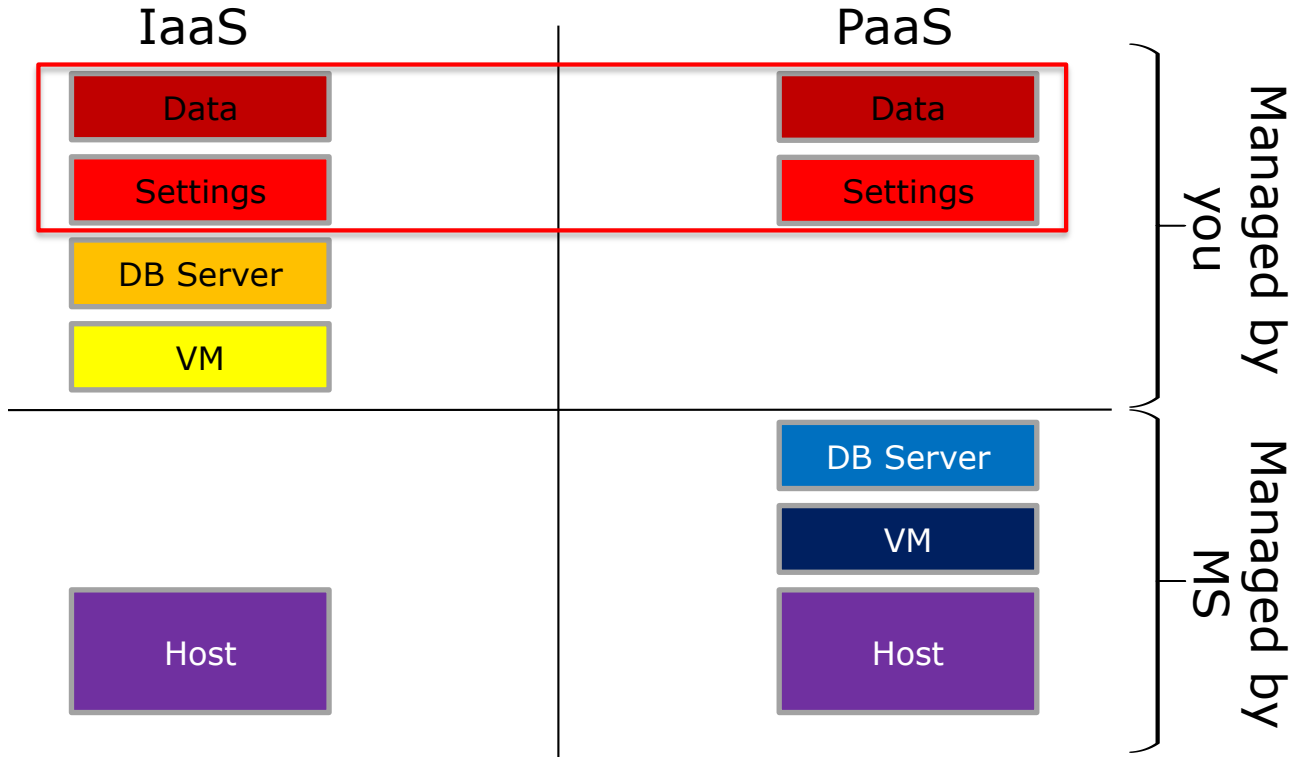
Imagine

- The simplicity of GoDaddy hosting
 - IIS with “no Windows Server”
 - Tomcat with “no Linux”
- But with enterprise features:
 - SLA / reliability
 - DevOps features
 - Scalability
 - Security & compliance architectures
 - Yes, even Docker containers!

IaaS Versus Azure App Services



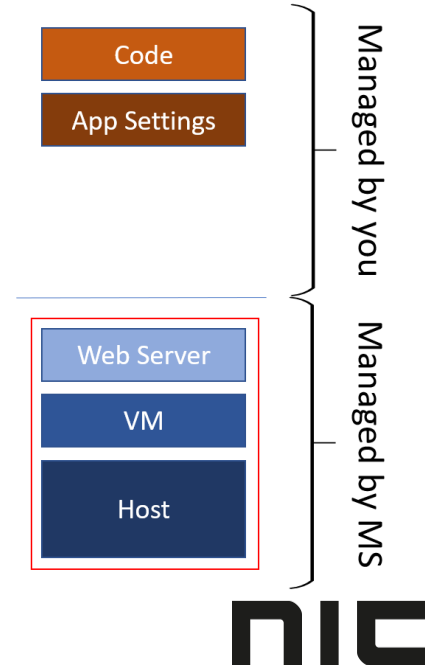
IaaS Versus Azure Database Services



App Services Terminology

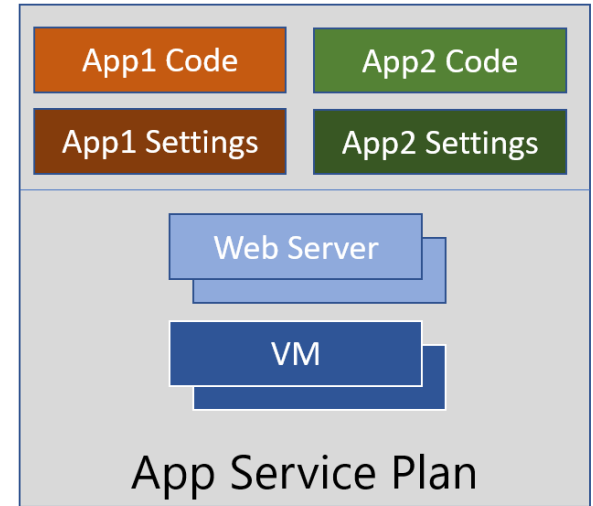
App Service Plan

- A platform on which you can build applications
- Based on Azure virtual machines
 - Windows
 - Linux
- Managed completely by Microsoft
 - Deployment
 - Patching
 - Security
- Allows you to focus on the app/code



Inside The App Service Plan

- Part of a shared environment
 - Front ends / load balancers
 - Other customers – more about privacy later
- Instances
 - 1+ web servers of a certain size
- Scale-up/down
 - Increase/decrease instance size
- Scale-out/in
 - Add/remove load-balanced instances

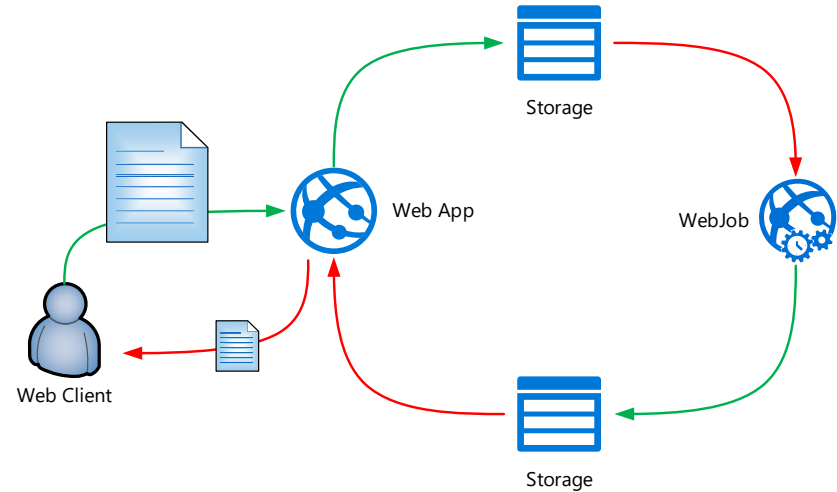


App Services

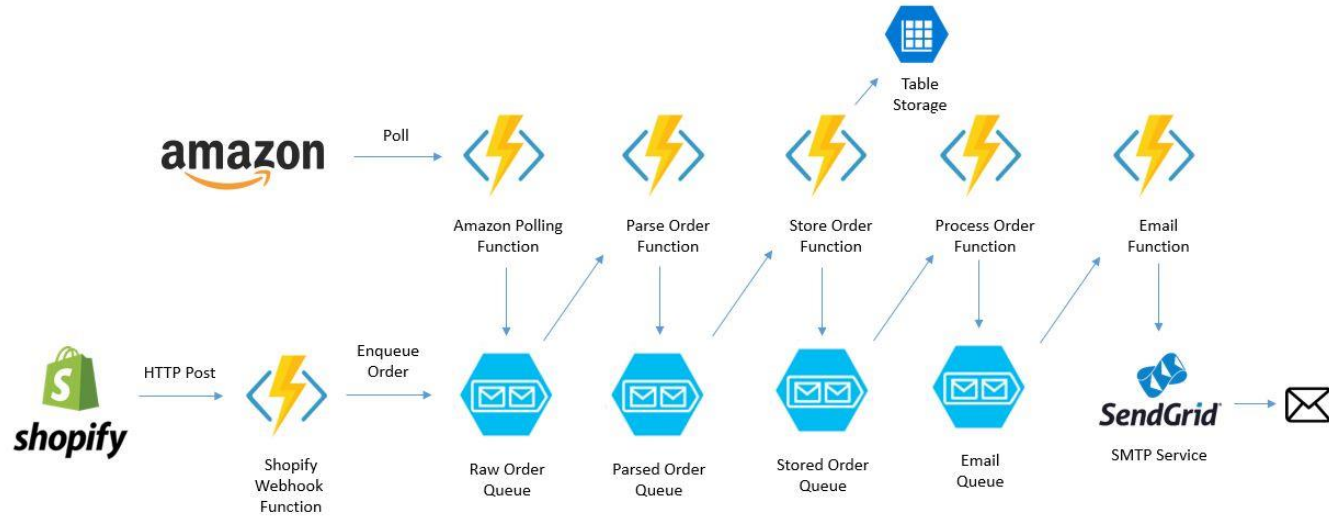
- App Service (AKA Web App) – Windows and Linux
 - Traditional site/app on a web server
 - The focus of this session
- WebJobs – Windows only
- Functions – Windows only
- API Apps – Windows only
- Mobile Apps – Windows only
- MySQL in-app

WebJob

- Hosted in an app service
- Executes some background batch task
- Started:
 - Manually
 - With the web app – runs continuously
 - Scheduled – do a batch job
- What it can run:
 - .cmd, .bat, .exe (using Windows cmd)
 - .ps1 (using PowerShell)
 - .sh (using Bash)
 - .php (using PHP)
 - .py (using Python)
 - .js (using Node.js)
 - .jar (using Java)
- Functions are a better alternative



Functions



MySQL In-App

- Create a small MySQL database for a web app
- Runs on App Service plan
 - Consumes storage from the app service plan allocation
 - Based on (Docker) containers if using Linux app service plan
- Supports PHP/MySQL Apps such as:
 - WordPress
 - Joomla
 - Drupal
- Limitations:
 - Storage capacity – the app service plan
 - Auto-scaling of the app service plan is not supported
 - Enabling local cache is not supported
 - No direct remote access to the database

Mobile Apps

- Apps need some kind of online service
 - This can be Azure App Services
- Features:
 - User authentication
 - OData v3 data source with Azure SQL, SQL Server, NoSQL, MongoDB (Azure CosmosDB), Azure Table, and more
 - Offline data sync
 - Push notifications
 - Xamarin cross-platform client SDKs

Under the Covers

App Service Plan Tiers

	FREE Try for free	SHARED Environment for dev/test	BASIC Dedicated environment for dev/test	STANDARD Run production workloads	PREMIUM Enhanced performance and scale	ISOLATED High-Performance, Security and Isolation
Web, mobile, or API apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Maximum instances	–	–	Up to 3	Up to 10	Up to 20	Up to 100*
Custom domain	–	Supported	Supported	Supported	Supported	Supported
Auto Scale	–	–	–	Supported	Supported	Supported
VPN hybrid connectivity	–	–	–	Supported	Supported	Supported
Network Isolation						Supported

App Services Plan Infrastructure

- IIS:
 - Windows Server 2016
- Tomcat:
 - Debian Linux Docker container
 - Marketplace or uploaded container images
- The virtual machines:
 - Free-Standard tiers: Standard A-Series VMs
 - Premium V2 & Isolated tiers: DS_v2 with Premium SSD disks

Auto-Scaling

- Two ways to scale:
 - Scale up: Increase/decrease the size of each instance (VM)
 - Scale out (more interesting): Increase/decrease the quantity of load balanced instances
 - Pay per minute for used instances
- Code is stored in persistent backend storage
 - Instantly available to new instances
- Auto-scaling
 - Scheduled
 - Rule/metrics-based

Multiple App Services Plans

- You might run many plans:
 - Dev-Basic
 - Prod-Standard-S2
 - Prod-Isolated-I1
- App Services can move between plans:
 - Same region
 - Destination plan must support the features used in the app service

Demo

- Creating an App Service Plan
- Scaling Out
- Scaling Up

Azure DB

Just a Taster

- Storage accounts: Table (NoSQL)
- SQL Server:
 - Managed Instance
 - Azure SQL
- MySQL:
 - App Services SQL In-App
 - Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Database for MariaDB
- Cosmos DB
- Redis Cache

App Service Basic Features


Basic App Service Attributes


- URL
 - Free Azure-managed DNS Name
 - Globally unique prefix: *mywebsite.azurewebsites.net*
- Outbound IP Addresses:
 - External interfaces of the front ends
 - Dynamic – so don't rely on them
- Deployment Trigger URL:
 - Webhook used by external code management systems
- FTP & FTPS
 - Username, host name, diagnostics logs FTP path



Application Settings


- Similar to IIS site settings!
- General Settings:
 - .NET Framework version
 - PHP version / Off
 - Java version / Off
 - Python version / Off
 - Platform: 32-bit / 64-bit
 - Web Sockets
 - Always on
 - Managed pipeline version
 - HTTP Version
 - ARR affinity
 - And many more
- Debugging
 - Remote debugging
 - Remove Visual Studio Version


General settings


.NET Framework version  v4.7


PHP version  5.6


 App Service supports installing newer versions of Python. Click here to learn more. 


Python version  Off


Java version  Off

Java minor version 

Java web container 



Platform  32-bit 64-bit

Web sockets  Off On


Always On  Off On

Managed Pipeline Version Integrated Classic

HTTP Version 1.1 2.0



 You can improve the performance of your stateless applications by turning off the Affinity Cookie, stateful applications should keep the Affinity Cookie turned on for increased compatibility. Click to learn more. 

ARR Affinity Off On

 Auto swap destinations cannot be configured from production slot

Auto Swap Off On

Auto Swap Slot

 FTP based deployment can be disabled or configured to accept FTP (plain text) or FTPS (secure) connections. Click to learn more. 

FTP access FTP + FTPS FTPS Only Disable

Custom Domain Names

- *Shared tier +*
- You can associate your own domain name
 - Purchase this *outside* of Azure/Microsoft
- Validation steps in the DNS zone:
 - A record -> IP address of the app service (plan)
 - TXT -> .azurewebsites.net name
 - CNAME -> .azurewebsites.net name



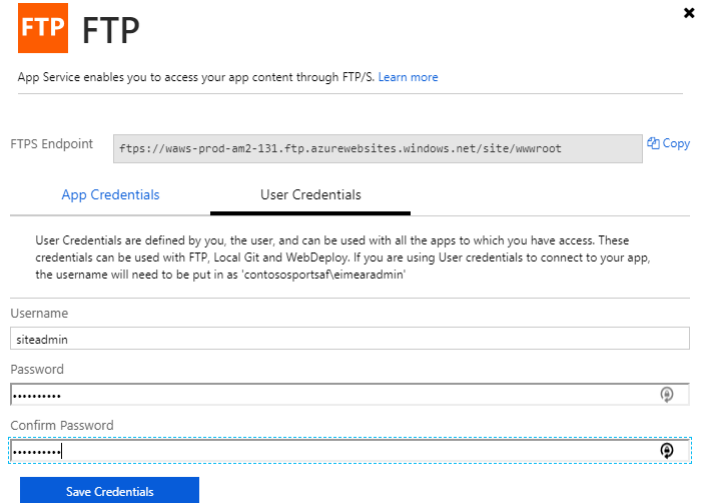
SSL / TLS

- *Basic tier +*
- Types:
 - SNI-Based: Supports multiple certs – supported by most modern browsers
 - IP-Based: Requires dedicated IPv4 per app service at extra cost.
- Requirements:
 - The domain name is associated with the app service
 - Trusted authority issuer
 - Password-protected PFX file
 - 2048-bit private key
 - PFX contains any intermediate certificates in the chain



FTP

- For the dinosaurs out there
 1. Configure user credentials – Deployment Center
 2. Get FTP server name – Properties
- Supports
 - FTP
 - FTP/S
- Disable/configure in Application Settings



The screenshot shows the 'FTP' configuration page in the Azure portal. At the top, there's a header with the 'FTP' logo and a close button. Below the header, a message states: 'App Service enables you to access your app content through FTP/S. [Learn more](#)'. The 'FTPS Endpoint' is displayed as 'ftps://waws-prod-am2-131.ftp.azurewebsites.windows.net/site/wwwroot' with a 'Copy' button. Two tabs are visible: 'App Credentials' and 'User Credentials', with the latter being the active tab. A descriptive text explains that user credentials are defined by the user and can be used with various apps. Below this, there are three input fields: 'Username' (containing 'siteadmin'), 'Password' (masked with dots), and 'Confirm Password' (also masked with dots). A 'Save Credentials' button is located at the bottom of the form.

FTP

App Service enables you to access your app content through FTP/S. [Learn more](#)

FTPS Endpoint [Copy](#)

`ftps://waws-prod-am2-131.ftp.azurewebsites.windows.net/site/wwwroot`

App Credentials User Credentials

User Credentials are defined by you, the user, and can be used with all the apps to which you have access. These credentials can be used with FTP, Local Git and WebDeploy. If you are using User credentials to connect to your app, the username will need to be put in as 'contosopostsaflieimearadmin'

Username

siteadmin

Password

Confirm Password

Save Credentials

Backup

- *Standard tiers +*
- Backup the app service to blob storage (storage account)
 - ZIP file
- Automatic (scheduled) and manual
- Optionally include linked databases (app service connection strings)
- In preview:
 - *Premium tier*
 - Snapshots – incremental and not affected by file locks



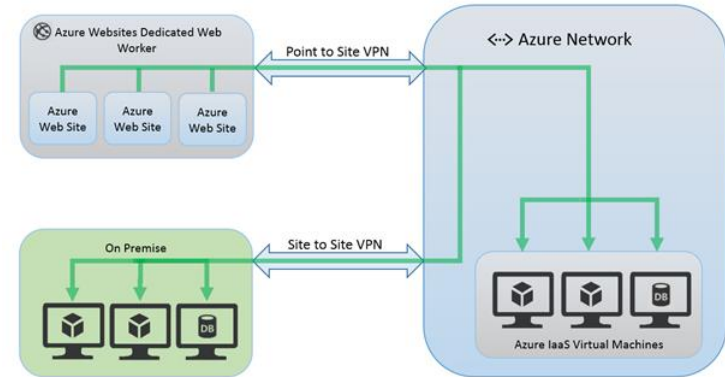
Networking

Typical Network Layout

- *Free - Premium tiers*
- Shared front ends
 - Layer-7 load balancing
- Shared network
 - Microsoft manages the network security
 - More options later in this session
- No direct connections to your other infrastructure (anywhere)

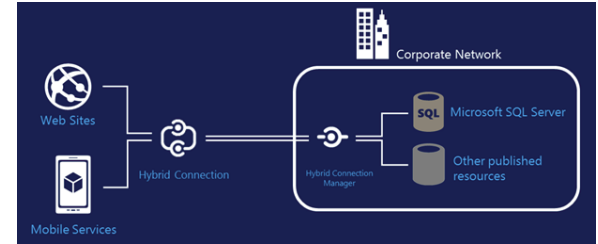
Point-to-Site VPN

- *Standard tier +*
- Enable an app service to connect to Azure virtual machines
 - And the services running on them
- Architecture:
 - Deploy a VPN Gateway (route-based Basic, or higher)
 - Configure Point-to-Site VPN
 - Connect App Service to VPN
- Limits:
 - 5 connections per App Service Plan
 - No drive mounting, AD integration, NetBIOS, private site access



Hybrid Connections

- *Basic tier +*
- Connect to services inside/outside of Azure
- Built on Service Bus Relay with TLS 1.2 & SAS keys
- Supports “any on-premises resource that uses a static TCP port”
- Benefits:
 - Secure access to on-premises services
 - Does not require Internet accessible endpoint
 - Quick & easy to set up
 - No inbound firewall rules – outbound connections only
 - Agnostic to development languages



Virtual Network Integration

- *Limited preview*
- Connect an App Service to a VNet
- Doesn't require:
 - Hybrid connections
 - Point-to-site VPN

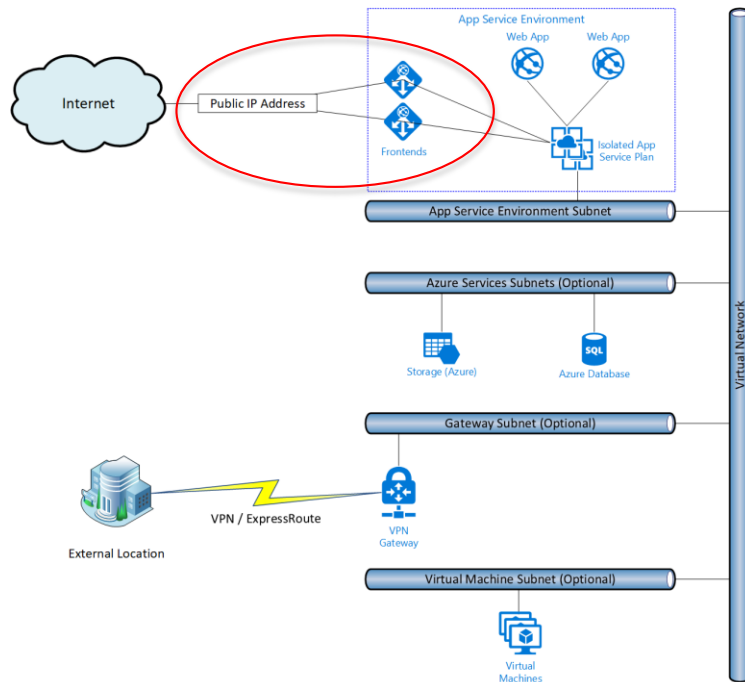
Isolated Tier

App Service Environment (ASE, pr. “Ace”)

- Major differences with public App Service Plans:
 - ASE is deployed into a virtual network
 - Dedicated front ends
 - You control the public IP address (PIP)
 - You control the network security grounds (NSGs / firewalls)
 - Allows for huge scale-out (100 instances/workers)
 - Larger specs of instances/workers
- Pricing:
 - A flat fee for the ASE ~€954.16/month
 - Instance cost for the App Service Plan

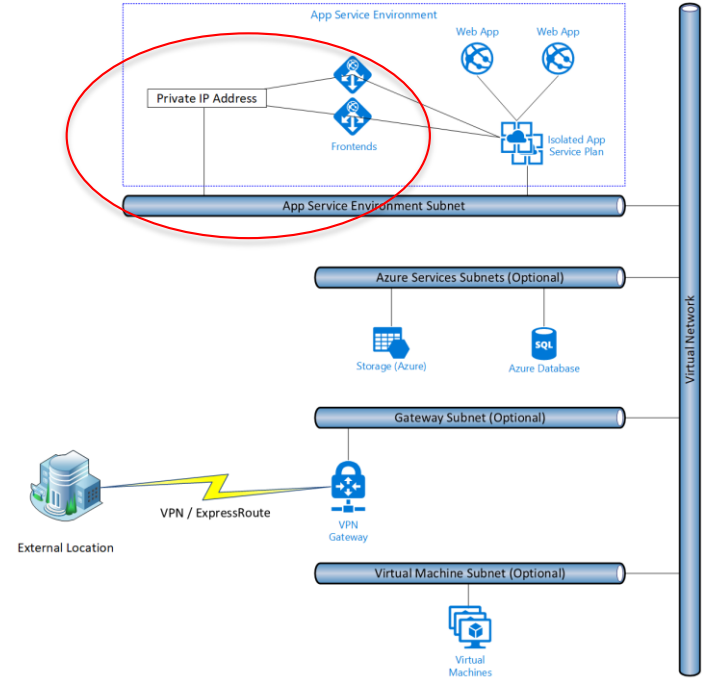
ASE – Public IP

- ASE is connected to Virtual Network
 - Can use S2S VPN
 - ExpressRoute
- Can connect to VMs across the VNet
- ASE presented to the world via a VIP/PIP
 - Source/destination of all ASE traffic with Internet



ASE – Internal Load Balancer

- ASE is connected to Virtual Network
 - Can use S2S VPN
 - Expressroute
- Can connect to VMs across the VNet
- All traffic to/from ASE via private IP address
- ASE not directly accessible from the Internet



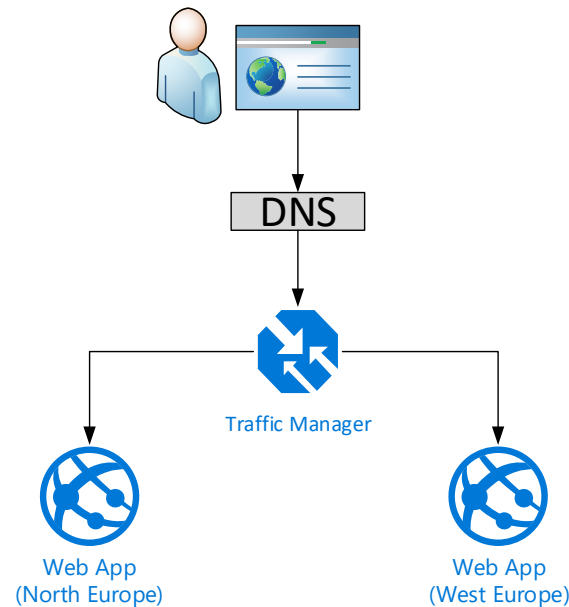
External Connectivity

Web Application Gateway

- Layer-7 load balancing
 - Use an App Service as a “backend pool”
 - URL redirection
 - SSL offload
 - And more ...
- Web Application Firewall
 - Layer-7 security filtering (OWASP 3.0)
 - Can be used with Free-Premium
 - Seen with ASE (Internal Load Balancer) for PCI-DSS blueprints, etc

Traffic Manager

- DNS redirection
 - Load balancing
 - Geographic direction
 - Failover (DR)
 - Scale-out
- Functions:
 - Unify multiple installations across regions
 - Blue/Green deployments
- Automatic or manual failover (DR)
 - I prefer manual via Azure Automation



NIC

Accelerated Networking

- Content Delivery Network (CDN)
 - Azure CDN
 - Akamai/Verizon via Azure
 - Third-party such as CloudFlare, Infracore, etc
- Azure Front Door
 - Reduced client/service latency
 - Clients enter Azure WAN closer to home
 - 100+ global entry points to world's 2nd largest WAN
 - Running for 5+ years for Xbox and Office 365

Code Deployment

Dinosaur Hunting

- Yes, you can FTP or FTPS code
- That's dumb
- Modern devs use tools like Github and Azure DevOps (VSTS)
- Integrate with those tools

Code Integration

- Store your code in management solution:
 - Visual Studio Team Services (VSTS)
 - GitHub, BitBucket, Local Git repository, Online external repository (Git or Mercurial)
 - OneDrive, DropBox
- Synchronize your code into an App Service
 - Simple “Sync” button
 - Can be automatic using a webhook
 - Triggered by a “release” in code management solution

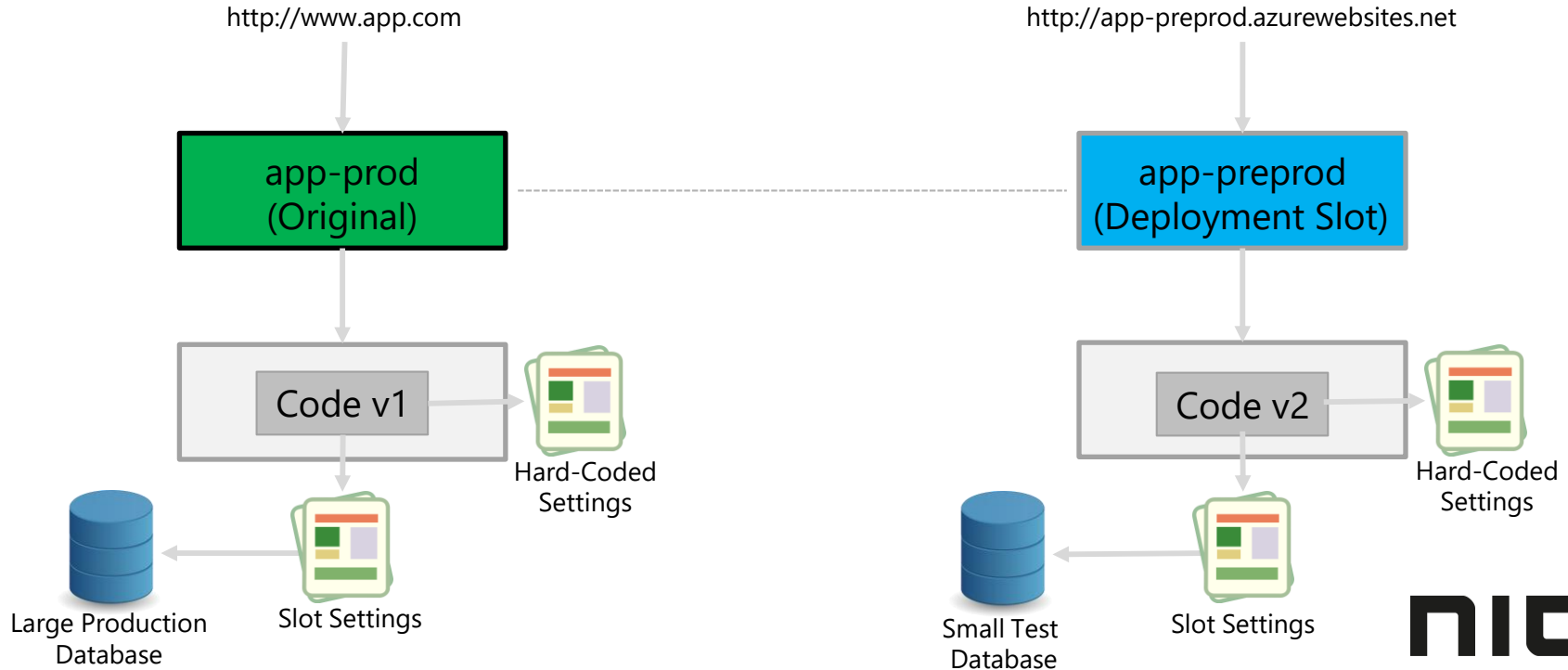
Application and Database Connection Strings

- Application strings:
 - Store a value for an application outside of the code
 - E.g. “Dev”, “Test”, “Production”
 - Code can call that value in
- Database Connection String
 - Don’t hardcode database connections
 - Store the value outside of the code
 - Code can call the string in

Deployment Slots

- Spawn a deployment slot instance from an App Service
- Example:
 - Production app service
 - Pre-production deployment slot
- The deployment slot:
 - It's own app service – linked to parent
 - Running on the same app service plan
 - No extra cost – you pay for the app service plan instances
 - Has a deployment link with the original app service
 - Has it's own URL

Deployment Slots & Settings



Demo

- Deployment Slots
- Code management with GitHub

For the Devs

- App Services make life easy for everyone:
 - Fewer servers for Ops to manage
 - Faster deployment for Dev
- But there are also more tools for devs

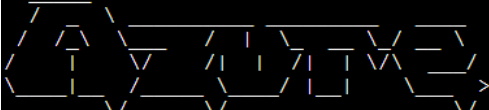
Application Insights

- Additional cost
 - Plan + additional monitoring data
- Operates at two levels
- App Service Plan extension
 - “Run time” monitoring
 - No web app code changes required
 - Response times, success rates, exceptions, dependencies, etc.
- Add code to the web app
 - “Build time”
 - Code-level monitoring, alerting & reporting
 - Client-side, server-side, and custom telemetry



Command Line Access

- You cannot RDP into the web server
- You can get command-line access to the web app
 - SSH for Linux
 - CMD.EXE for Windows
- Accessed via the Azure Portal > web app
- Intended for basic file manipulation

```
Last login: Mon Jan 8 11:01:27 2018 from 172.19.0.2  
  
A P P S E R V I C E O N L I N U X  
  
Documentation: http://aka.ms/webapp-linux  
PHP quickstart: https://aka.ms/php-qs  
  
-bash: export: `g++': not a valid identifier  
-bash: export: `libc-dev': not a valid identifier  
-bash: export: `pkg-config': not a valid identifier  
root@04bcf737e6d9:/home/site/wwwroot# cd site/wwwroot  
root@04bcf737e6d9:/home/site/wwwroot# ls  
hostingstart.html  
root@04bcf737e6d9:/home/site/wwwroot# █
```

Kudu

- Console full of useful tools
- Process troubleshooting
- Diagnostics
- Runtime data

The End



Thank you!

Aidan Finn

- <http://aidanfinn.com>
- <http://petri.com>
- @joe_elway

INNOFACTOR[®]

NIC