



# **Azure Services** Every .NET Developer Needs to Know

Chad Green

January 17, 2019



## **Chad Green**

Data & Solutions Architect at ProgressiveHealth

### Community Involvement

Code PaLOUsa Conference Chair

Louisville .NET Meetup Organizer

Louisville Tech Leaders Meetup Organizer

Louisville Tech Ladies Co-Organizer

### Contact Information

✉ [chadgreen@chadgreen.com](mailto:chadgreen@chadgreen.com)

🌐 [chadgreen.com](http://chadgreen.com)

🐦 [@ChadGreen](https://twitter.com/ChadGreen)

🌐 [ChadwickEGreen](https://www.linkedin.com/in/ChadwickEGreen)

# Azure Terminology

**Account** – Your Azure account is the credentials that you sign into Azure with (e.g. what you would use to log into the Azure Portal)

**Subscription** – The billing plan that Azure resources are created inside. These can either be individual or managed by your company. Your account can be associated with multiple subscriptions.

# Azure Terminology

**Resource** – Any service created in Azure

**Resource Group** – Groups Azure resources.  
All resources must be placed into a resource group.

**Provisioning** – Creating Azure resources.

# Azure Terminology – Hosting vs Service

## Hosting

You provide the code  
Azure runs the code you wrote  
e.g. Running an ASP.NET app

## Service

You provide data/information  
Azure's implementation takes  
action on what was provided  
e.g. Blob storage

## Service #1 – App Service

- Web Apps, Mobile Apps, API Apps, Logic Apps
  - Great for front-end Web apps and Web APIs
- Fully managed: OS patching, High Availability, Scaling
- Diagnostics & monitoring integration
- Deployment slots
- First class Visual Studio integration

## Service #2 – SQL Database

- Built from the same code as SQL Server
  - Updates available in Azure before SQL Server
- Managed: Automatic patching, high availability, and scalability
- Extensive monitoring and altering capabilities
- Integration with Visual Studio

## Service #3 – Azure Storage

- Managed, Scalable, and Secure data storage
  - Blobs – object and file storage via an HTTPS URL
  - Tables – NoSQL key/value data
  - Queues – Messages and work items
  - Files – Mountable file shares
- Automatically replicated and backed up
- Optional geo-replication
- X-Archive, Cold Storage, Soft Delete, Snapshot, and Restore



## Service #4 – Azure Functions

- Serverless
  - Never create or manage VMs or clusters
  - Auto-scales automatically
- Event driven execution off of HTTP, Queues, Timer, etc.
  - Great for simple jobs and workers
- Pay only for CPU and memory used during execution
- First class integration with Visual Studio

## Service #5 – Azure Monitor

- Built in monitoring and diagnostics for failures and slow performance
- Rich diagnostic data to root cause issues
- Collects metrics, logs, requests, dependences, correlation IDs
- Distributed E2E tracing
- Snapshot debugging in production
- Profiler for performance issues

# Honorable Mentions (if I had more time)

- **Hosting**

- Azure Kubernetes Service – Kubernetes and Container Orchestration
- Azure Service Fabric (Mesh) – distributed microservice apps
- App Service Linux

- **Services**

- Key Vault – store secrets and read out using managed identity or certificate
- Service Bus – more sophisticated queues, topic routing
- Azure SQL Managed Instance – dedicated, full control, lift and shift
- Azure Cosmos DB – Modern database designed for the cloud
- Azure Active Directory – Active Directory support in Azure

## Tools Shown

- Azure Portal
- Visual Studio
- Storage Explorer

# Summary

- Azure Service for Web and API hosting
- SQL Database for storing relational data
- Storage for objects, files, queues
- Functions for simple backend jobs and triggers
- Azure Monitor for application monitoring and diagnostics

thank you.

 chadgreen@chadgreen.com

 chadgreen.com

 @ChadGreen

 ChadwickEGreen