Understanding Azure

A GUIDE FOR DEVELOPERS

Microsoft

Abstract

There is no time better to be a developer. With the cloud, you can compose solutions that were never possible before. You can release new features to millions of users within minutes. You can push the boundaries of current technological limitations in days. Developers can turn ideas from the ground up to successful businesses in only months. Companies need apps that allow them to maximize customer engagement and differentiate against competitors. Teams must have agility with app development for faster time to market. Developers need a flexible platform to scale up and down based on business demands, yet rock-solid resources that can withstand failure. Microsoft Azure offers a cloud platform designed for developers to build the most innovative apps.

This guide breaks down the "why" and "how" for scenarios suited to the cloud with a focus on building apps using platform services available in Microsoft Azure. The second half of the guide showcases the breadth and depth of the Azure platform and services designed to help developers make truly competitive and differentiated applications.

The intended audience for this guide includes:

- Developers wanting to understand why Azure is the best cloud platform for creating applications and how to get started quickly based on the app you want to build today.
- Technical leaders considering Azure to support new or existing application development.

PUBLISHED BY
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Table of Contents

Overview

Introduction

The cloud is changing expectations. Your customers expect more. Your business expects more. You expect more.

Companies you already know, such as Uber and Facebook, were born in the cloud, and almost every industry sector is adopting the cloud to drive business growth. These companies move faster, deliver more value, and meet our ever-changing needs more effectively than those constrained by more traditional approaches. They find new ways to use the flexibility of the cloud to their advantage. They design ways to almost infinitely scale out and gain deeper insights into their customers that other companies can only dream about.

Now you want to move to the cloud, too. And you're wondering how to go about it. You want the benefits, but where do you start and how do you avoid the pitfalls that cloud pioneers faced?

British computer scientist David Wheeler famously said, "All problems in computer science can be solved by another layer of indirection." And indeed, once companies figured out how to massively scale while controlling costs, it was just a matter of time before that layer would come to cloud computing. That layer is the application platform—and it's here today.

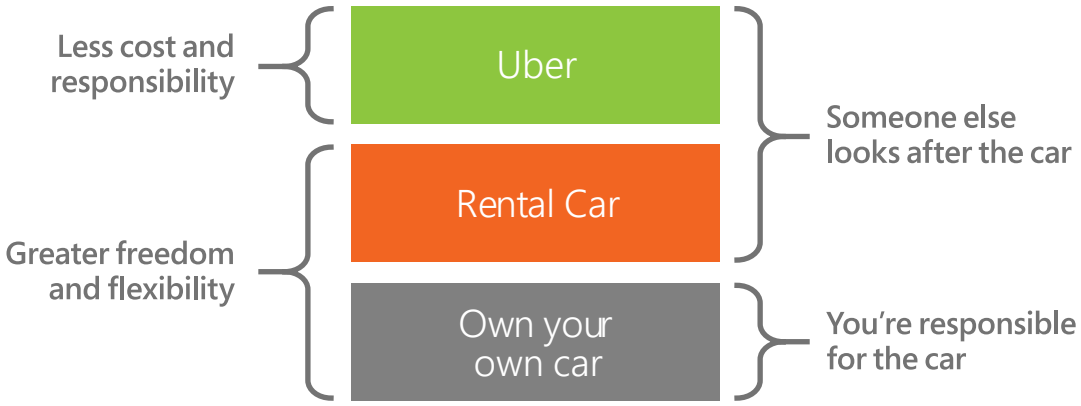
You can still control the maintenance, scaling, and redundancy of your application across a farm of commodity servers that expands and contracts to suit your needs. But now you can deploy on a platform that handles all that for you. This is the "second wave" of cloud computing— where you focus on delivering the innovation your customers demand, not the infrastructure required to run it.

This guide is an introduction to the Azure application platform. It will provide the guidance and direction you need to start building new applications or moving your existing applications to the cloud.

The Changing World of App Development

In the beginning, there were physical machines. They were expensive and developers loaded them up with multiple applications to save costs. This caused no end of conflicts and unexpected bugs, yet effective machine utilization was still low. Developers struggled to make sure changes to one application didn't break an unrelated application.

Then came virtual machines. Since multiple virtual machines could be run on a single physical box, developers installed their applications in contained virtual machines, which reduced conflicts and improved utilization, but welcome to



shared memory and unpredictable performance. And since provision had to be made for the largest loads, "scaling in" wouldn't really save money.

Enter the cloud. Suddenly, companies could rent the virtual machines they needed, when they needed them. They could scale applications up and down on demand. You still needed to install and configure OS level patches and updates, and you still had to contend with low-level networking. Still, hundreds of companies adopted infrastructure as a service (IaaS) and successfully leveraged the flexibility to dramatically improve speed to market and scalability while controlling costs. But most of the benefits of this model accrued to operations and infrastructure, not app development. As more companies realize that app innovation needs to move faster than ever because of customer and competitive demand, focus is shifting from traditional infrastructure cost savings to how to make app development more productive.

Azure—the Cloud Platform Designed for You, the Developer

The history of Microsoft is a history of developers. No other cloud provider is as steeped in development and developers as Microsoft. A BASIC interpreter was the very *first* Microsoft product back in 1975, and we have relentlessly focused on the needs of developers since. We don't just build the world's best platforms and developer tools, we use them ourselves to build software and services for our customers. And we didn't just build the world's most developer-friendly cloud, we use it ourselves. We are a cloud-first company, focusing on building next gen products and services for the cloud. Our services need to be always on, scale to millions of users, and update seamlessly with new capabilities. Just like our customers, we need an application platform to work hard for us. We designed Azure to accelerate developer productivity.

So what differentiates a developer platform from an infrastructure platform?

It's pretty simple, there are tiers of responsibility with benefits and tradeoffs between them. For example, to use a car analogy, if you own a car you have to go through the purchase process, buy insurance, maintain and service it over a long period of time, and provide parking or garaging.

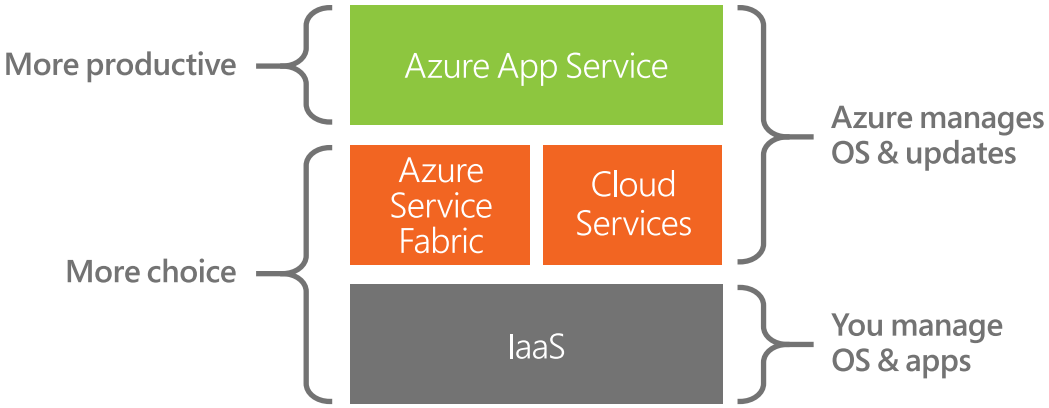
If you're renting a car, you get the benefit of a car at your disposal for a fixed time period without any of the overhead of ownership. And, of course, an Uber or a taxi is likely to be the most affordable option, but can have limits on flexibility—

Alaska®

Alaska Airlines needed to make an internal web app to book standby travel available to mobile employees outside of the corporate firewall. Developers took the existing logic and moved the application to Azure App Service, created a mobile front end with Xamarin, and made the app available to employees in record time.

More about Alaska and Azure

Overview



you might have to wait five minutes, or maybe stand out in the rain waiting for a ride.

You can get more from your scarce developer time by using an application platform that removes complexity and maintenance responsibility. There might be some tradeoff considerations around flexibility, but overall, you can get more done faster and with lower total cost of ownership building on platform as a service (PaaS).

(There are still some situations where it makes sense to build apps on IaaS that we'll discuss later.)

Let's get down to the details with a short tour through the main Azure platform services, explain where and when you might use them, then we'll guide you through some common customer app scenarios that will help you make the best design decisions.

App Service—Web and Mobile Apps

Azure App Service is a set of services that has everything you need to build apps that target both web and mobile clients from a single app back end. Supporting a range of language options—.NET, Node.js, Java, PHP, and Python—it consists of Web Apps, Mobile Apps, Logic Apps, and API Apps. Web Apps and Mobile Apps provide back-end services for web and mobile applications. API Apps allow you expose APIs that can be securely consumed by any application and connect your app to dozens of popular services, like Office 365 and Salesforce.com. And Logic Apps let you automate business processes and coordinate workflows using tools ranging from a no-code experience to the full power of Visual Studio. Fully integrated DevOps allow you to deploy app updates with built-in staging, roll-back, testing-in-production, and performance testing capabilities. You can monitor all aspects of your apps in real time and historically with detailed operational logs. App Service handles the underlying infrastructure, removing the need for you to maintain and patch your infrastructure.

When to use it

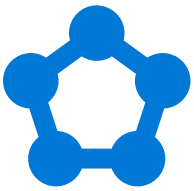
App Service is the preferred option for building web or mobile back ends because it provides the productivity, scale, performance, and deployment options for most requirements. Use App Service when you want to focus on building unique features for your app and need the infrastructure to just work. Additionally, it's great for building an API layer which can be exposed to customers, partners and employees securely using the related API Management service. You can use App Service in conjunction with the many other Azure services (Redis Cache or Azure Traffic Manager, for instance) to quickly build and deploy powerful apps.

jet

Jet.com needed to innovate extremely rapidly to build an effective e-commerce platform. Starting from a blank canvas they could choose any platform, and thanks to Microsoft Azure and its rich application platform services, the company was able to release in 12 months instead of two years.

More about Jet.com and Azure

Service Fabric—Microservice Based Apps



Azure Service Fabric is a platform that developers and ISVs can use to build and manage custom microservice-based applications at cloud scale and with 24x7 availability. Microservices are an architectural approach based on fine-grained, loosely coupled services that can enable advanced developer agility, cloud scale, self-healing resilience and always on availability. Service Fabric is the foundation for many Microsoft services such as Azure SQL Database, Azure Document DB, Cortana, Windows Intune, and Skype for Business. We've taken the exact same technology and made it available to you. And use Service Fabric wherever you want to run your microservice based apps—whether in Azure, in your own datacenter or in other clouds—giving you maximum flexibility.

When to use it

The primary Service Fabric scenario is for building highly scalable, always on services that can be updated with zero downtime (like the Microsoft first party services mentioned previously). Use Service Fabric when you are creating a new app or re-writing an existing app to leverage microservices, have committed to a cloud-first architecture, and require rolling version updates with no downtime, distributed scalability and high performance, and low-latency data read and write.

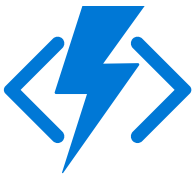
Cloud Services – Monolithic Apps

Azure Cloud Services was the original Microsoft platform as a service (PaaS) service launched in 2010. Inspired by the needs of developers, Cloud Services enabled them to easily deploy .NET applications, taking advantage of cloud infrastructure like Virtual Machines and Virtual Networks without having to manage the low-level details and configuration of Windows Server and the underlying infrastructure.

When to use it

Cloud Services is an effective solution when you need direct access to the underlying infrastructure or need to configure Internet Information Services (IIS) directly. Like App Service, you don't need to deal with patching or updating the underlying OS. Cloud Services is sometimes used to migrate existing web apps with dependencies on IIS configurations or required third party components—it provides autoscaling while still allowing for many developer-focused platform benefits. Since Cloud Services was introduced, design options have evolved significantly. App Service and Service Fabric reflect a more modern approach to PaaS that you should consider as the foundation for your application development.

Azure Functions—Event Driven Execution



Azure Functions provides serverless, compute for event-driven solutions that extends the existing Azure application platform with capabilities to implement code triggered by events occurring in other Azure services, SaaS products (like Office365 and Salesforce.com), and on-premises systems. With Functions, you only pay while your function is actually executing. It provides an intuitive, browser-based user interface allowing you to create scheduled or triggered pieces of code implemented in a variety of programming languages.

When to use it

When you want to create independent, unattended functions that respond to events across Azure, SaaS products and your on-premises services. Use them to add housekeeping or value-add features to your application without directly requiring a modification of your codebase. For instance, if your family-friendly site allows people to post their own photos, you could add an Azure Function that listens to the storage area for new photos and uses some of the advanced Azure APIs to identify images that contain inappropriate

Containers

Containers are top of mind for developers, enabling you to run your apps anywhere with agility and efficiency. Microsoft is working closely with partners like Docker and Mesosphere to offer solutions in this space. Containers are often use in the IaaS layer but are also popular for implementing microservices based solutions like Service Fabric. Read more about Docker and Containers and Azure Container Service in this guide.

content to replace the photo with a redacted version. All this can be achieved with no code changes at all to your existing application—it can be deployed completely independently with no impact on the existing application.

Summary

As a developer, you're always curious about new technology. You want to experiment with new and interesting ideas to make your apps better or create new apps. Previously, you had to do much of the heavy lifting yourself—writing your own capabilities or recreating a feature or service for every app you wrote.

But now you can implement modern solutions in a few lines of code. In addition to a rich application platform, Azure has powerful services like machine learning, mobile engagement and analytics, and notifications ready for you to plug into your app and be on your way. You can always write your own mobile notification service, but why not take advantage of pre-built platform services and spend your time on the features that make your app great?

On Azure, you'll find open source and cross-platform support across the broadest selection of programming languages, frameworks, tools, databases and architectures allowing you to realize the maximum reach for every line of code. Write once, run anywhere has been an industry mantra forever, but with Azure you have unparalleled options through products like Azure Stack that allow you to run your code in more places, unchanged. You bring the tools you love and skills you already have, and run virtually any application, using your data source, with your operating system, on your device.

The modern developer is a *solution composer*, combining custom code with the best of finished services to create apps that were unimaginable not so long ago. And delivering those applications at amazing speed. Companies around the world are using Azure to build new apps and extend existing apps to support and transform their business. How can you be next?

TalkTalk

TalkTalk TV, the U.K.'s third largest cable TV provider, chose to redesign parts of their IaaS application to a take advantage of a microservices architecture with Azure Service Fabric. The new content management and resolution platform allowed them to achieve faster delivery cycles and maintain uptime during upgrades

More about TalkTalk and Azure

Common Cloud App Scenarios

You're ready to move to the cloud. You want the benefits, but are unsure where to start. The following scenarios are designed to give you a jumping-off point using common app workloads that many customers have.

Note: Be sure to download the latest version of the Azure SDK. Check out this link to find all available downloads.

Scenario 1: Building a Web App

Building a web app is maybe the most common app pattern. In this section we'll look at how to migrate, scale, and secure an existing web app.

A. Building a Simple Web App

Overview

You have an internal line-of-business application with a web front end and a relational database back end. The application doesn't need to scale to millions of users, but is a critical component of your business. Currently, it's accessed only inside the firewall, but you'd love it if people could reach into the application from outside the LAN while still maintaining good security principles.

The problem

Hosting an application yourself imposes very real, but somewhat hidden, costs. Some of these costs may include manually deploying updates, patching operating systems, and using VPN connections required outside of the firewall. The cost is not overwhelming and is likely hidden in the overall budgets and current assigned workloads. But that doesn't make it any less real.

The solution

For this type of application, moving to Azure is relatively easy, and the benefits can be large. When on-premises, developers may need to work with infrastructure teams whenever new versions of the application need to be rolled out to validate that the application won't interfere with other, potentially more critical applications. In Azure, developers can deploy updates earlier and more often using continuous deployment with on-premises and online version control systems like TFS, GitHub, BitBucket, and Visual Studio Team Services.

Services Used

- *Web Apps*
- *SQL Database*
- *Azure Active Directory*

Common Cloud App Scenarios

Patching operating systems is a necessary task. Poorly managed infrastructure can have serious security consequences or software implications. In Azure, the machines that sit under Azure Web Apps are automatically patched and managed for you.

Accessing company websites that require authentication outside of a firewall requires some form of a VPN connection. In Azure, integrating Azure Active Directory with your on-premises Active Directory can enable users to connect with their same credentials and even use multi-factor authentication without the need for a VPN connection.

The how

Instinctively, you might consider shifting your virtual machines into Azure, setting up SQL Server, and configuring IIS. It may take more time for you to configure everything just the way you want, but you have the control to do so. Your team will need to be in charge of patching, maintenance, and administration. This lift-and-shift model gives you the same functionality that you would expect from on-premises deployments. It may even be the fastest way to get to the cloud. But the cost and resource requirements are probably similar to your on-premises solution, just hosted in the cloud.

Instead, you can bypass virtual machines and use an Azure Web App in App Service to host your web app, Azure SQL Database for your database, and secure your web app by using Azure Active Directory. This means you can take advantage of continuous deployment options to get up and running faster and maintain your app over time.

Hosting a web app in the cloud

Whether your app is written in .NET, Java, PHP, Node.js, or Python, you can deploy your web app to an Azure Web App in App Service. You can manage multiple Web, Mobile, API, and Logic Apps in App Service. For most web application needs, App Service is the answer.

- Deploy, copy, or migrate your web app to an Azure Web App.
- Upload an SSL certificate and configure a custom domain name to the Web App.
- Deploy to the Web App repo continuously through BitBucket, TFS, GitHub, or Mercurial.

You can set up Azure Active Directory to sync with your organization's on-premises Active Directory and connect users to the web app through Windows authentication outside of the firewall.

- Download and install Azure AD Connect to integrate on-premises identities.
- Configure features of Azure AD Connect, such as filtering or password synchronization.
- Activate directory sync for Azure Active Directory in the portal.
- Enable authentication with Azure Active Directory for the Web App and Azure SQL Database in the Azure Management Portal.

Storing data in the cloud

Azure SQL Database is a managed relational SQL database-as-a-service (DBaaS) solution in Azure – and distinct from SQL Server running on Azure Virtual Machines. You don't have to manage any virtual machines, operating systems, database software, or worry about upgrades, high availability, and backups. In general, Azure SQL Database can dramatically increase the number of databases managed by a single IT or development resource.

- Deploy to SQL Database using SQL Server Management

Pro Tip

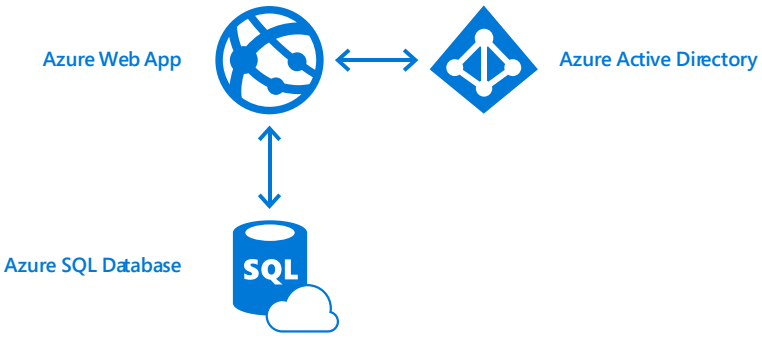
Implement staged deployment to swap staging and production slots of your web apps with zero downtime, and run A/B testing in production with a configurable fraction of your live traffic.

Common Cloud App Scenarios

Studio, export the database to a BACPAC file then import or use BCP, or use SQL Server transaction replication with minimal downtime.

- Once the database is up in Azure, connect to it the same way you did on-premises through SQL Server Management Studio.
- Copy the connection string and place it into the connection strings area for the Web App in Azure or other resources that connect to it.

Architecture diagram



Additional resources

Deploy your app to Azure App Service

Continuous delivery to Azure using Visual Studio Team Services

Integrating your on-premises identities with Azure Active Directory

SQL Server database migration to SQL Database in the cloud

B. Scaling a web app

Creating a simple web app in Azure is easy enough, but any growing company wants to be able to scale and provide global reach for their web apps.

The solution

To scale your Azure Web App in App Service, you only need to slide a horizontal bar to increase the instance count of the web apps available. But you also want a consistent experience for all users, whether your customer is at an Internet café in Rabat or in a hotel in Mexico City. Choose an Azure datacenter where your web app and database get deployed, and Azure Traffic Manager will route users to the instance of your app closest to their location. Geographic expansion is a priority for Azure with support for new regions being added over time.

The how

One way to achieve scalability and availability is through virtual machines, load balancers and redundant resources. However, if you want your web app to always be available, you need to pay for the virtual machine and replicas to be always available.

In the same way that App Service removes the burden of managing infrastructure it also makes it easier to scale out. Use an Azure Web App to host your web app, the database in Azure SQL Database, and use Azure Traffic Manager to route users to copies of the web app globally. Enable Active Geo-Replication in SQL Database for synchronizing replicas of databases. Use Azure Redis Cache to improve performance by loading frequently accessed data and state information.

Scaling and replicating an application

Azure Web Apps allows you to scale the maximum number of instances of your app up and down with a slider bar. Set a schedule for scaling up/down (such as during business hours) or set a target CPU percentage and Azure will handle the rest. Then quickly deploy the same code to Web Apps around the world.

- Deploy, copy, or migrate your web app to an Azure Web App in App Service.
- Set scheduled times for auto-scaling the web app up or down or choose to scale by CPU load.
- Create at least one other Web App in another region.
- If pointing to another database, update the connection strings.

Azure Web Apps already provides failover and round-robin traffic routing functionality for Web Apps within a datacenter. Traffic Manager allows you to specify failover and round-robin traffic routing for Web Apps in different datacenters.

Routing global users to a web app

Azure Traffic Manager allows you to control the distribution of user traffic to your specified endpoints, such as Web Apps. The service applies an intelligent policy engine to Domain Name System (DNS) queries for the domain names of your Internet resources so that you can run your web apps in datacenters anywhere.

- Create a unique Traffic Manager profile and choose the load balancing method.
- In the Endpoints tab, add the Web App to point to.
- Set the monitoring settings for the Traffic Manager profile to ensure that the endpoints are available. You can specify the protocol, the port, and the relative path.

Services Used

- *Web Apps*
- *SQL Database*
- *Traffic Manager*
- *Azure Redis Cache*

- Point your company domain name to a Traffic Manager domain name.

Storing data across the world

Azure SQL Database is distinct from SQL Server in an Azure Virtual Machine because it provides Active Geo-Replication. Active Geo-Replication asynchronously replicates committed transactions from a database to up to four copies of the primary database on different servers. Use this feature for disaster recovery or use an online secondary readable database as a load balancer for read-only workloads serving clients distributed across several regions.

- Deploy your database to SQL Database using SQL Server Management Studio, export the database to a BACPAC file then import or use BCP, or use SQL Server transaction replication with minimal downtime.
- Scale to the Premium service tier to choose the location where the SQL Database is replicated.
- In Geo-Replication, add at least one secondary database, the target region, and target server.

Boosting an application's responsiveness

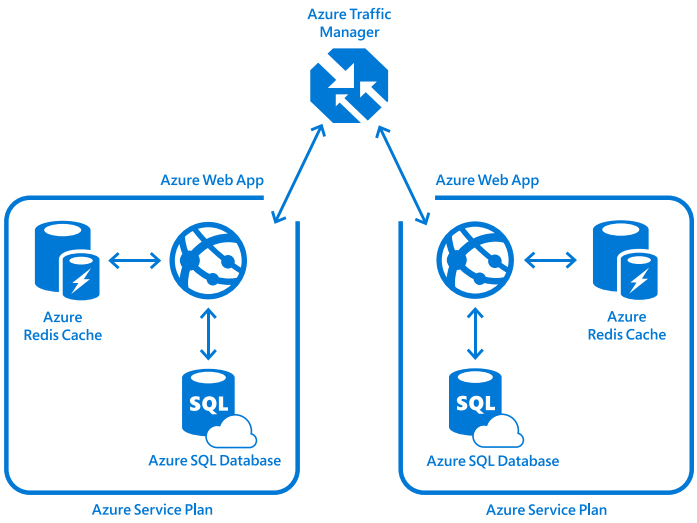
Azure Redis Cache is based on the popular open-source Redis cache, an advanced key-value store that lets you operate on values (strings, hashes, lists, sets, and sorted sets) and has master-subordinate replication and other performance-boosting features. This separate, distributed cache layer allows your data tier to scale independently for more efficient use of compute resources in your application layer.

- Create the cache and configure the cache client.
- Connect to the cache and store the credentials in the connection strings area in the Web App.
- Add and retrieve objects from the cache and specify expiration of items in the cache.
- Enable cache diagnostics to monitor the health of the cache.

Pro Tip

Learn how the viral site "How-Old.NET" was created and scaled with Traffic Manager in this Azure Friday session.

Architecture diagram



Additional resources

About Traffic Manager Monitoring

Active Geo-Replication for Azure SQL Database

How to use Azure Redis Cache

C. Isolating a web app

Many companies are faced with the requirement of securing apps that hold personally identifiable information (PII). A simple web app hosted publicly in Azure might not meet your security requirements. Instead, you need a locked-down environment with one door to the outside world and active defenses against potential threats.

The solution

If your organization hosts highly sensitive information, the number-one priority is having a fully-isolated and dedicated environment for only your organization's applications. Using an App Service Environment, your organization can have security and isolation for your web apps and use a virtual network for control over traffic. From a single open port, one option to block most traffic would be a product like Barracuda Web Application Firewall to protect your App Service Environment. Connect to on-premises resources with a site-to-site VPN or with Azure ExpressRoute.

The how

One option to host your application in Azure in an isolated way is to deploy the existing web app and database into an Azure Virtual Machine surrounded by an Azure Virtual Network. It would be fast to deploy, but you would need to patch and maintain those machines, likely the very same issues you have to deal with today.

Another option is to use an App Service Environment, surrounded with an Azure Virtual Network. Open one port to the public and use a product like Barracuda Web Application Firewall in front of the App Service Environment so that only approved users can access the apps in the environment. This option gives you the most flexibility in scaling up and down resources that are locked down in an isolated environment, while being able to take advantage of the rich features of App Service apps.

Isolating a web app in the cloud

An App Service Environment is a premium service plan option of Azure App Service that provides a fully isolated and dedicated environment. App Service Environments are isolated to run only a single customer's applications and are always deployed into an Azure Virtual Network. At a high level, an App Service Environment consists of compute resources running in the Azure Hosted Service, Storage, Database, a Virtual Network, and a subnet with the hosted service running in it.

To host an Azure Web App in an App Service Environment, create the App Service Environment first and then add the Web App into it.

- Create an App Service Environment and choose the Virtual Network, scale of the front end and worker pools, and specify the instance counts and number of IP addresses to use.
- Set up alerts for monitoring CPU and memory for the App Service Environment.
- Add or remove IP addresses to your App Service Environment for your apps to use.
- Specify auto-scaling individual worker pools based on metrics or schedule.
- Create a new Azure Web App and add it to an existing App Service plan or create a new plan in the App Service Environment.

Services Used

- *Web Apps*
- *App Service Environment*
- *A third party firewall*
- *Virtual Network*
- *Optional: ExpressRoute*

Common Cloud App Scenarios

- If creating a new App Service plan, select the App Service Environment and a worker pool.

Protecting a web app in the cloud

You can use a third party web application firewall (WAF) that helps secure your web apps by inspecting inbound web traffic to block malicious requests. Many of these firewalls also inspect the responses from the back-end web servers for data loss prevention (DLP). Combined with the isolation and additional scaling from App Service Environments, this provides an ideal environment for securing apps. The WAF is hosted on an Azure Virtual Machine, and you may consider deploying at least two instances of the virtual machines for redundancy and no single point of failure.

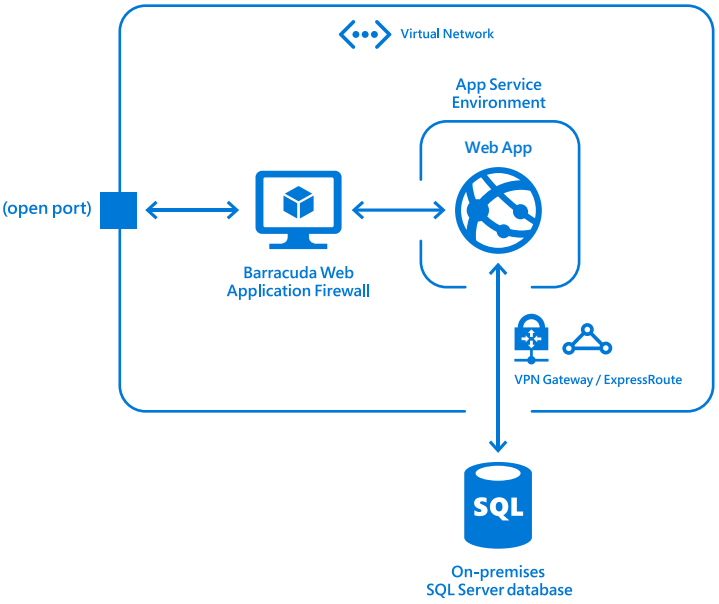
- Create and deploy the Barracuda Web Application Firewall (or similar firewall) and assign it a static IP address.
- Add HTTP and HTTPS endpoints that are used by your Azure Web App.
- Configure your firewall through its Management portal (adding and removing TCP/8000).
- Configure the firewall to protect your Web App.
- Create a network security group and assign it to a subnet in your Azure Virtual Network to restrict traffic to the App Service Environment from the WAF only by using the VIP address.

Accessing on-premises resources securely

To access on-premises resources, you can either use a site-to-site VPN with the Azure Virtual Network that surrounds the App Service Environment and the firewall, or you can add Azure ExpressRoute to the virtual network for a larger pipe to transfer data between resources.

- If using a site-to-site VPN, add a gateway subnet to the Azure Virtual Network, add your local site, request a public IP for the gateway, create the gateway IP addressing configuration, create the gateway, configure your VPN device, and create the VPN connection.
- If using ExpressRoute, create and modify a circuit and routing configuration, and link and configure a Virtual Network for ExpressRoute.

Architecture diagram



Additional resources

App Service Environment Overview

Barracuda Web Application Firewall on Azure Marketplace

Network Architecture Overview of App Service Environments

Pro Tip

You can use site-to-site, point-to-site, VNet-to-VNet, multi-site, and ExpressRoute connections cross-premises. Check out this link for the differences between them.

Scenario 2: Building a Mobile App Back end

In this section, learn how to extend your mobile app functionality with back end features and how to analyze user behavior.

A. Adding mobile features in your web app

Overview

You have an existing Azure Web App that connects to a storage back end. Your customers expect a mobile experience, so you need to build a mobile app. You would also like to have the mobile app share data and APIs with an existing Web App. Being able to send notifications to users through the mobile app would improve user engagement.

The problem

Building mobile client apps that target multiple platforms can be time consuming. Developers need to become familiar with various languages, platforms and IDEs , such as Xcode and Eclipse. You want a common way to add back end features like push notifications, offline data sync, and auto-scaling for those mobile apps as well. Creating a solution to expose your web app's API and reusing it in your native mobile app isn't a trivial task.

The solution

Microsoft provides comprehensive solutions for both front end and cloud-enabled back end development of cross-platform, mobile solutions. You can build native mobile app front ends in Objective-C, Swift, and Java, but if you're a .NET developer, you can use Xamarin to create mobile client apps in C# and share your client business logic across iOS, Android, and Windows Phone.

When creating mobile apps, you need to think about the client app functionality and user experience—and also adding shared back-end features like push notifications. You can add these pre-built mobile features and others with Mobile Apps in App Service. Hook in Azure Notification Hubs to send personalized push notifications to users. Then, share data with your Web App and use those back-end features for a complete mobile solution.

Create an API for your web app or, if you have an existing one, extract and deploy your API as an API App in App Service and share the API between the apps in the same App Service plan.

The how

When creating a mobile app, you're going to want to take advantage of services like auto-scaling and authentication. An effective approach is to create your mobile client app and connect it to an Azure Mobile App in App Service back end, tying in Notification Hubs for push notifications. Deploy your API as-is to API Apps in the same App Service plan, then share the API and data between both apps.

Creating a mobile app

Azure Mobile Apps is designed to allow you to build highly scalable, globally available mobile app back ends. Azure Mobile Apps supports native client apps as well as cross-platform Xamarin native and Cordova hybrid mobile client apps.

- Create a mobile client app (here's a Xamarin example) and connect it to your Azure Mobile App back end.
- Add offline data sync, allowing end-users to interact with the mobile app even without a network connection.

Services Used

- *Mobile Apps*
- *API Apps*
- *Xamarin*
- *Notification Hubs*

- In the Mobile area in Settings, choose the Storage account used by your Web App to add a data connection.
- Add auto-scaling properties, authentication (such as with Active Directory), Web Jobs, traffic routing, custom domains, and SSL to your Mobile App.

Sending push notifications to users

Azure Notification Hubs allows you to send cross-platform, personalized mobile push notifications with a single API call. Developers can easily integrate Notification Hubs into their apps.

Here's how to create a Notification Hub:

- Create a new Notification Hub and configure the mobile back end to point to that hub.
- Register your app to use push notifications and create an SSL certificate.
- Update the server project to send push notifications and add push notifications to the app.

Hosting an API in the cloud

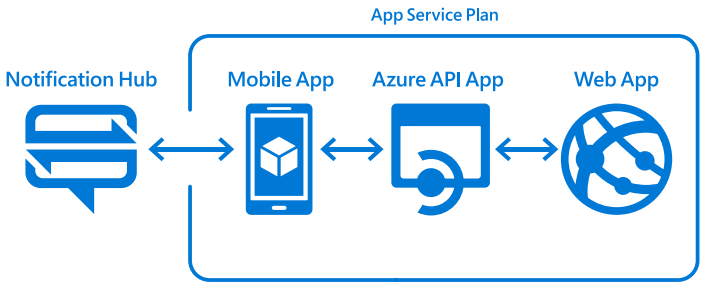
If you have existing APIs, deploy them as-is to API Apps in App Service and benefit from enterprise-grade security, simple access control, hybrid connectivity, automatic SDK generation, and integration with tools like Visual Studio. With support for Swagger API metadata, you can consume the APIs through a variety of clients and automatically generate client code. If you need additional access management capabilities, you can use Azure API Management to control client access to APIs hosted by App Service.

Pro Tip

Learn about connecting your mobile app with Azure in two minutes with this Channel 9 video.

- Download the latest Azure SDK for your language and tool of choice.
- Install the Swashbuckle package to work with Swagger API metadata.
- Publish your API (Visual Studio example) to a new API app in the same App Service plan as the existing Web App.
- Generate REST API client code through the Swagger URL.

Architecture diagram



Additional resources

I use Mobile Services: How does App Service help?

Enterprise push architecture for Notification Hubs

Get started with API Apps and ASP.NET in Azure App Service

B. Analyzing user behavior with your mobile back end

Once you've built a mobile back end, you need to know what to improve to keep your users happy. You want to receive fast feedback through real-time analytics of users' behavior, targeted push notifications for feature requests, and you might want to post tweets about your product to your internal #UserFeedback Slack channel in order to quickly change course and improve your apps based on user feedback.

The solution

Connect Azure Mobile Engagement with an Azure Mobile App back end to collect real-time user behavior analytics and send personalized notifications. Create a Logic App with a user-friendly designer that searches for particular tweet content and sends a message to Slack using a conditional statement.

The how

You could decide to build your own tools and utilities to provide analytics for your mobile app, and use custom code to connect to external resources like Twitter and Slack APIs. Instead of building these solutions yourself, you can use existing solutions in Azure so you can focus on improving the things that make your app unique.

Add Mobile Engagement to the Mobile App to collect analytics and send personalized notifications. Use Logic Apps in the same App Service to create if-this-then-that functionality for tweets and posting messages in a Slack channel.

Hosting your mobile app back end

Use Azure Mobile Apps to host the back end for your mobile apps in Azure.

- Create a new Azure Mobile App in App Service.
- Configure the server project.

Better engage users with analytics

Mobile Engagement is a software as a service (SaaS) user-engagement platform that provides data-driven insights into app usage, real-time user segmentation, and enables contextually-aware push notifications and in-app messaging. For example, you can create custom dashboards to measure key performance indicators (KPIs), rapidly find and fix usage bottlenecks in a user funnel path, track retention and user stickiness, and by that determine what campaigns are driving the highest return on investment. Mobile Engagement provides in-app messaging capabilities and works seamlessly with native push notifications gateways such as Google's GCM, Apple's APNS, and Microsoft MPNS.

- Create a new Mobile Engagement application.
- Copy the connection string for the Mobile Engagement app and connect your Mobile App to it.
- Enable and connect your Mobile App to real-time monitoring.
- Enable push notifications for the Mobile App with Mobile Engagement.

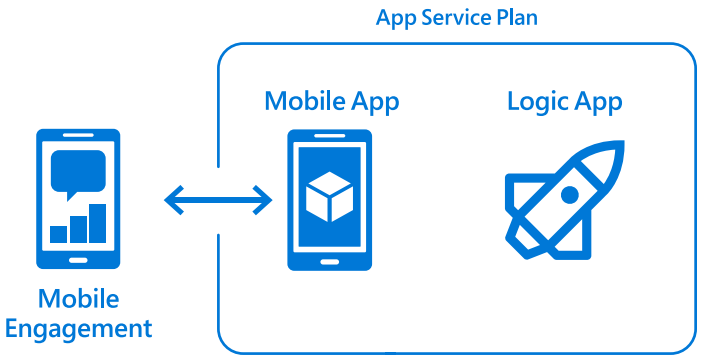
Connect data sources for events

Logic Apps allow developers to design workflows that start from a trigger and then execute a series of steps. Each step invokes an API while taking care of authentication, security, creating checkpoints, and durable execution.

- Create a new Logic App in the same App Service as your Mobile App.
- Create a new action to query tweet content related to your product and sign into Twitter.
- Create a conditional statement for if the tweet text containing bugs, then perform an action.
- Create a new action in the conditional statement for posting to a Slack channel for your team.

Common Cloud App Scenarios

Architecture diagram



Additional resources

Send personalized notifications with Mobile Engagement

List of managed APIs in Logic Apps

Services Used

- *Mobile Apps*
- *Mobile Engagement*
- *Logic Apps*

Pro Tip

Take a deep dive into the end-to-end platform and tools from Microsoft for creating mobile enterprise apps.

Scenario 3: Building an Internet of Things (IoT) App

Overview

You develop applications that collect large amounts of information from connected devices. You want to monitor for usage and anomalies by pushing real-time data to dashboards for business and infrastructure users. You also want to predict trends for growth and avoiding failures for the connected devices with predictive analytics. And, you want to archive raw data and clean up the data automatically.

The problem

Although many companies are using analytics as part of their business models, there can be a lot of useful data that is going to waste within organizations. The task of processing a lot of data, especially IoT data, sounds daunting. When developers often think of machine learning, they think of using R programming and needing an advanced degree in data science.

The solution

Do you need to understand R and get a doctoral degree to work with Azure Machine Learning? Both of those certainly wouldn't hurt. But Microsoft has intentionally made Machine Learning with developers in mind, being both accessible to curious beginners and powerful enough for data scientists.

You can connect various Azure technologies together to ingest, predict, and output data to a dashboard. Use Azure IoT Hub and Stream Analytics to connect to your devices, grab, and process those events. Store that data in Azure Storage for archival use while using Azure Machine Learning to create models for predictions. Connect Power BI to Stream Analytics directly and view those predictions in a dashboard in real-time for your business users. Use Azure Functions to clean up data by responding to changes in Azure Storage.

Services Used

- *IoT Hub*
- *Stream Analytics*
- *Machine Learning*
- *Storage*
- *Power BI*
- *Functions*

The how

Competitive businesses are able to take data that they already have and turn it into valuable predictions. Those companies are able to use those predictions for creating product recommendations, fraud detection, and preemptive maintenance. But how do you get from point A with just raw data to point Z, creating predictive analytics solutions from your data?

Use IoT Hub to log millions of events per second from connected devices. Implement Stream Analytics to ingest those events and process them in real-time, and add in a Machine Learning function to create a Machine Learning model and output the Stream Analytics job result to Azure Blob Storage for archiving. Then, add a Power BI output to your Stream Analytics job and create a dashboard from the dataset. Create a function to execute cleanup functions when Blob Storage reaches a certain threshold in size.

Collecting data from connected devices

IoT Hub is a new service we've introduced to meet the needs of IoT. The IoT Hub is the centerpiece of an Azure IoT solution, serving as the cloud gateway for your "things" to connect to. It can scale to millions of connections per hub and can process massive volumes of data. It also provides the ability to communicate from the Cloud back to the devices it knows, thus enabling command and control capabilities. The IoT Hub speaks multiple protocols, including HTTP, MQTT, and AMPQ. The IoT Hub also plays an important role in helping to secure

your solution by providing per-device authentication support.

- Get started with our step-by-step guide to connecting your IoT devices.
- Add rules with send, and manage, send, and listen rights and copy the two connection strings.

Processing data in real time

Stream Analytics is a fully-managed, cost-effective, and real-time event processing engine that helps to unlock deep insights from data. Stream Analytics makes it easy to set up real-time analytic computations on data streaming from devices, sensors, web sites, infrastructure systems, and more. Stream Analytics connects directly to IoT Hub for stream ingestion, and results can be written from Stream Analytics to Storage Blobs or Tables, among other solutions in Azure. You can also optionally output directly to Power BI.

- Create a new Stream Analytics job.
- Specify the job input for the Stream Analytics job.
- Specify the job query (or queries) for describing transformations for real-time processing.
- Specify the job output of the Storage blob container to store the processed data.
- Specify another job output for Power BI as well to post real-time data to a dashboard.

Storing processed output data

Table or Blob Storage is available for storing processed data from Stream Analytics. Storage is a much lower-cost solution for storing large amounts of data over a traditional relational data store.

- Create a unique Storage account in the portal.
- Create a new container, and set its access for the container to be Public Blob.
- Create Shared Access Signatures to lock down storage containers.
- Set the job output of Stream Analytics to the storage container.

Creating predictive solutions from data

Azure Machine Learning not only provides tools to model predictive analytics, but also provides a fully-managed service you can use to deploy your predictive models as ready-to-consume web services. You can either use Machine Learning as a function in Stream Analytics (which then stores results in Blob storage), or you can use the Machine Learning REST API and connect that to the Power BI REST API.

- Build a new experiment through Azure Machine Learning Studio.
- Create a model by getting data from Storage, pre-processing data, and defining features.
- Train the model by applying a learning algorithm.
- Score and test the model to create predictive solutions.

Displaying your predictions in a dashboard

Microsoft Power BI helps you connect to multiple datasets to bring all of your relevant data together in one place. Power BI dashboards display tiles that you can click to open reports for exploring further. Use the REST API for Power BI to connect to other REST APIs, and transform data on your machine with Power BI Desktop.

- Add the job output for the Stream Analytics job to Power BI and authorize the connection.
- Query the data for what you want in a report or dashboard (if using a new job).
- Create the dashboard in Power BI online.

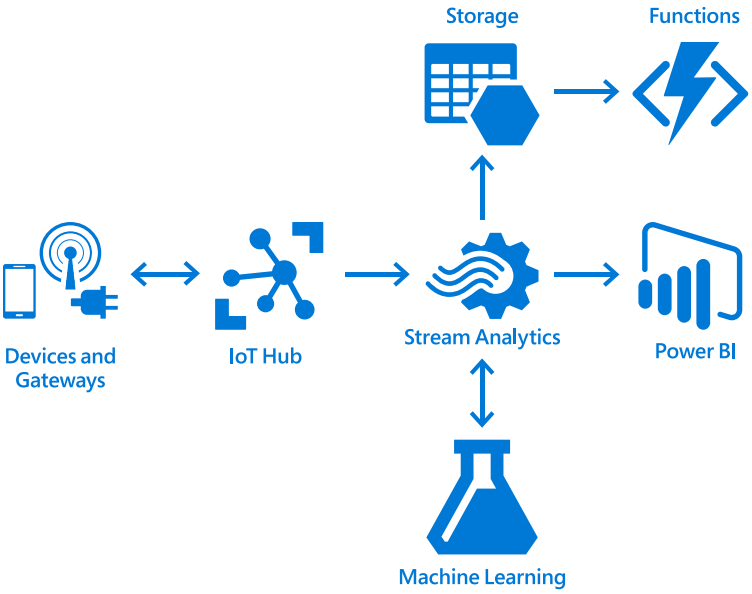
Running scalable event-driven functions

Azure Functions is a new server-less compute service that allows you to create a function in a variety of languages that can respond to events such as HTTP requests, or Azure services events like Azure Blob Storage updates. Your Azure

Common Cloud App Scenarios

Functions run in a dynamic compute environment which automatically scales out to meet demand, and you're only charged for the time your function runs. Learn more about using Azure Functions.

Architecture diagram



Additional resources

Azure IoT Developer Center

Machine learning algorithm cheat sheet

Power BI overview and learning

Pro Tip

Learn how to perform sentiment analysis using Stream Analytics and Machine Learning functions here.

Scenario 4: Building a Custom Microservice-based App

Overview

Your startup, ISV, or enterprise business faces explosive growth and you want the confidence that key applications and services supporting the business will scale and be available around-the-clock while your developers continually patch the code and add new features. You want to build new applications and services with cloud-native architectures optimized for the dynamic, unlimited scale of the cloud.

If your organization's application needs don't fit the standard approaches mentioned previously, then a custom approach is called for to build a unique solution that will drive the success of your business. If you are a startup ISV company or enterprise whose core business depends on a specific service being available, you want a higher degree of control and customization to ensure success.

The problem

Whether you're building financial trading systems, instant messaging services, data collection hubs, dynamic video ad-serving, or managing patient records at national scale, you want more control and flexibility than the typical web or mobile app platform can provide.

You're willing to invest in the ongoing development and operation of a custom architecture to create an application matched exactly to the success of your business. However, you want your developers to build on a platform that lets them focus on delivering business value in an agile way without having to manage the underlying infrastructure.

The solution

Build your application using a microservice approach and deploy it onto Azure Service Fabric. The microservice architecture breaks down an application into small, independently executing microservices resulting in the following benefits:

- Services are easier for developers to understand and build/rebuild.
- Services can be developed and deployed in a more independent way speeding development.
- Services start and stop quickly speeding deployments and increasing developer productivity.
- It's easier to dynamically scale only those components that are needed to meet demand.
- The system is more tolerant to faults and therefore has higher availability.
- Individual microservices can be upgraded live without interruption to the service.

But microservice architectures have a higher degree of complexity which requires a PaaS layer to effectively deploy and manage them at scale. That's what deployment onto Azure Service Fabric does for you – so your developers can focus on building business value. (For a deeper look at Service Fabric and microservices see "Service Fabric" later in the guide)

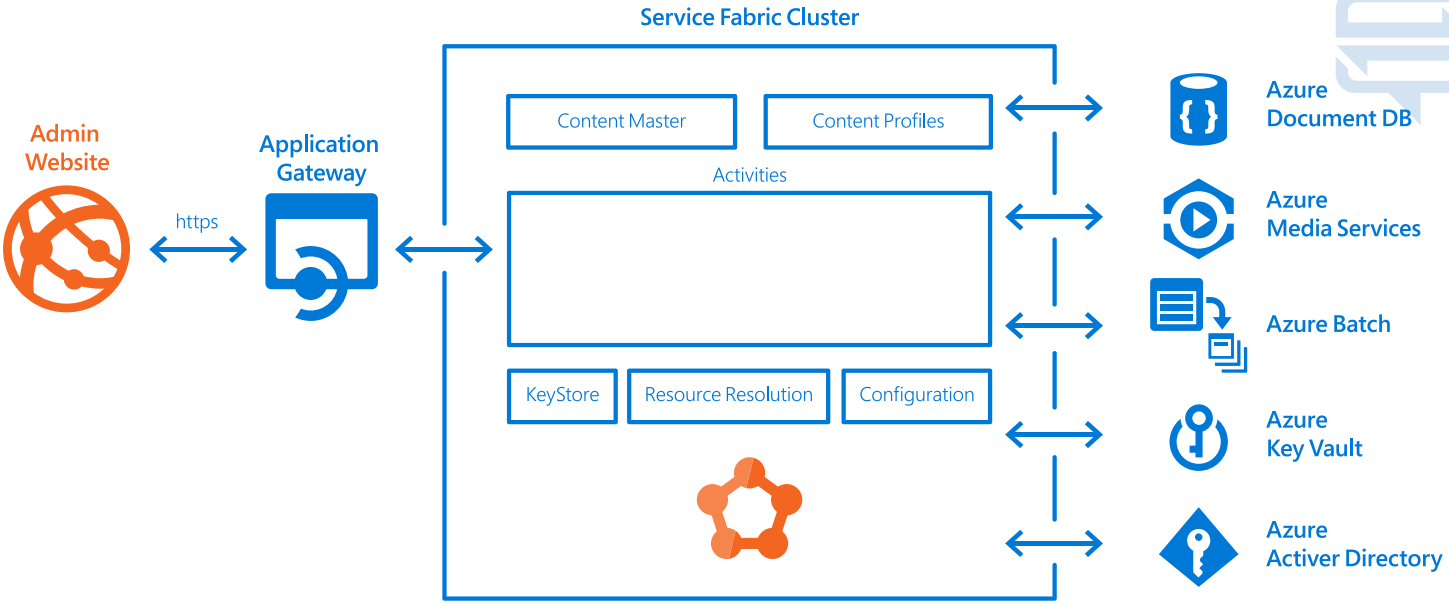
The how

Architect your application using a collection of smaller microservices. Each service should:

- Be independently deployable and isolated
- Maintain its own data
- Publish at least one secured API endpoint

To make the process of building microservices simpler, Service Fabric supports the Actor programming model which allows developers to model object-oriented services and their interactions as a means to abstract away from explicit data and networking operations and be very productive building microservice based applications.

Common Cloud App Scenarios



Architecture diagram

Service Fabric supports both stateless and stateful services, where stateful services have local data that persists even if the service fails. The potential different architectures are limitless. However, as an example, the above figure shows how a customer, TalkTalk TV, built a microservices application on Service Fabric while leveraging multiple other Azure services such as Azure Web Apps for the Admin website and Azure AD for authentication.

Additional resources

TalkTalk Case Study

Azure Service Fabric Website

The Azure Platform Services

Azure App Service

Azure App Service is a cloud platform to build powerful web and mobile apps, for any platform and any device, that connect to data anywhere, in the cloud or on-premises. Built for developers, App Service is a fully managed platform with powerful capabilities such as built-in DevOps, continuous integration with Visual Studio Team Services and GitHub, staging and production support, and automatic patching.

App Service allows you to create the following app types from a single development experience:

- **Web Apps**—Quickly create and deploy mission critical web apps that scale with your business.
- **Mobile Apps**—Build mobile app back ends with notifications, data sync, and authentication.
- **API Apps**—Easily build and consume cloud APIs.
- **Logic Apps**—Automate the access and use of data across clouds without writing code.

App Service provides an integrated set of enterprise capabilities through a single development and management experience offering you the following benefits:

- **Build web and mobile apps fast.** Rapidly build, deploy and manage web and mobile back end apps for employees or customers. Use your existing languages skills—.NET, Java, NodeJS, PHP, or Python. Accelerate development with access to a rich gallery of APIs, connectors, and logic available in the Azure Marketplace.
- **Connect to any service and unlock your data.** Connect your web or mobile app to enterprise systems or SaaS in minutes with built-in connectors. Choose from more than 50 connectors for enterprise systems such as SAP, Siebel, and Oracle to popular enterprise SaaS services like Salesforce and Office 365 to popular internet services such as Facebook, Twitter, and Dropbox.
- **Integrate more easily.** Logic Apps lets you integrate data across clouds and automate business processes in minutes using a visual design experience. Easily integrate your logic with any mobile or web app via standard REST APIs. Build sophisticated enterprise application integration, B2B solutions using Electronic Data Interchange (EDI) and business policies (rules engine).
- **Increase developer productivity.** Optimized for DevOps, with continuous integration support for Visual Studio Team Services and GitHub, so you can focus on rapidly improving your apps without worrying about infrastructure. Deploy app updates with built-in staging, roll-back, and in-production testing capabilities.

- **Rely on Enterprise-grade services** – App Service has full enterprise-grade security and management. Provide delegated and role-based administration; easily secure and manage data flowing to your mobile apps; and protect your assets with built-in backup and restore capability. Fully PCI-compliant with dedicated environments, and the ability to deploy across public and private clouds.

As a single integrated service, App Service makes it easy to compose the above app types into a single solution, allowing you to easily build apps that target both web and mobile clients using the same back end and integrate with on-premises systems as well as popular SaaS services.

App Service Plans represent a set of features and capabilities that you can share across your apps. App Service Plans support a number of pricing tiers (e.g. Free, Shared, Basic, Standard, and Premium) where each tier has its own capabilities. You can quickly change which App Service Plan an app is hosted on with no downtime.

Azure Service Fabric

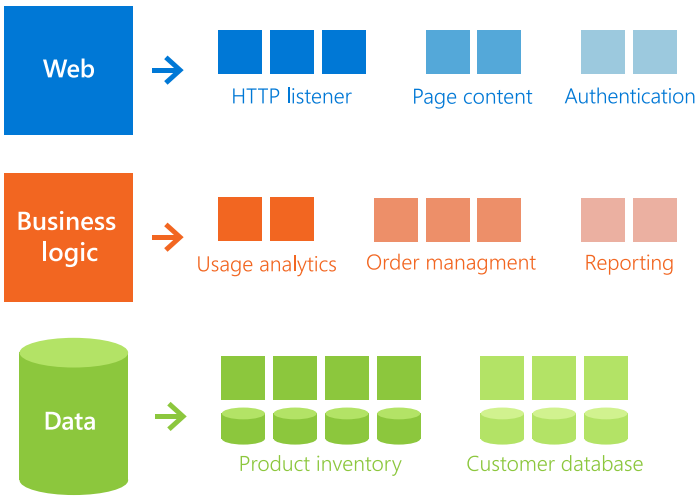
Azure Service Fabric is a mature, feature-rich microservices application platform with built-in support for lifecycle management, stateful and stateless performance at scale, hybrid deployments, 24x7 availability, and cost efficiency.

Microservices are an application development and deployment approach perfectly suited to the agility, scale, and reliability requirements of modern cloud applications. In a microservices model, you individually build and deploy small, independently-executing services or "microservices," that collaborate using published API calls across the network to deliver the overall application's functionality. This results in a fine-grained, loosely coupled application that can easily be distributed across multiple host machines for scale and reliability. Contrast this with the monolithic three-tier approach common in many applications today.

The term "microservice" emphasizes the fact that applications should be composed of services small enough to truly implement a single role. Each has well-defined contracts (API contracts) for other microservices to communicate and share data with it. Microservices must also be able to version and update independently of each other. This loose coupling is key to supporting rapid and reliable evolution of an application. What would have been a single tier of a monolithic application decomposes into many discrete microservices, each independent and isolated.

In its initial public release, Azure Service Fabric runs on Windows and include C++ and C# language integration, but Linux and Java support are under development. Service Fabric has built-in support for lifecycle management, hybrid deployments, and 24x7. The platform offers extensible health models for both the infrastructure and microservices to enable automated health-based upgrades and automatic rollback, simplifying DevOps. It supports both stateless and stateful microservices with leadership election to support data consistency and a state replication framework that supports transactions for stateful data guarantees. Learn more about running Service Fabric on Azure.

Service Fabric applications can be composed of both stateless and stateful microservices. The reliable stateful data management Service Fabric is highly tuned to provide maximum availability and durability of data, while optimizing every cycle and I/O operation for high performance.



To make microservice development more productive, Service Fabric includes an approach called the actor programming model. This approach enables you to think about microservices as "actors", each representing a logical entity functions and state, whether it's an IoT device, a game player, game session, or patients and doctors in a health provider network. One way to look at the actor programming model is as object-oriented microservices, and it can greatly simplify the app development.

Service Fabric has been in production use at Microsoft since 2010 and powers many of our applications and services, including Azure SQL Database, DocumentDB, Intune, Cortana and Skype for Business. In its largest deployments, it manages hundreds of thousands of microservices across thousands of servers. We've taken the exact same technology and released Service Fabric directly, as a service on Azure and it will soon be available to run on-premises and in other clouds.

Visual Studio's integration with Service Fabric makes coding and deployment easy, and there's a complete developer-machine experience that runs the actual Service Fabric runtime on a single box, simulating a multi-server Service Fabric cluster for validation and diagnostics.

Finally, Service Fabric can be easily plugged into your application lifecycle allowing for a seamless continuous integration (CI) / continuous delivery (CD) of both stateless and stateful updates. To help manage the application lifecycle, Service Fabric supports both rolling and side-by-side upgrades (e.g. scaling up version 2 while scaling down version 1), and automatically rolls back updates if it detects that the upgrade has degraded application health.

Cloud Services

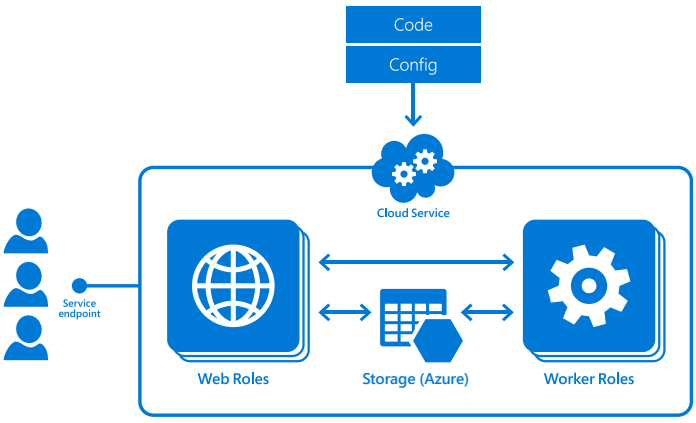
Cloud Services is designed to support applications that are scalable, reliable, and cheap to operate. Just like an App Service is hosted on virtual machines, so too are Azure Cloud Services, however, you have more control over the virtual machines. You can install your own software on Azure Cloud Service virtual machines and connect into them remotely.

More control also means less ease of use; unless you need the additional control options, it's typically quicker and easier to get a web application up and running in Web Apps in App Service compared to Azure Cloud Services.

The technology provides two slightly different virtual machine options: instances of *web roles* run a variant of Windows Server with IIS, while instances of *worker roles* run the same Windows Server variant without IIS. A Cloud Services application relies on some combination of these two options.

For example, a simple application might use just a web role, while a more complex application might use a web role to handle incoming requests from users, then pass the work those requests create to a worker role for processing. (This communication could use Service Bus or Azure Queues.)

Even though applications run in virtual machines, it's important to understand that Azure Cloud Services provides PaaS, not IaaS. All you have to do is deploy your application. Management of the platform it runs on, including deploying new versions of the operating system, is handled for you.



Azure Functions

Azure Functions is a serverless, event-driven experience that extends the existing Azure application platform with capabilities to implement code triggered by events occurring in other Azure services, SaaS products, and on-premises systems.

Azure Functions extends our market leading PaaS platform, building on the existing WebJobs infrastructure to let developers easily implement code that reacts to events generated from across the breadth of Azure. Whether you want to respond to changes in Azure storage containers, events emitted in SaaS products that support Web Hooks, or by calls to an HTTP endpoint, they're all easy to set up and require minimal configuration.

You can implement your functions in a variety of languages including JavaScript, C#, Python and PHP. In addition, you can choose scripting options like Bash scripts, PowerShell scripts, or Windows batch files. Azure Functions provides an intuitive web-based code editing experience or you upload and trigger pre-compiled executables built in the development tool of your choice. You can quickly and easily iterate on your Azure Functions with continuous deployment using Visual Studio Team Services, GitHub, or BitBucket. Monitor and troubleshoot your Azure Functions using the embedded logging environment.

Azure Functions introduces a new pricing model where you will only be charged for the time your code is running. When an Azure Function is invoked it will be provided with as many resources as it needs to execute only for as long as it is executing. You can help secure your Azure Functions by hosting them on an App Service Environment that can be configured to be addressable only from internal networks.

Building on IaaS

The host operating system is *the* original application platform, and many developers continue to write applications that run natively on Windows Server, Linux, or other operating systems. Traditionally the operating system would be installed on a single physical host machine and the application installed and run using resources, such as files, networking, time-slicing, memory, and security, provided by the operating system. Today the operating system would most likely be running on a virtual machine rather than physical host but the pattern is the same; the application runs on a single machine consuming services provided by the OS.

This arrangement provides developers with a very detailed level of control over the environment in which their application executes and, depending on the programming environment, can afford the maximum level of customization where aspects like performance are critical. However, the developer or their IT coworker are now responsible for the upkeep and configuration of the operating system and the intricate details that make it compatible with the application. When the OS needs to be upgraded, the application will need to be tested for compatibility before putting into production.

One serious limitation of this style of application that's highlighted with the advent of application servers and now the cloud is that while the application can be scaled-up by adding capacity to the host machine, the application can't be easily scaled-out to run across multiple machines without developers moving to stateless, load balanced models and making other code changes just to suit the infrastructure.

Applications also become entangled with the OS, making the task of migrating applications between OS versions very risky without fully testing on the new OS version. Docker's packaging format and tools have become popular because they provide a means to build portable images of applications and OS dependencies that can be moved around virtual machines and containers without the risk of incompatible dependencies.

Azure provides a range of options for developers looking to build directly on the operating system:

- **Virtual Machines**—Azure provides a wide range of Windows Server and Linux base operating system images.
- **Virtual Machine Scale Sets**—Virtual Machine Scale Sets provide an easy to use 'slider' based mechanism for scaling identical groups of stateless virtual machines.
- **Azure Container Service**—Container Service is a fully open-source based service providing deployment, orchestration and failover for Docker-based images across a cluster of virtual machines, using Apache Mesos, Mesosphere's Marathon, or Docker Swarm.
- **Azure Marketplace**—Azure Marketplace includes many popular application platform/PaaS platforms in virtual machine images that you can operate and manage for yourself, including Pivotal's Cloud Foundry.

Docker and Containers

You can't have a discussion on cloud computing without talking about containers and the Docker toolset. Organizations across all business segments want to understand what containers are, what they mean for applications in the cloud, and how to best use them for their specific development and IT operations scenarios.

Containers are a form of OS virtualization, similar to Virtual Machines but operating inside an instance of the OS, creating the perception of a fully isolated and independent OS. To the application running in the container the local disk looks like a pristine copy of the OS files, the memory appears only to hold files and data of a freshly-booted OS, and the only thing running is the OS. The host OS also controls resources like CPU, RAM and network bandwidth to ensure a container gets the resources it expects and that it doesn't impact the performance of other containers running on the host.

The combination of instant startup that comes from OS virtualization and reliable execution that comes from isolation and resource governance makes containers ideal for application development and testing and developers can quickly iterate. Because its environment and resource usage are consistent across systems, a containerized application that works on a developer's system will work the same way on a production system. The instant-start and small footprint also benefits cloud scenarios, since applications can scale-out quickly and many more application instances can fit onto a machine than if they were each in a virtual machine, maximizing resource utilization.

Azure Container Service

Containers offer a compact form of virtualization with fast start-up times, highly-efficient system utilization (enabling hundreds of containers on a single machine) and, in Docker images, a packaging format that is fully portable across environments.

Using this common image format, you can test your application on a local development box, using the exact same deployment artefact as will eventually run in production, reducing the need for further testing on production environments. However, the agility and scale offered by containers presents a management challenge for developers and IT administrators when deploying applications into production at scale.

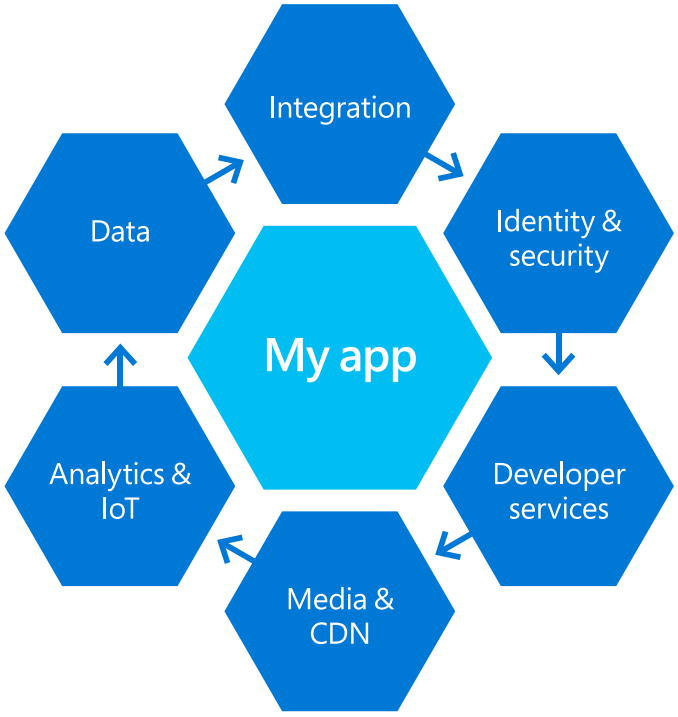
Azure Container Service provides a way to simplify the creation, configuration, and management of a cluster of virtual machines that are preconfigured to run containerized applications. Using an optimized configuration of popular open-source scheduling and orchestration tools, Container Service enables you to use your existing skills or draw upon a large and growing body of community expertise to deploy and manage container-based applications on Microsoft Azure.

Container Service uses the Docker container format to ensure that your application containers are fully portable. It also supports your choice of Marathon and Apache Mesos or Docker Swarm to ensure that these applications can be scaled to thousands, even tens of thousands, of containers.

Azure Container Service exposes the standard API endpoints for your chosen orchestrator. Using these endpoints you can take advantage of any software that is capable of talking to those endpoints. For example, in the case of the Docker Swarm endpoint you might choose to use Docker Compose, while for Apache Mesos you may choose to use the Mesosphere DCOS command-line interface.

At this time, the service supports Linux containers only. Microsoft has committed to providing Windows Server Containers using Docker and Apache Mesos is being ported to Windows. This work will allow us to add Windows Server Container support to Azure Container Service in the future.

Adding Superpowers to your Apps



Once you've addressed the question of what is nature of your app, where will it run and what design approach you should take, you have a world of powerful Azure services at your disposal to make your app truly great.

There are more than sixty—and growing—Azure services today that you can use to develop, host, manage, support, secure, integrate, extend, and complement your application development. In this section we'll take a look at some of the most common services you should consider when building modern apps.

View a complete list of all Azure services.

Database Services

PaaS database services (sometimes referred to as *database as a service*) have a number of advantages over the IaaS approach of running a database management system (DBMS) in a virtual machine:

- **Fully managed**—You don't need to install, patch, or manage a DBMS.
- **Built-in scalability**—Rather than adding database servers manually as your application's load increases, a PaaS database service can do this on demand with no downtime.
- **Built-in reliability and fault tolerance**—Rather than requiring specialized skills and plenty of time to set up a clustered DBMS, a PaaS database service can do this for you.
- **Service level agreements (SLAs)**—Rely on Azure to meet your availability targets, instead of worrying about availability yourself.

SQL Database

Based on SQL Server, Azure SQL Database offers a familiar relational store, including support for SQL queries, transactions across an entire database, and stored procedures. It also provides built-in fault tolerance and scalability.

SQL Database is a good choice when an application needs the full power of a relational system. It's also a good choice when the development team is already familiar with SQL and relational technologies. Since this PaaS database service is based on SQL Server, learning it commonly isn't difficult.

DocumentDB

As its name suggests, Azure DocumentDB stores *documents*, each containing JSON data. This PaaS database service allows RESTful access to the documents it contains, and it also lets applications issue queries using an extended subset of SQL. And like SQL Database, DocumentDB provides transactions, built-in scalability, and built-in high availability.

DocumentDB is an attractive choice for developers working in JavaScript and other modern languages, all of which provide built-in JSON serializers. It's also good for situations where the data's structure changes frequently, because unlike a relational database, DocumentDB doesn't define a fixed schema.

HDInsight HBase

HBase is part of the Hadoop technology family, and so it's designed for processing big data. HBase can be viewed as a *column-family store*. The columns in each table are grouped into families, and requests for data can specify which column family to look in. Unlike relational tables, however, HBase allows adding a new column to a column family at runtime—the schema isn't fixed. It's also designed to be highly scalable, letting applications create tables with millions of columns and billions of rows.

HBase is an excellent choice for applications that need to create big but sparse tables. It's also a good option when the data it stores will be processed with Hive or another HDInsight technology, since all of these rely on the same underlying clustering technology.

Tables

Some situations need the full power of a relational database, including SQL queries against relational tables. Others, however, can get by with a much simpler approach to storing and accessing data. Despite the name of this PaaS database service, it doesn't really store data in tables. Instead, an application accesses data by providing a unique key. This service then returns whatever set of values are associated with that key.

This simple approach works well in a surprising number of situations. Think about storing user profile data, for example. Each user can have a unique key, with each key providing access to whatever profile data is stored for that user. Different users can have different data—there's no fixed schema—and

so Tables provides a flexible approach.

Other data services

Along with the technologies described so far, Azure also provides two other PaaS database services for working with operational data. Neither one addresses the same type of problem as the ones already mentioned, but they're both important to understand. Those two services are Azure Search and Azure Redis Cache.

Azure Search

For many people, search has become the most attractive way to interact with applications. Rather than choose items from menus, why not let an application's users search for what they're interested in, much as they would on the Internet? Allowing this would make many applications easier to use.

Azure Search helps developers add search capabilities to their application and provides things such as automatic bolding of search terms in results and a way to control the order in which these results are returned. It also supports the ability to provide suggestions, offering possible search phrases based on a user's initial entry.

Redis Cache

With Redis Cache, an application can access data from any Azure PaaS database service, as usual. The application can then store a copy of that or other data in Redis Cache. When the application needs this data in the future, it can access the data in the in-memory cache rather than going back to the PaaS database service. Doing this is faster, letting applications have better response time and handle more simultaneous users.

(Database Services section adapted from "Data in a PaaS World: A Guide for New Applications" by David Chappell)

Advanced Analytics & IoT

Advanced analytics helps transform data into intelligent action through collecting and managing unlimited data, extending apps with predictive insights, and operationalizing data science pipelines for iterative learning.

Cortana Intelligence Suite

A fully managed big data and advanced analytics suite that enables you to transform your data into intelligent action:

- **Information management**—Orchestrate data movement on a fully managed, end-to-end platform. Use Azure Data Factory to build pipelines and collect and orchestrate data from the services you use for easier analysis. Plus, use Azure Data Catalog to effectively manage data sources and Azure Event Hubs to provide a staging area for incoming streaming data.
- **Big data stores**—Store and manage structured data using Azure SQL Data Warehouse that elastically scales with massively parallel processing. Implement a hyper-scale repository with no file size limits for unstructured data using Azure Data Lake Store to attain massive throughput and analytic performance.
- **Machine learning and analytics**—Design and publish predictive models with Azure Machine Learning, use Azure HDInsight to analyze data in Storm and Spark for Hadoop environments, integrate your code from R or Python, and analyze any kind or any size of data you need with Azure Data Lake Analytics and Azure Stream Analytics. Plus, use Microsoft Power BI to create rich visualizations that bring your data to life.
- **Intelligence**—Explore the Cortana Intelligence Gallery for downloadable, fully managed APIs to quickly implement predictive APIs for business scenarios (recommendation, forecasting, anomaly detection, and customer churn), as well as Cognitive Services APIs (vision, face, text, and speech) to interact in new ways with customers. Get started

with solution templates tailored to your specific business needs. Plus, integrate your analytics services and models with Cortana, our personal digital assistant, to let users naturally interact via speech or receive proactive notifications. Integrate your analytics services and models with intelligent agents (Bot Framework) as well as Cortana, our personal digital assistant, to let users naturally interact via speech or receive proactive notifications.

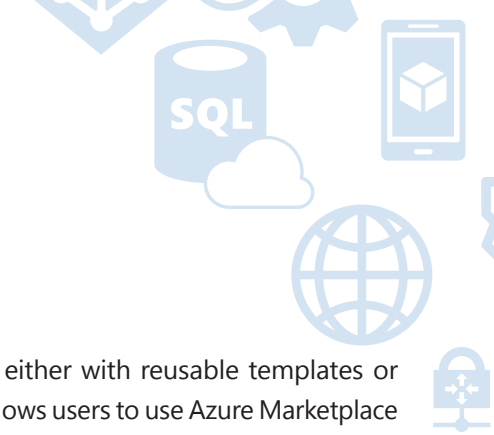
Common Scenarios

- **Healthcare:** The proliferation of healthcare data available—from new data sources streaming in real time to historical data stored on health provider systems—combined with the power of advanced analytics can help transform current healthcare business challenges into predictive and prescriptive solutions.
- **Financial Services:** Existing anti-money-laundering systems are robust and well developed through years of iteration and improvement, yet trillions of dollars are still lost to money laundering every year. Using big data tools to manage unstructured data, the Cortana Intelligence Suite augment existing systems to provide additional clarity in combatting money laundering.
- **Retail:** The proliferation of structured and unstructured data available combined with the power of advanced analytics solutions provides an opportunity for big data to solve problems. The technologies united within the Cortana Intelligence Suite can help retail organizations successfully turn big data into insights into personalized customer experiences.

Internet of Things (IoT)

Azure IoT Suite is an enterprise-grade solution that enables developers to get started quickly through a set of extensible preconfigured solutions that address common IoT scenarios, such as remote monitoring and predictive maintenance. These are complete, working end-to-end solutions, including simulated devices that make use of Azure services.

- **Azure IoT Hub**—Developers can easily and securely connect new devices, and connect existing ones, using open-source device SDKs for multiple platforms, including Linux, Windows, to reliably (intermittent connection) and securely send commands and notifications to connected devices and track message delivery.
- **Azure Event Hubs**—A highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple apps. This lets developers process and analyze the data produced by connected devices and apps and transform and store it by using any real-time analytics provider or with batching/storage adapters.
- **Azure Stream Analytics**—Developers can rapidly develop and deploy low-cost solutions to gain real-time insights from devices, sensors, infrastructure, and applications, e.g. real-time remote management and monitoring or gaining insights from devices like mobile phones and connected cars.
- **Azure Machine Learning**—A powerful cloud-based predictive analytics service that makes it possible to quickly create and deploy predictive models as analytics solutions. It provides tools to model predictive analytics, but also provides a fully-managed service to deploy predictive models as ready-to-consume web services. Quickly create, test, operationalize, and manage predictive models.



Developer Services

Visual Studio Team Services

Visual Studio Team Services provides a set of cloud-powered collaboration tools that work with your existing IDE or editor, so your team can work effectively on software projects of all shapes and sizes, in any language and on any platform. These tools cover the entire software lifecycle—right from the time you add a task to the backlog/plans, to the time when the task gets coded and delivered in a release.

- **Tools for agile teams**—Capture, prioritize, and track work with backlogs and customizable Kanban boards. Work items link directly to code to ensure transparency, and can be used to build rich dashboards for easy reporting.
- **Version control**—Store and collaborate on code with unlimited private repositories and Nuget package management service. Use Git for distributed version control to maximize collaboration or use Team Foundation Version Control (TFVC) for centralized version control.
- **Continuous Integration and cross-platform builds**—Catch quality issues early with continuous integration (CI) builds that compile and test your application automatically after any code change. Use continuous delivery to automatically deploy applications or websites that pass tests. Set up release environments and policies to manage release pipelines.
- **IDE integration**—Use your favorite language and development tool. Version control supports any language, as well as any Git client (including Xcode). Java teams can access code and work items through free plugins for Eclipse, Android Studio and IntelliJ—and run continuous integration builds based on config files from Ant or Maven.

Dev-Test Labs

Dev-Test Labs allows for fast, easy, and lean Dev/Test environments in the cloud. You can quickly provision development and test environments by creating virtual machines in few a clicks either with reusable templates or Azure marketplace. This allows users to use Azure Marketplace images as virtual machine base in the lab in addition to their custom images (VHDs) uploaded to the lab. You minimize waste with quotas and policies, and you can set automated shutdowns to minimize costs—all benefits of using the cloud for your development and test environments—whether for Windows or Linux workloads.

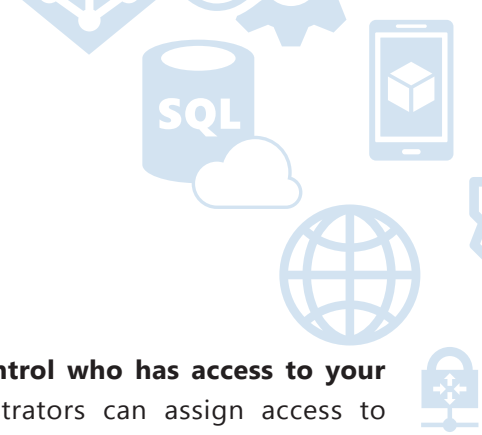
HockeyApp

HockeyApp makes distributing, testing and getting feedback on your client mobile apps simple. With support for iOS, Android, or Windows; and native code, along with Xamarin, Cordova, and Unityyou are covered with the open source HockeyApp SDKs.

HockeyApp provides webhooks and an API to support integration with your existing ALM tools. This allows you to adopt a Mobile DevOps practice without needing to change your existing workflows. HockeyApp supports work item creation based on app feedback and crashes so you can manage your backlog in one place.

Application Insights

Application Insights gives you rich web app and web service performance monitoring capabilities and deep diagnostics information, enabling you to quickly respond to web app and service issues. With support for public, private, and hybrid clouds and on-premises deployments—and nearly any web development language—Application Insights can provide you the detect, triage and diagnose capabilities you need to always have confidence in your web apps and services.



Identity

Azure Active Directory (AD) provides organizations with enterprise-grade identity management for cloud applications. Azure AD integration gives your users a streamlined sign-in experience, and helps your application conform to IT policy.

Support Azure AD as a way to sign in to your application

- **Use Azure AD to sign in to your application.** Your users won't have one more name and password to remember, and you'll have one less password to store and protect. Azure AD powers sign in for some of the world's most popular cloud applications, including Office 365 and Microsoft Azure. Learn more about adding support for Azure AD sign in.
- **Simplify sign up for your application.** During sign up for your application, Azure AD can send essential information about a user so that you can pre-fill your sign up form or eliminate it completely. Users can sign up for your application using their Azure AD account via a familiar consent experience similar to those found in social media and mobile applications. Learn more about signing up your application for Azure AD Account login.

Browse for users, manage user provisioning, and control access to your application

- **Browse for users in the directory.** Use the Graph API to help users search for other people in their organization when inviting others or granting access, instead of requiring them to type email addresses. Learn more about the Graph API.
- **Re-use existing Azure AD groups and distribution lists.** Azure AD contains the groups that your customer is already using for email distribution and managing access. Re-use these groups instead of requiring your customer to create and manage a separate set of groups in your application.
- **Use Azure AD to control who has access to your application.** Administrators can assign access to applications to specific users and groups. Read this list and use it to control provisioning and de-provisioning of resources and access within your application.

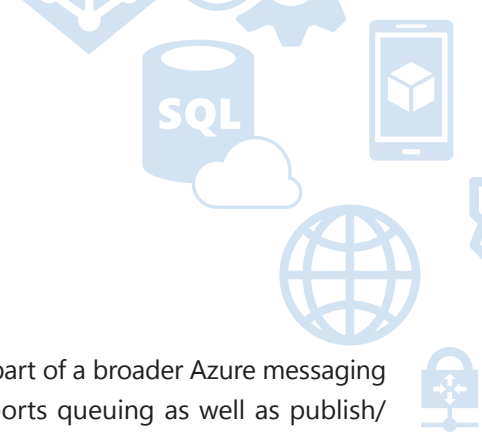
Advanced Security Features

- **Multi-factor authentication.** Azure AD provides native multi-factor authentication. IT administrators can require multi-factor authentication to access your application, so that you don't have to code it yourself. Learn more about Multi-Factor Authentication.
- **Anomalous sign in detection.** Azure AD processes more than a billion sign ins a day, using machine learning algorithms to detect suspicious activity and notify IT administrators of possible problems. By supporting Azure AD sign in, your application gets the benefit of this protection. Learn more about viewing Azure Active Directory access report.

Microsoft Graph

Microsoft Graph exposes multiple APIs from Microsoft cloud services through a single REST API endpoint (https://graph.microsoft.com). Microsoft Graph gives you:

- A unified API endpoint for accessing aggregated data from multiple Microsoft cloud services in a single response
- Seamless navigation between entities and the relationships among them
- Access to intelligence and insights coming from the Microsoft cloud



Integration

Azure provides a wide variety of integration services allowing you to you to extend integration solutions to the cloud:

BizTalk Services

BizTalk Services provides out-of-the box, cloud to on-premises and line-of-business application integration for SAP, Oracle EBS, SQL Server, and PeopleSoft. It lets you connect with any HTTP, FTP, SFTP, or REST data source. You can route messages by using various Azure artifacts such as Service Bus queues, Topics, SQL Database, and Blob storage.

Hybrid Connections

The Hybrid Connections feature of BizTalk Services lets you connect Azure Websites or Azure Mobile Services to any on-premises TCP or HTTP resource—such as Microsoft SQL Server, MySQL, or any web service—with just a few configuration changes and without using any custom code.

Service Bus

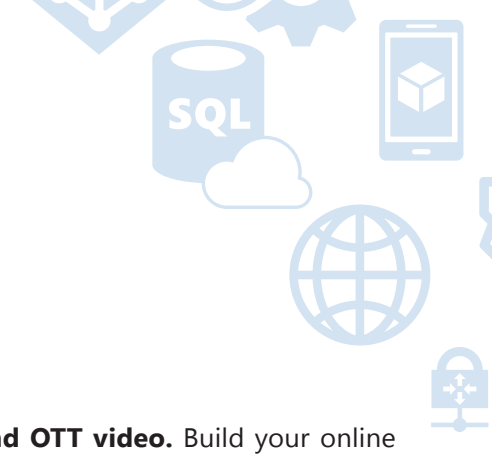
Azure Service Bus is a generic, cloud-based messaging system for connecting just about anything—applications, services, and devices—wherever they are. You can connect apps running on Azure, on-premises systems, or both. You can even use Service Bus to connect household appliances, sensors, and other devices like tablets or phones to a central application or to each other.

Queues

Azure supports two types of queue mechanisms: Azure Queues and Service Bus Queues.

- Azure Queues, part of the Azure storage infrastructure, feature a simple REST-based Get/Put/Peek interface, providing reliable, persistent messaging within and between services.
- Service Bus queues are part of a broader Azure messaging infrastructure that supports queuing as well as publish/subscribe, Web service remoting, and integration patterns. For more see the overview of Service Bus messaging.

Azure Queues were introduced first, as a dedicated queue storage mechanism built on top of the Azure storage services. Service Bus queues are built on top of the broader "brokered messaging" infrastructure designed to integrate applications or application components that may span multiple communication protocols, data contracts, trust domains, and/or network environments.



Media Services & CDN

Media Services

Azure Media Services combines powerful and highly scalable cloud-based encoding, encryption, and streaming components to help customers deliver premium video content to larger audiences on today's most popular digital devices, such as tablets and mobile phones.

- **Encoding**—From simple web delivery for HTML5 to complex media decision logic that solves demanding studio workflows, Azure Media Services allows you to define your own encoding workflows.
- **Indexing**—Make your media content deeply searchable. Using deep neural net (DNN)-based speech recognition technology from Microsoft Research, Media Indexer converts digital audio into natural language and automatically extracts meaningful metadata from your media.
- **Content protection**—Azure Media Services is the only cloud media solution offering on-the-fly encryption for both video on demand (VOD) and live streams
- **Azure Media Player**—Automatically picks the best format for a browser or device, using the dynamic packaging capabilities of Azure Media Services to play adaptive streaming content in multiple formats. For developers there's a simple, unified interface to access APIs.

Common Scenarios

- **Deliver business video in your organization.** The platform capabilities, combined with partner solutions, make it easy to integrate video into your apps and organization - for training, corporate communication, and meetings. Media Services provides scalable, always-available, secure delivery of video to both employees and external customers via Azure.
- **Stream broadcast and OTT video.** Build your online audience and extend your reach by distributing your content to viewers on phones, tablets, and all their digital devices. Plus, take advantage of current event or niche markets by bringing new channels to market quickly and encoding over-the-top (OTT) video to multiple formats.
- **Deliver government content.** Government agencies can deliver secure video streaming to mobile devices by using Azure Media Services and Azure Government, a CJIS-compliant and secure platform.
- **Get Content Closer to your Users.** Azure Content Delivery Network lets you deliver high-bandwidth content to users around the world with low latency and high availability via a robust network of global data centers. It sends audio, video, applications, images, and other files to users from the nearest servers.

Conclusion

Like developers, apps come in all sizes. Apps once thought impossible due to scale, complexity, or because they simply couldn't be imagined, are now a reality with the cloud. In this guide we've explored the Azure app platform services and highlighted Azure's support for platform as a service (PaaS) in particular. So why consider PaaS as your default choice? Let's consider one last analogy.

Phones provide huge social and business advantages. But laying and maintaining the millions of miles of wire to support a ubiquitous infrastructure is expensive. Today, phone usage is growing fastest in Africa. But are they following the path of Europe, America and others who laid down land lines? No! In fact, line installation peaked in 2009 with 4 percent of the population having access to a land line. Instead, they are leapfrogging landlines and moving directly to cell phones, which are far cheaper to install and maintain.

In many ways, PaaS is a like the cell phone. It provides greater functionality, lower upfront investment, and greater developer productivity than IaaS, eliminating much of the initial infrastructure requirements and reducing ongoing maintenance costs. Azure gives developers a comprehensive PaaS application platform for building, deploying and managing apps of all kinds, from the simplest website to the most complex business solution.

Welcome to the developer wave. Welcome to Azure.

Recommended next steps

- Be our guest for up to an hour of Azure App Service experience with no subscription, free of charge and commitment.
- Explore the range of free options available to get you started, like hosting up to ten free web and mobile apps on Azure App Service, sending up to one million notifications with Notification Hubs or creating Machine Learning experiments.
- Create an Azure account and get started for free with $200 in Azure credit.