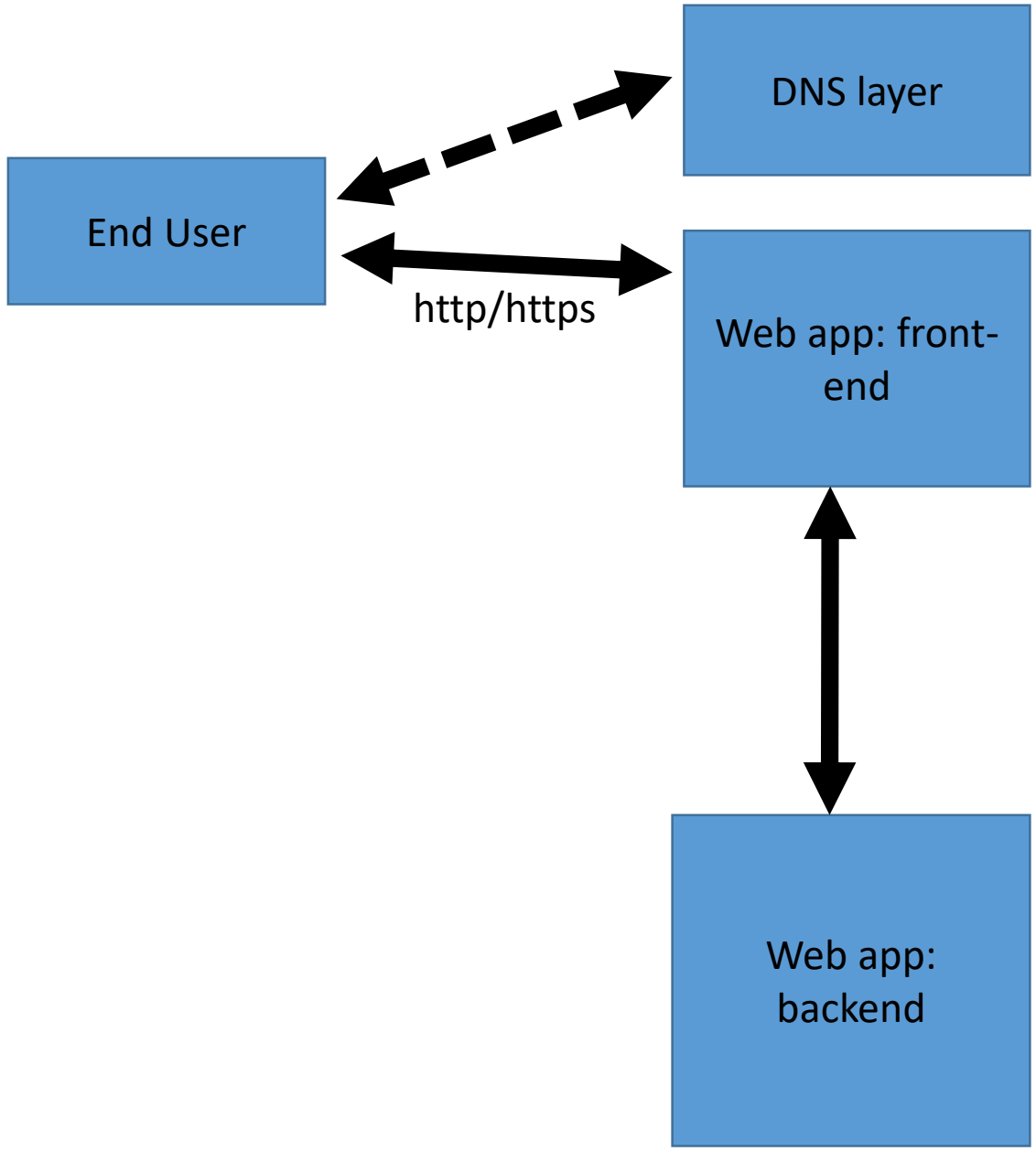
Microsoft Azure App Services: technical deep dive into Web Apps, DNS, TrafficManager and Application Gateway

Alexey Bokov / abokov @ microsoft.com / twitter: @abokov
Senior Program Manager
Technical Evangelist and Development team
Microsoft Corp

Contents

Azure Web application principal scheme

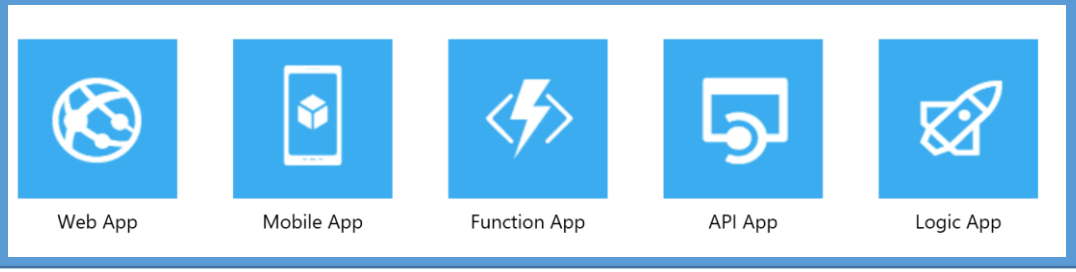


Azure DNS

Traffic Manager (load balancing/health check-DNS layer)

Load balancing services:
Azure Load Balancer
Application Gateway (Windows Application Firewall)

Web app: front-end based on AppServices



Backend (may include load balancing services as well)

- Virtual Machines/Service Fabric/Docker Containers/..
- SQL Azure
- Hadoop Cluster
- NoSQL DB
- ….

High availability services

DNS level :

- Azure DNS
- Traffic Manager

Application level:

- Azure Internal Load Balancer (ALB)
- Application Gateway
- Web Application Firewall (WAF)

High availability services

Service	Azure Load Balancer	Application Gateway	Traffic Manager
Technology	Transport level (level 4)	Application level (level 7)	DNS level
Application protocols supported	Any	HTTP and HTTPS	Any (An HTTP/S endpoint is required for endpoint monitoring)
Endpoints	Azure VMs and Cloud Services role instances	Any Azure Internal IP address or public internet IP address	Azure VMs, Cloud Services, Azure Web Apps and external endpoints
Vnet support	Can be used for both Internet facing and internal (Vnet) applications	Can be used for both Internet facing and internal (Vnet) applications	Only supports Internet-facing applications
Endpoint Monitoring	supported via probes	supported via probes	supported via HTTP/HTTPS GET

Azure DNS

- Azure DNS is a hosting service for DNS domains
- DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers.
- We use Anycast networking DNS query is answered by the closest available DNS server.
- Currently only domain delegation is supported (??)
- Azure DNS supports all common DNS record types, including A, AAAA, CNAME, MX, NS, SOA, SRV, and TX, as well as wildcards.
- Note: you can buy domain via Azure.com (somewhere I saw Buy button)

Azure Traffic Manager

Works on DNS level, best scenarios:

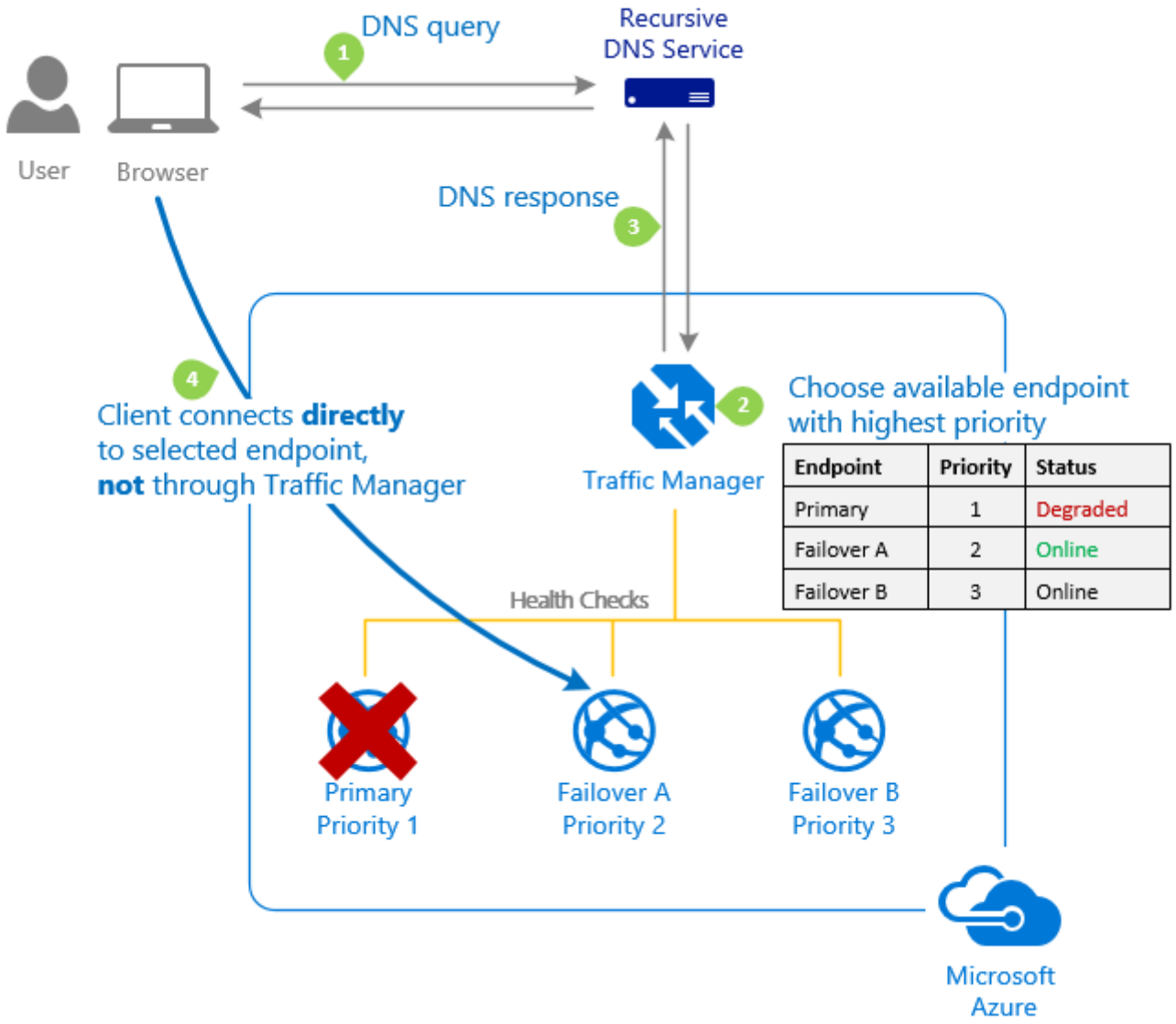
- **Improve availability of critical applications**
- **Improve responsiveness for high performance applications** – allows you to run cloud services or websites in datacenter (any hosting, not limited to Azure .
- **Upgrade and perform service maintenance without downtime**
- **Combine on-premises and Cloud-based applications** – Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including the “burst-to-cloud,” “migrate-to-cloud,” and “failover-to-cloud” scenarios.
- **Distribute traffic for large, complex deployments** – Traffic-routing methods can be combined using nested Traffic Manager profiles

Azure Traffic Manager

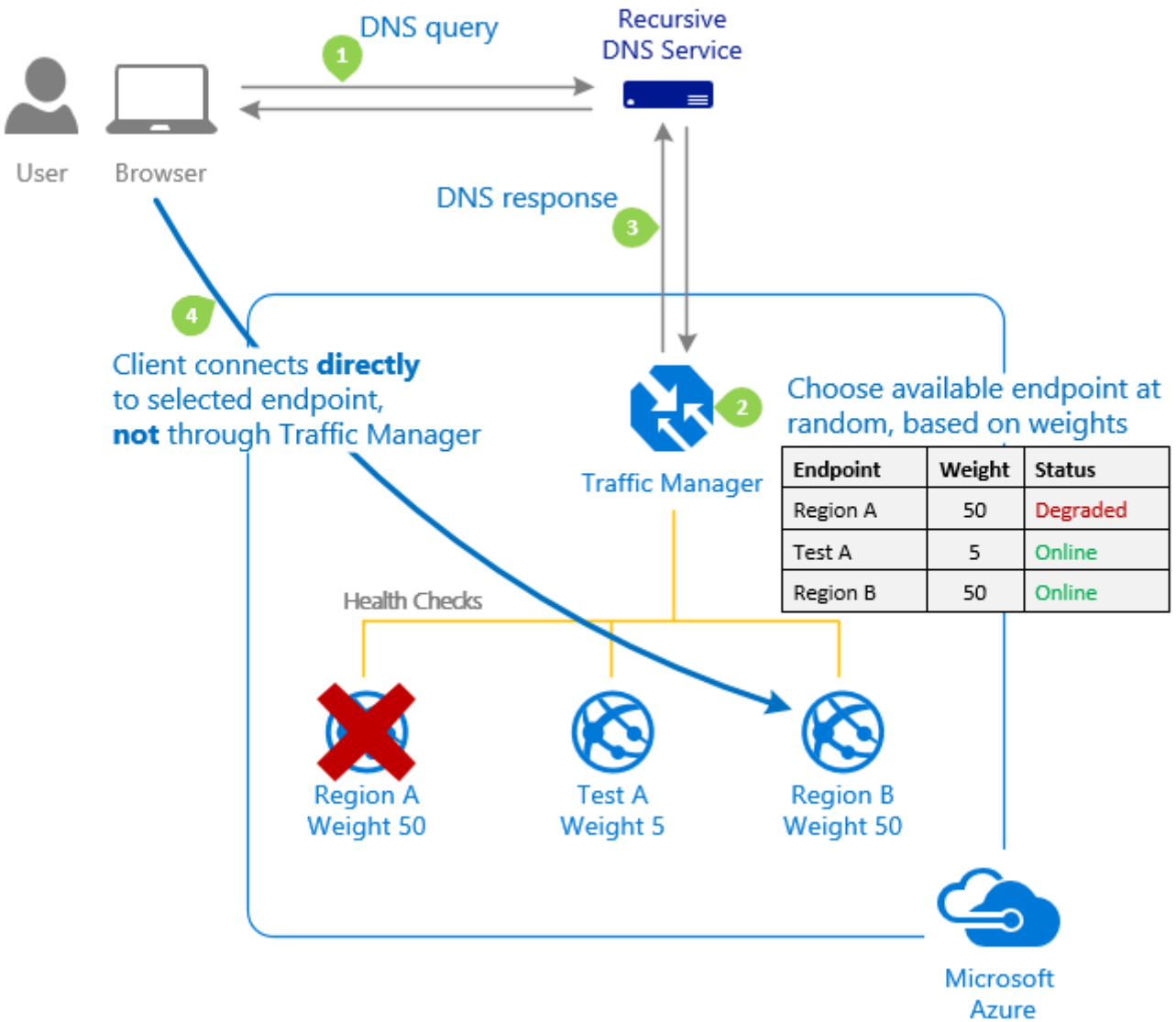
Traffic routing methods available in Traffic Manager:

- **Priority**: Select ‘Priority’ when you want to use a primary service endpoint for all traffic, and provide backups in case the primary or the backup endpoints are unavailable. For more information, see Priority traffic-routing method.
- **Weighted:** Select ‘Weighted’ when you want to distribute traffic across a set of endpoints, either evenly or according to weights which you define. For more information, see Weighted traffic-routing method.
- **Performance:** Select ‘Performance’ when you have endpoints in different geographic locations and you want end users to use the "closest" endpoint in terms of the lowest network latency. For more information, see Performance traffic-routing method.

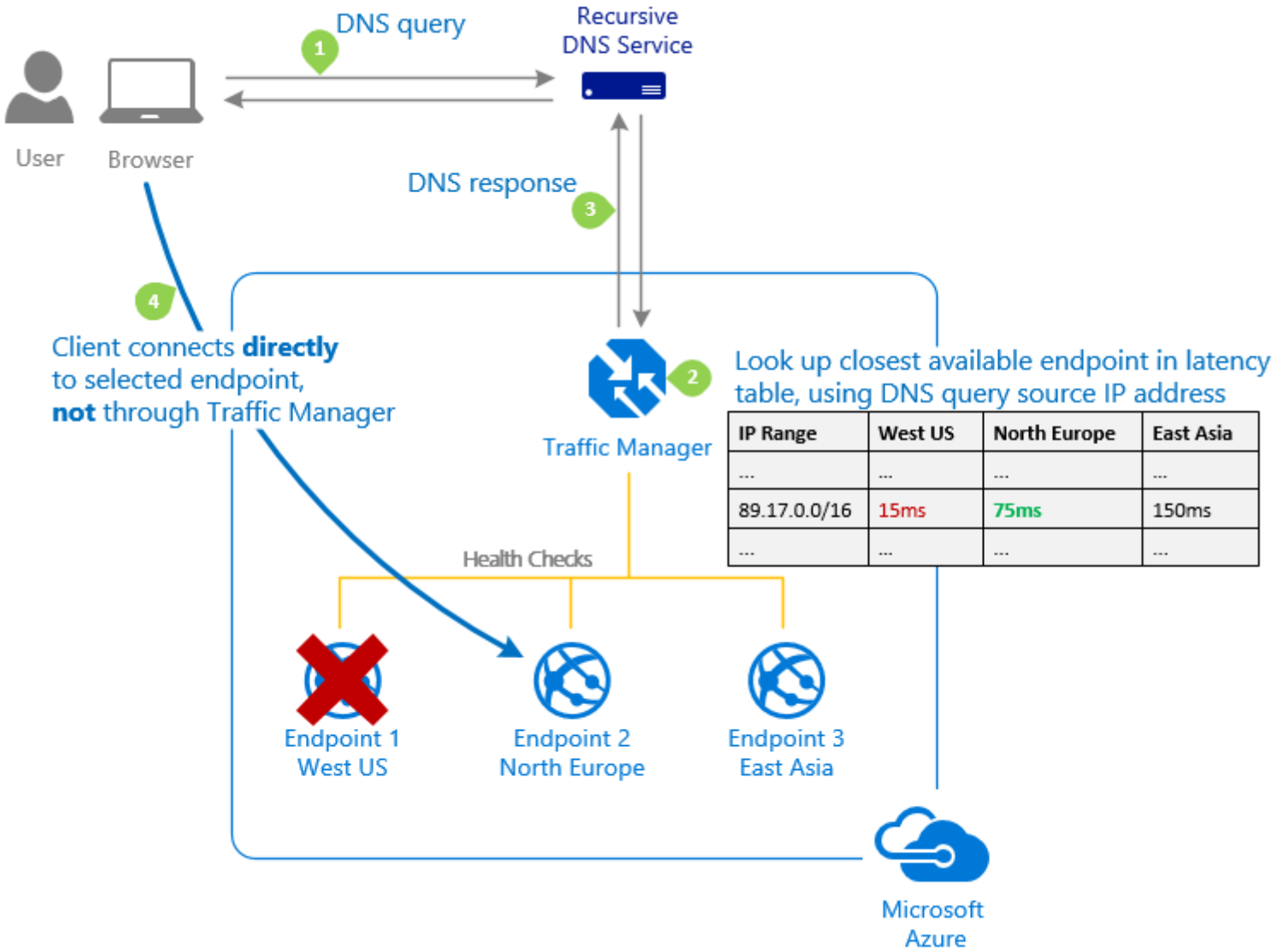
Azure Traffic Manager: priority routing



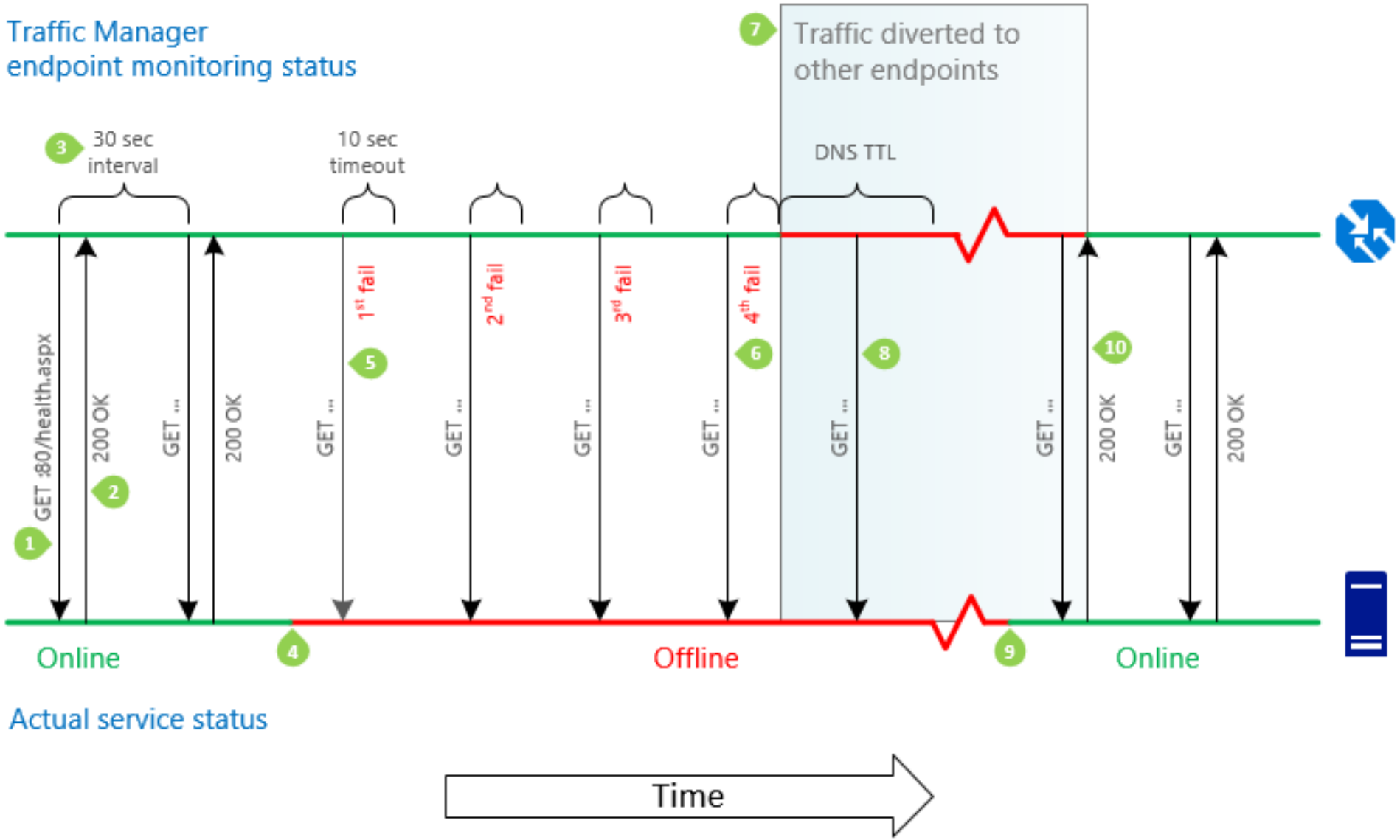
Azure Traffic Manager: weighted routing



Azure Traffic Manager: performance routing



Azure Traffic Manager: example



Azure Internal Load Balancer

Azure Load Balancer delivers high availability and network performance to your applications. It is a Layer 4 (TCP, UDP) load balancer that distributes incoming traffic among healthy instances of services defined in a load-balanced set. o

Azure Load Balancer configuration:

- Load balance incoming Internet traffic to virtual machines. This configuration is known as Internet-facing load balancing.
- Load balance traffic:
  - Between virtual machines in a virtual network
  - between virtual machines in cloud services
  - between on-premises computers and virtual machines in a cross-premises virtual network (internal load balancing)
- Forward external traffic to a specific virtual machine.

All resources in the cloud need a public IP address to be reachable from the Internet. Within the cloud infrastructure, Microsoft Azure uses non-routable IP addresses for its resources. Azure uses network address translation (NAT) with public IP addresses to communicate to the Internet.

Azure Load Balancer features

Hash-based distribution :

- By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, and protocol type) hash to map traffic to available servers.
- Stickiness only within a transport session.
- Packets in the same TCP or UDP session will be directed to the same instance behind the load-balanced endpoint.
- When the client closes and reopens the connection or starts a new session from the same source IP, the source port changes. This may cause the traffic to go to a different endpoint in a different datacenter.

Port forwarding

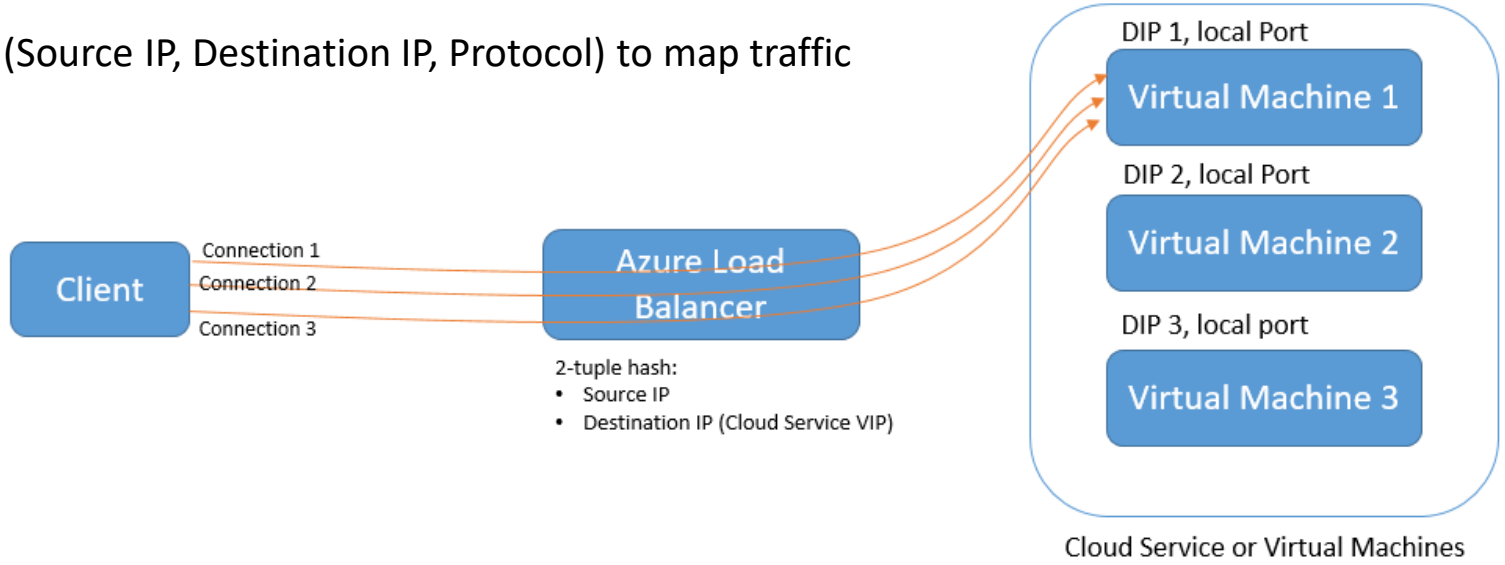
Automatic reconfiguration during scale up/down

Service monitoring by probes:

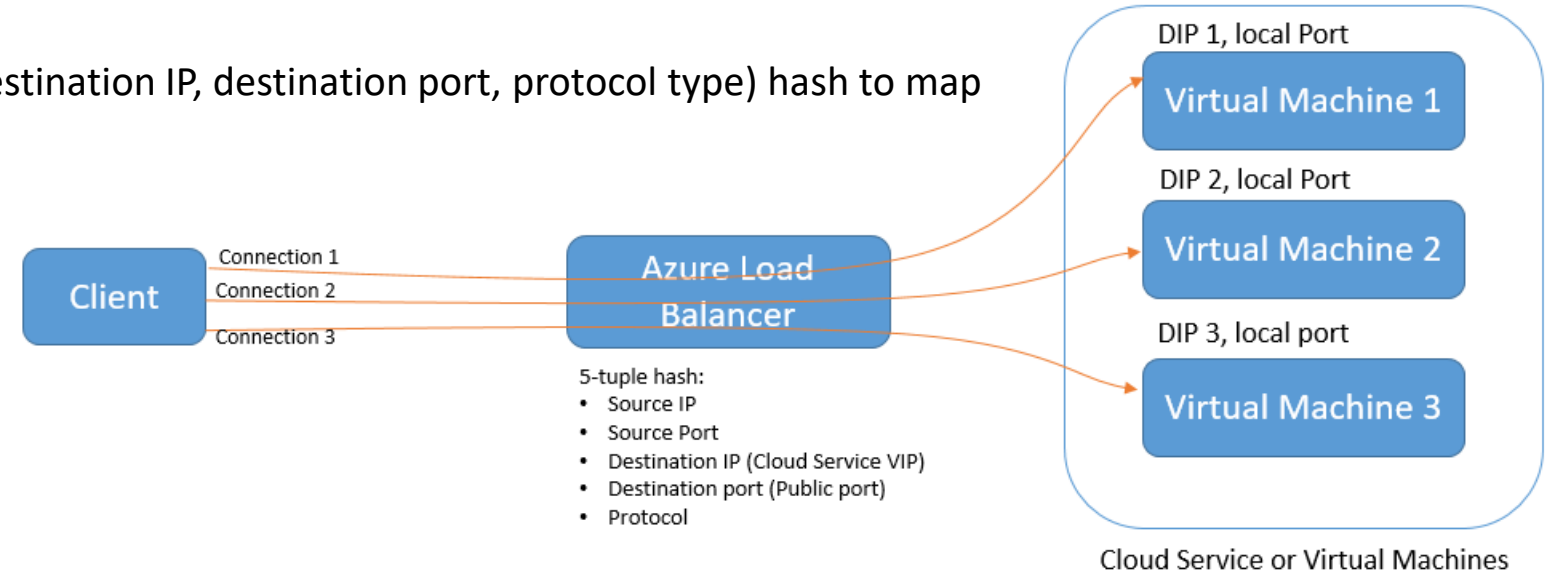
- **Guest agent probe (on PaaS VMs only):** utilizes the guest agent inside the virtual machine – check to HTTP 200
- **HTTP custom probe:** Probe your endpoint on instance each 15 sec for TCP ACK or HTTP 200 within the timeout period.
- **TCP custom probe:** relies on successful TCP session establishment to a defined probe port.

Azure Load Balancer distribution modes

IP Affinity mode: use (Source IP, Destination IP, Protocol) to map traffic

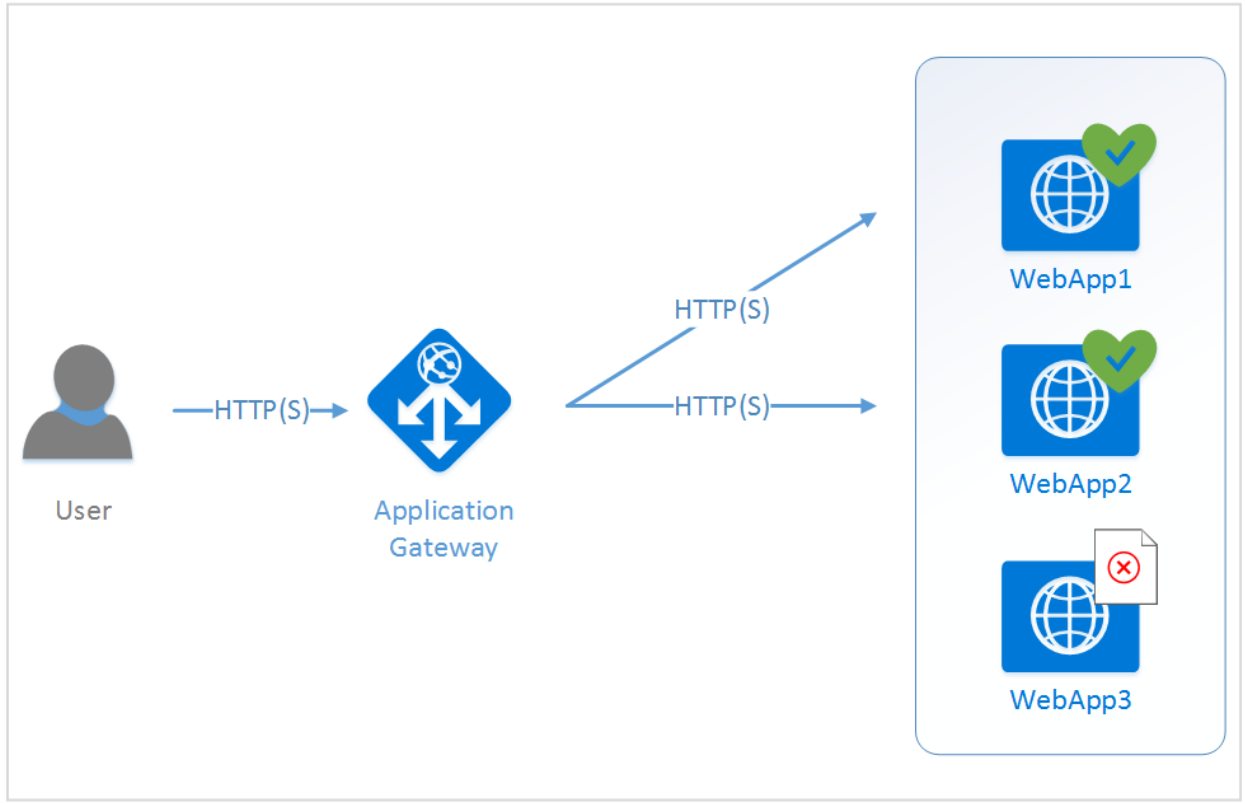


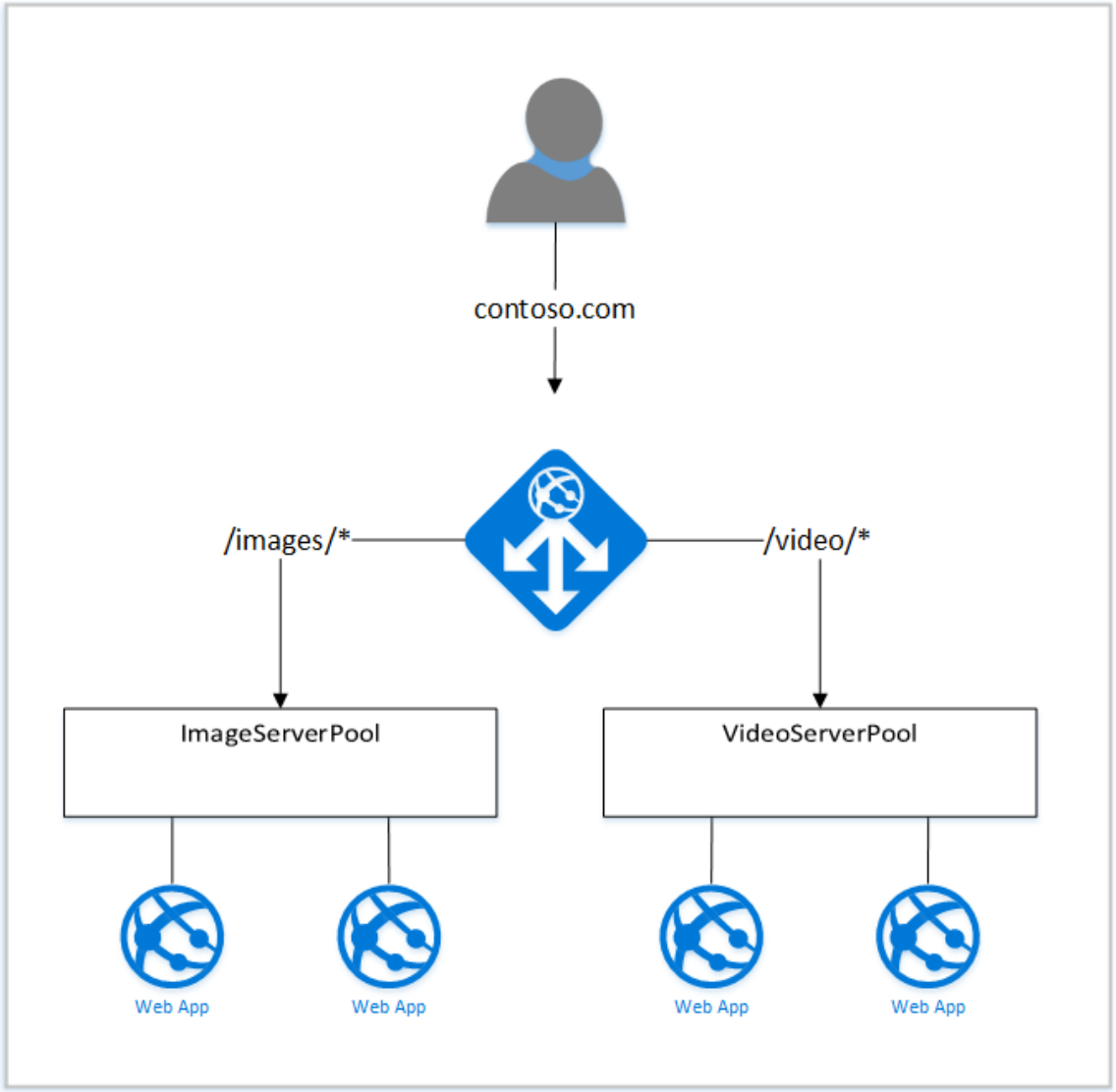
Hash based mode: use (source IP, source port, destination IP, destination port, protocol type) hash to map traffic, stickiness only within a transport session



Azure Application Gateway

Requests balancing/routing, firewall features and health checking





Azure Application Gateway

Application Gateway currently supports layer-7 application delivery for the following:

- **HTTP load balancing**
- **Cookie-based session affinity**
- **Secure Sockets Layer (SSL) offload**
- **URL-based content routing**
- **Multi-site routing**

HTTP layer 7 load balancing is useful for:

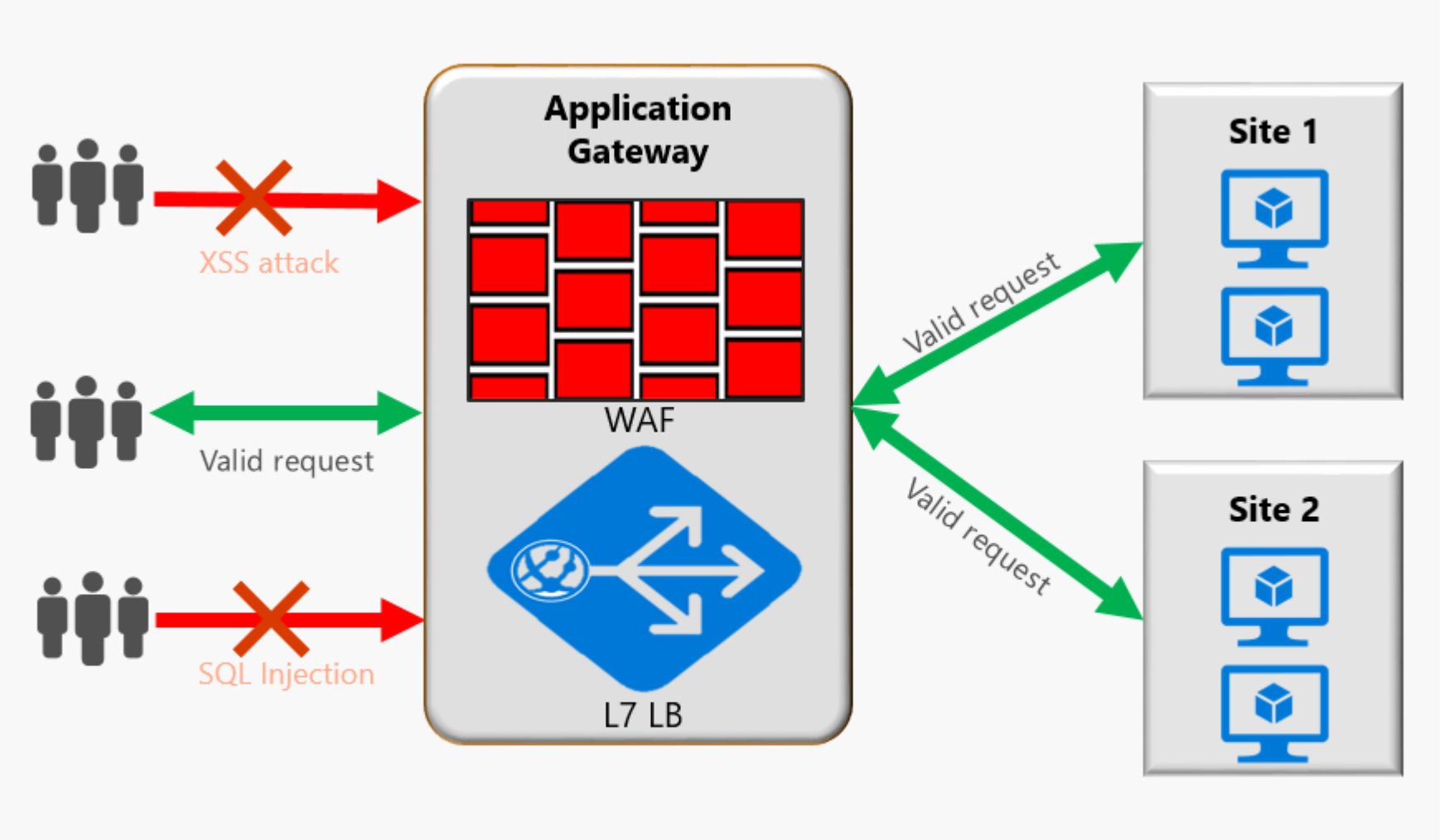
- Applications that require requests from the same user/client session to reach the same back-end virtual machine.Examples of these applications would be shopping cart apps and web mail servers.
- Applications that want to free web server farms from SSL termination overhead.
- Applications, such as a content delivery network, that requires multiple HTTP requests on the same long-running TCP connection to be routed or load balanced to different back-end servers.

Azure Application Gateway

The following table shows an average performance throughput for each application gateway instance:

Back-end page response	Small	Medium	Large
6 K	7.5 Mbps	13 Mbps	50 Mbps
100 K	35 Mbps	100 Mbps	200 Mbps

Azure Application Gateway: Web Application Firewall (WAF)



Web Application Firewall

Baseline protection against most of the OWASP top 10 common web vulnerabilities.

- SQL injection protection
- Cross site scripting protection
- Common Web Attacks Protection such as command injection, HTTP request smuggling, HTTP response splitting, and remote file inclusion attack
- Protection against HTTP protocol violations
- Protection against HTTP protocol anomalies such as missing host user-agent and accept headers
- HTTP DoS Protections including HTTP flooding and slow HTTP DoS prevention
- Prevention against bots, crawlers, and scanners
- Detection of common application misconfigurations (i.e. Apache, IIS, etc)

OWASP = Open Web Application Security Project - owasp.org/

Web Application Firewall

Application Gateway WAF can be configured to run in the following two modes:

Detection mode – WAF monitors and logs all threat alerts into a log file:

- Logging diagnostics for Application Gateway should be ON in Diagnostics section.
- WAF log is selected and turned ON

Prevention mode – blocks intrusions and attacks detected by its rules:

- The attacker receives a 403 unauthorized access exception and the connection is terminated.
- Prevention mode continues to log such attacks in the WAF logs.

Another option for Firewall is to use Barracuda application firewall in front of web facing apps.

Application Gateway vs Load Balancer

Type	Azure Load Balancer	Application Gateway
Protocols	UDP/TCP	HTTP/ HTTPS
IP reservation	Supported	Not supported
Load balancing mode	5 tuple(source IP, source port, destination IP,destination port, protocol type)	CookieBasedAffinity = false,rules = basic (Round-Robin)
Load balancing mode (source IP /sticky sessions)	2 tuple (source IP and destination IP), 3 tuple (source IP, destination IP and port). Can scale up or down based on the number of virtual machines	CookieBasedAffinity = true,rules = basic (Roud-Robin) for new connections.
health probes	Default: probe interval - 15 secs. Taken out of rotation: 2 Continuous failures. Supports user defined probes	Idle probe interval 30 secs. Taken out after 5 consecutive live traffic failures or a single probe failure in idle mode. Supports user defined probes
SSL offloading	not supported	supported

Azure App Services

Web App

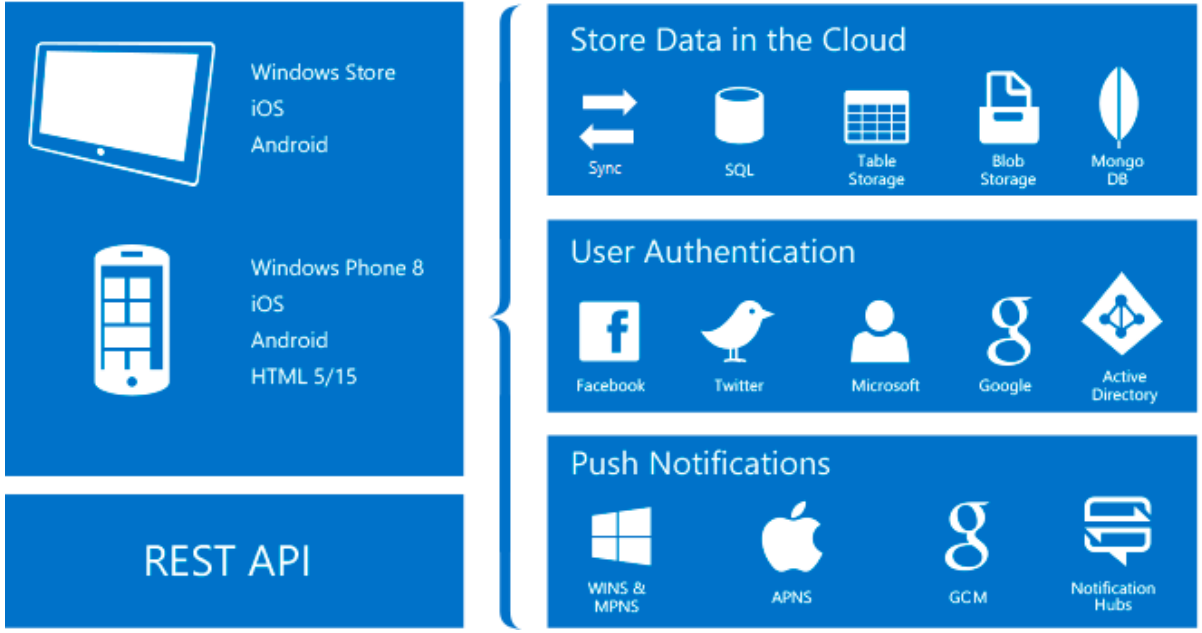
Mobile App

Function App

API App

Logic App

AppServices: Mobile Apps



- **Authentication and Authorization** - based on Oauth 2.0: Azure Active Directory for enterprise authentication, Facebook, Google, Twitter, Microsoft Account…

- **Data Access** - OData v3 data source linked to SQL Azure/on-premises SQL Server. Entity Framework with integration to noSQL and SQL data providers, including Azure Table Storage, MongoDB, DocumentDB and SaaS API providers like Office 365 and Salesforce.com.

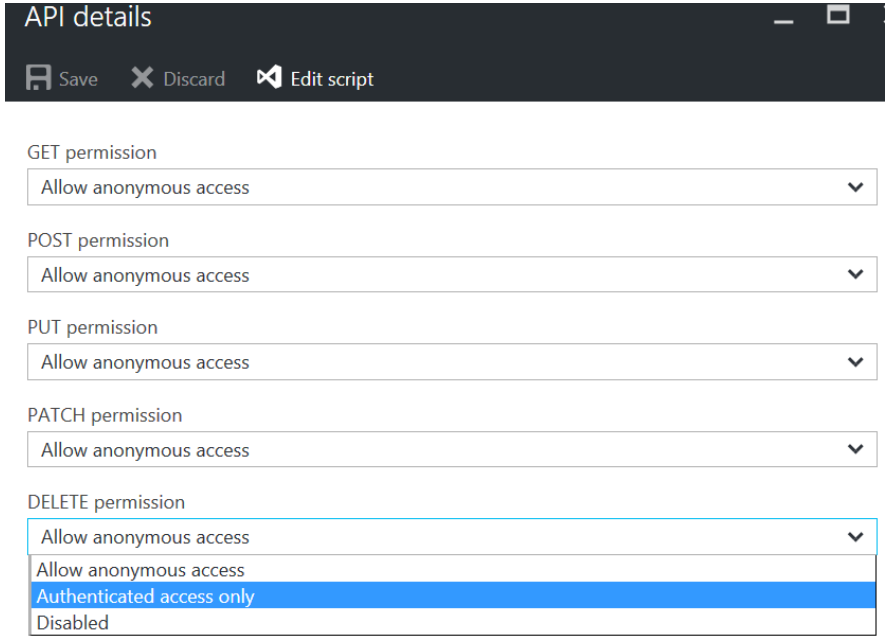
- **Offline Sync** - Our Client SDKs helps to build apps that operate with an offline data set that can be automatically synchronized with the backend data, including conflict resolution support.

- **Push Notifications** – Azure Notification Hubs, allowing you to send push notifications to millions of users simultaneously.

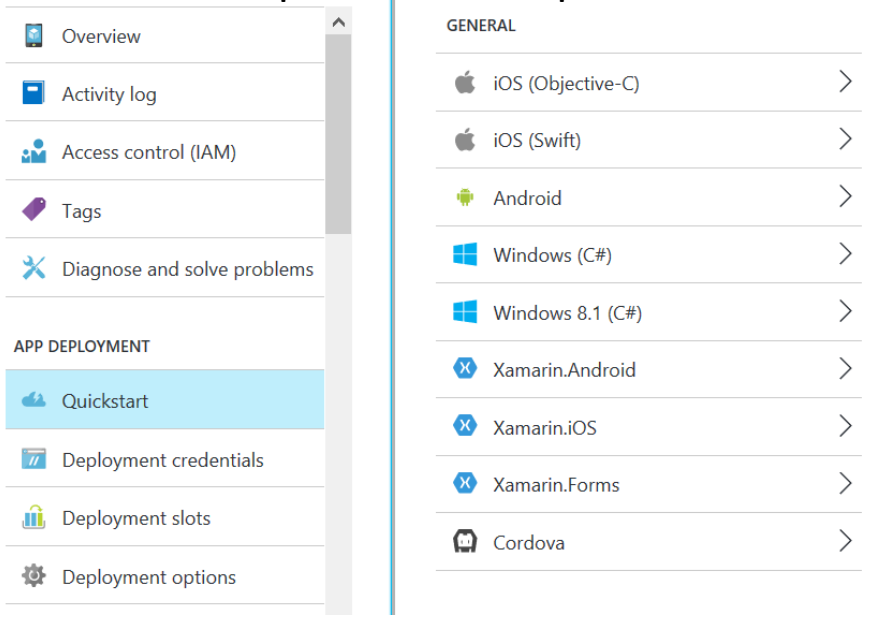
- **Client SDKs** - Native development (iOS, Android, Windows), cross-platform development (Xamarin for iOS and Android, Xamarin Forms) and hybrid application development (Apache Cordova). Each client SDK is available with an MIT license and is open-source.

App Services: Mobile Apps

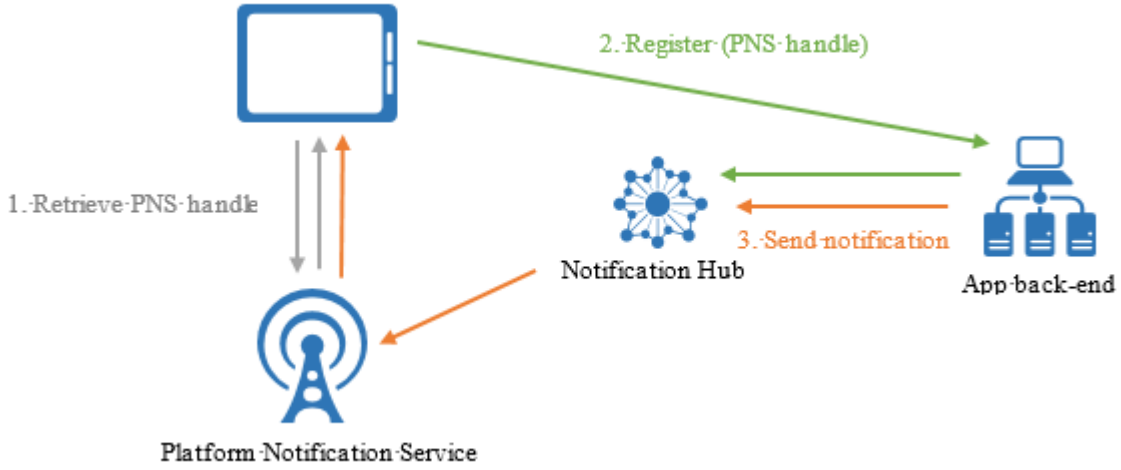
Easy management for API calls and handlers



Built-in portal features – quick start templates for client apps



Integration with Notification Hub to manage Push notifications



Azure Notification Hub

1. **Multiple platforms:**
    - Support for all major mobile platforms
    - Common interface to send notifications to all supported platforms.
    - Device handle management.
2. **Works with any back-end**: Cloud or on-premises, any programming language .NET, PHP, Java, Node…
3. **Scale:** Notification hubs scale to millions of devices without the need to re-architect or shard.
4. **Rich set of delivery patterns**:
    - *Broadcast*: allows for near-simultaneous broadcast to millions of devices with a single API call.
    - *Unicast/Multicast*: Push to tags representing individual users, including all of their devices; or wider group; for example, separate form factors (tablet vs. phone).
    - *Segmentation*: Push to complex segment defined by tag expressions (for example, devices in New York following the Yankees).
5. **Personalization**: Each device can have one or more templates, to achieve per-device localization and personalization without affecting back-end code.
6. **Security**: Shared Access Secret (SAS) or federated authentication.

AppServices: Management API

- Reuse. Support legacy APIs without rewriting them
- Access without headaches. Provide developers with access to APIs easily and automatically
- Security. Secure backend systems and protect APIs against rogue use or overuse
- Visibility. Measure which APIs are being used, how much, and by whom

API gateway:

- Accepts API calls and routes them to your backends.
- Verifies API keys, JWT tokens, certificates, and other credentials.
- Enforces usage quotas and rate limits.
- Transforms your API on the fly without code modifications.
- Caches backend responses where set up.
- Logs call metadata for analytics purposes.

Publisher portal is the administrative interface where:

- Define or import API schema, package APIs into products.
- Set up policies like quotas or transformations on the APIs.
- Get insights from analytics
- Manage users.

Developer portal :

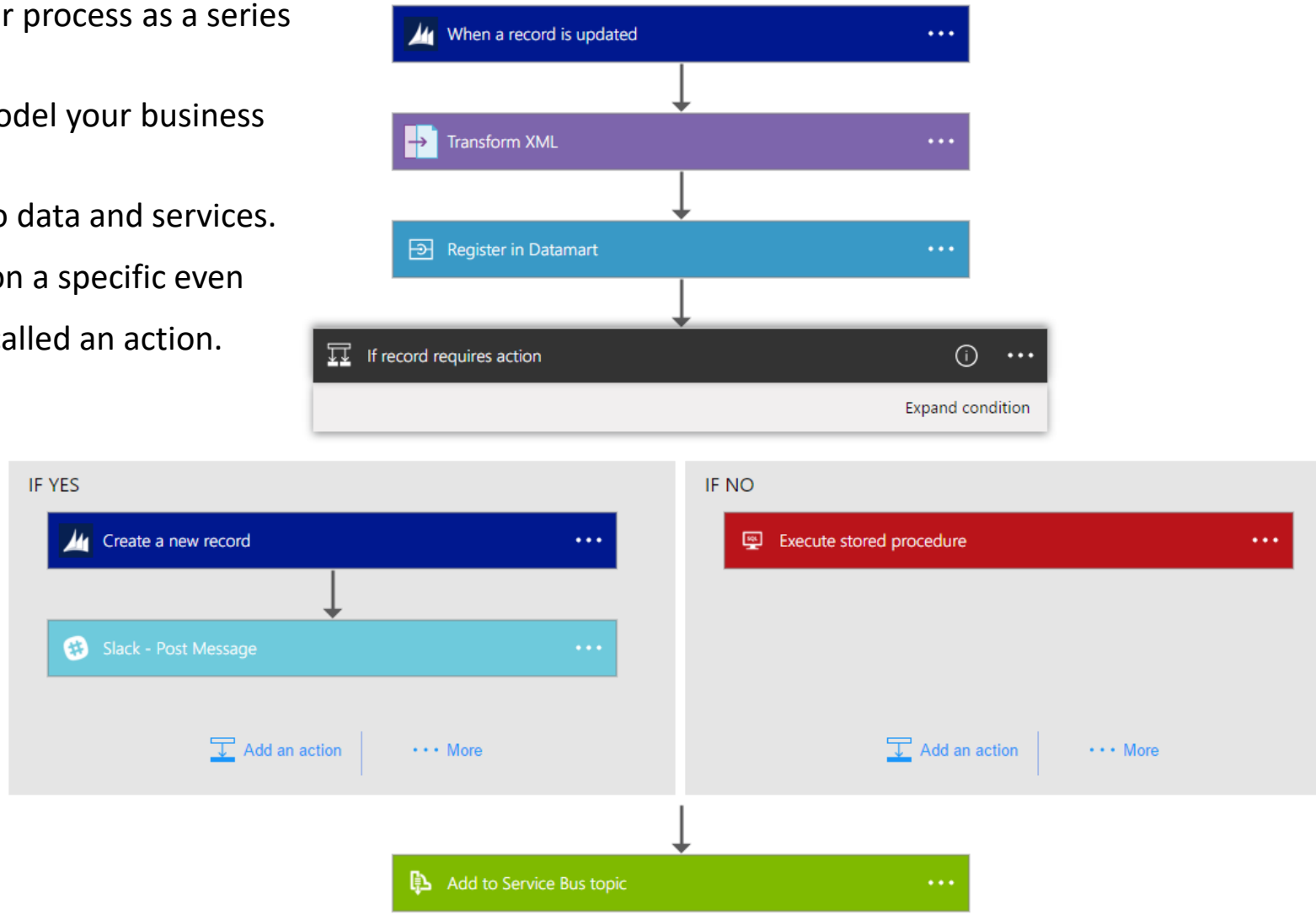
- Try out an API via the interactive console.
- Create an account and subscribe to get API keys.

App Services: Logic Apps

Logic Apps - way to simplify and implement scalable integrations and workflows in the cloud.

It provides a visual designer to model and automate your process as a series of steps i.e. workflow.

- **Workflow** - Logic Apps provides a graphical way to model your business processes as a series of steps or a workflow.
- **Managed Connectors** - Your logic apps need access to data and services.
- **Triggers** - starts a new instance of a workflow based on a specific even
- **Actions** - Each step after the trigger in a workflow is called an action.
- **Enterprise Integration Pack** - For more
- advanced integration scenarios, Logic Apps
- includes capabilities from BizTalk.



App Service plans

Recommended for try/dev/test or personal usage.

- Free = try for free
- Basic = host very basic applications
- Shared = more advanced for dev/test

Recommended for commercial production workloads

- Standard = Go live with web and mobile
- Premium = Enterprise scale and integration

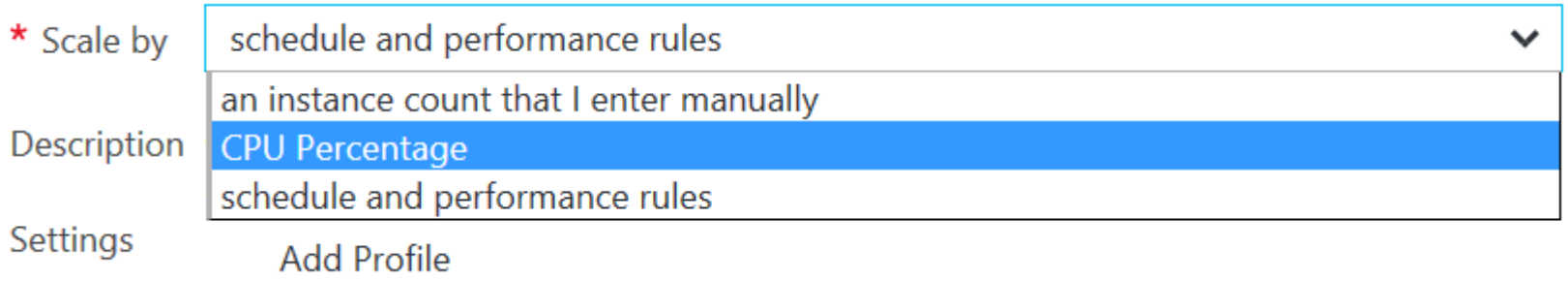
	Free	Basic	Shared	Standard	Premium
Number of apps	10	100	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	500 GB
Max instances	--	--	Up to 3	Up to 10	Up to 50
SLA	--	--	99.95%	99.95%	99.95%
Auto scale	--	--	--	Supported	Supported
Geo-distributed environment	--	--	--	Supported	Supported
VPN connectivity	--	--	--	Supported	Supported

App Service plans: continue

	Free	Basic	Shared	Standard	Premium
Custom domain	10	100	Unlimited	Unlimited	Unlimited
SSL certificates	--	--	Unlimited SNI SSL certs	Unlimited SNI SLL + 1 IP SSL included	Unlimited SNI SLL + 1 IP SSL included
Auto backups/day	--	--	Up to 3	Up to 10	Up to 50
Active mobile devices	500 / day	500 / day	Unlimited	Unlimited	Unlimited
Offline sync/day	500 calls	1k calls	1k calls	Unlimited	Unlimited
Staging environment	--	--	--	5	20

App Service plans: continue

Autoscale : by schedule or automatically by CPU level (average for all instances!).



Geo-distributed environment: You can use different geo-locations for web-sites under same Service Plan.

VPN: Site-to-Site and Point-to-Site VNETs connection

Auto backups: backup App configuration, file content and Azure SQL Database/Azure MySQL connected to app, into storage account by schedule.

Active mobile devices: related to push/call notifications for mobile devices via Notification Hub.

Offline sync: sync for mobile apps under Azure Mobile Apps, for offline mode.

Staging environments: support for staging, including split requests for different environments in the same time

Azure Websites Features & Capabilities

Enterprise-class
Designed for secure mission-critical applications

- Hybrid Connections / VPN Support
- Scheduled Backup
- Azure Active Directory Integration
- Site Resiliency, HA, and DR
- Web Jobs
- Role Base Access Control
- Audit / Compliance
- Enterprise Migration
- Client Certs
- Redis Caching
- IP Restrictions/ SSL
- Web Sockets
- SQL, MySQL, DocDB, & Mongo

Global scale
Optimized for Availability and Automatic scale

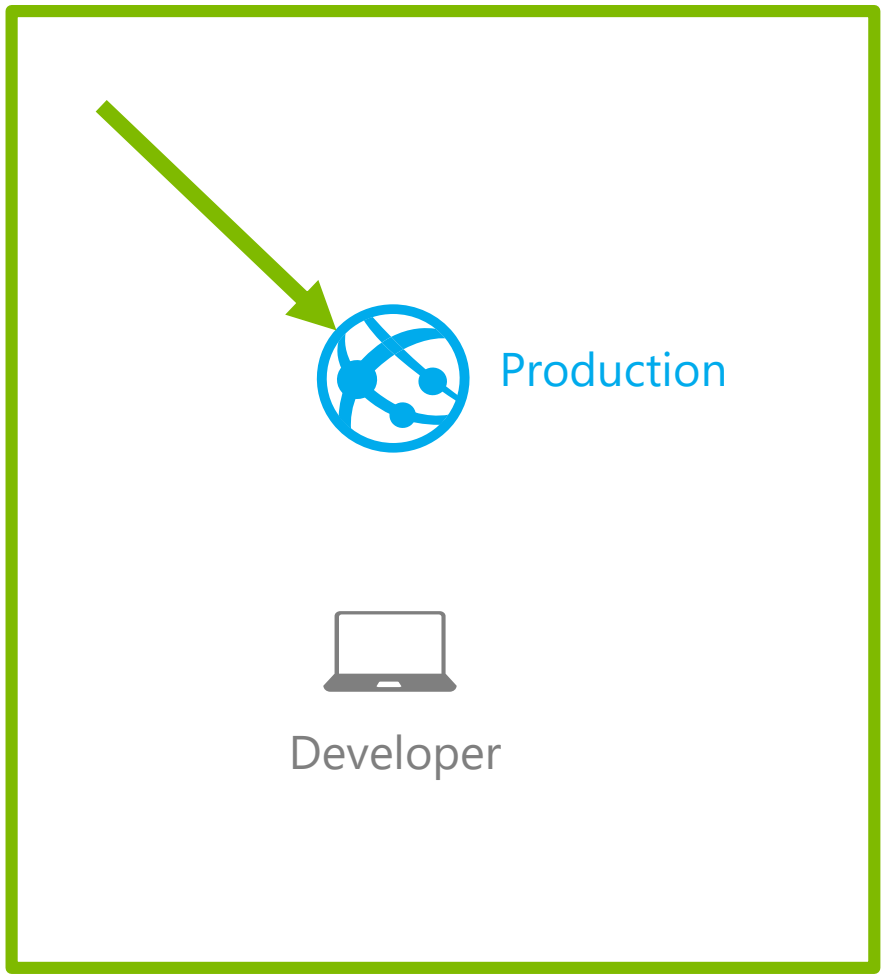
- Automated Deployment
- AutoScale
- Built-in Load Balancing
- WW Datacenter Coverage
- End Point Monitoring & Alerts
- App Gallery
- DR Site Support
- WildCard Support
- Dedicated IP address
- HTTP Compression
- WebJobs
- Sticky Sessions

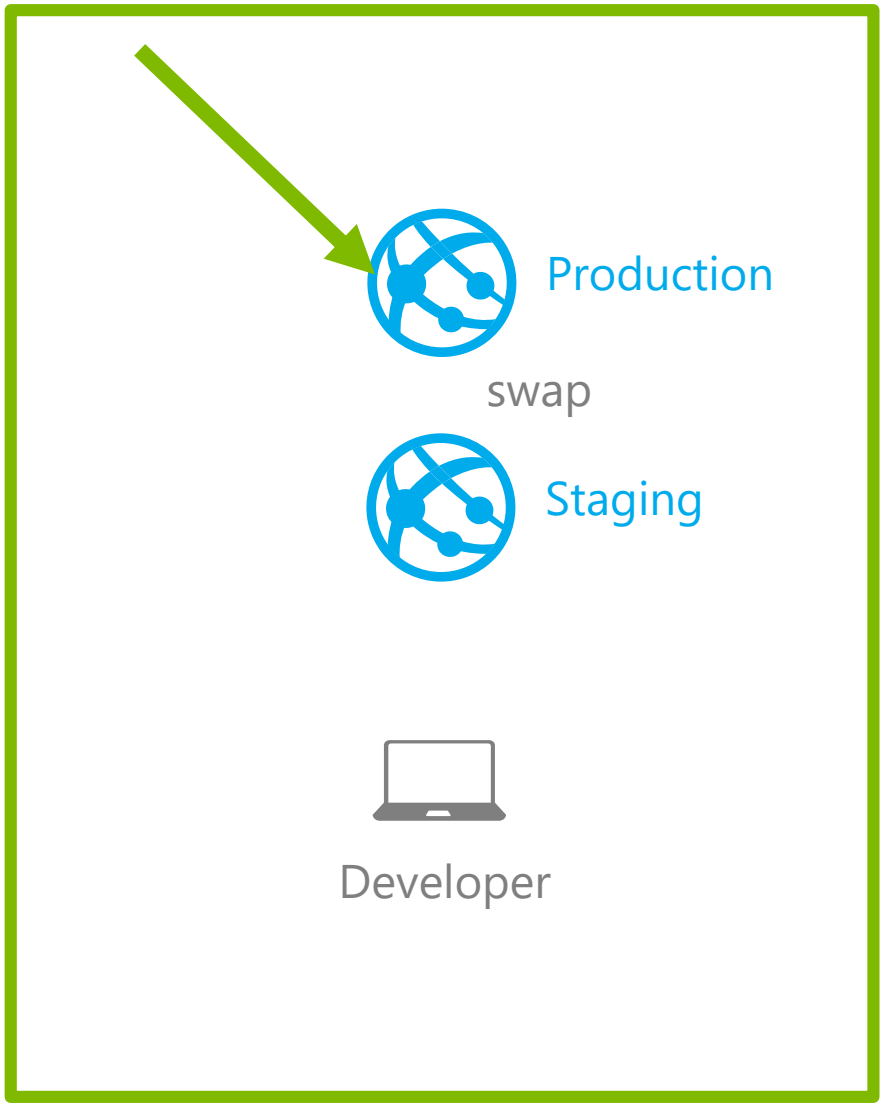
Built for DevOps
Agility through Continuous Deployment

- Remote Debugging w/ Visual Studio
- Site Staging Slots
- Testing in Production
- Continuous Integration/Deployment
- Git, Visual Studio Online and GitHub
- App & Site Diagnostics
- OS & Framework Patching
- Site Extensions Gallery
- NET, PHP, Python, Node, Java
- Framework Installer
- Browser-based editing
- Auto-Healing
- Logging and Auditing

I- Development and Deployment

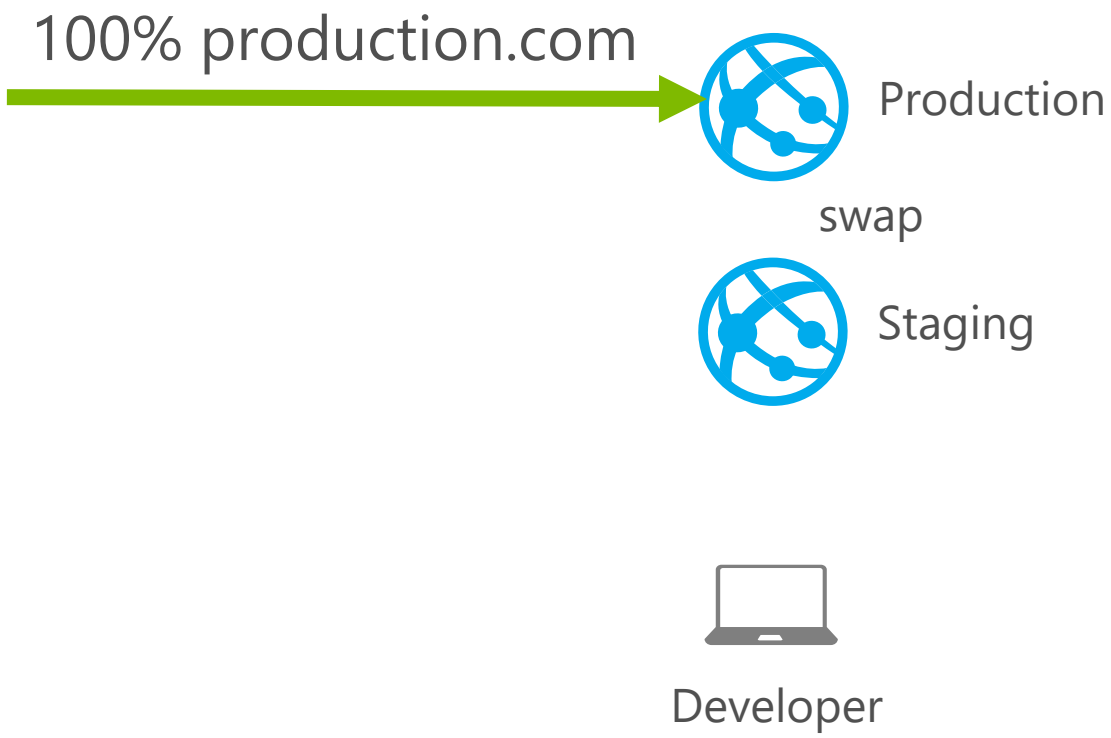
Staged Publishing





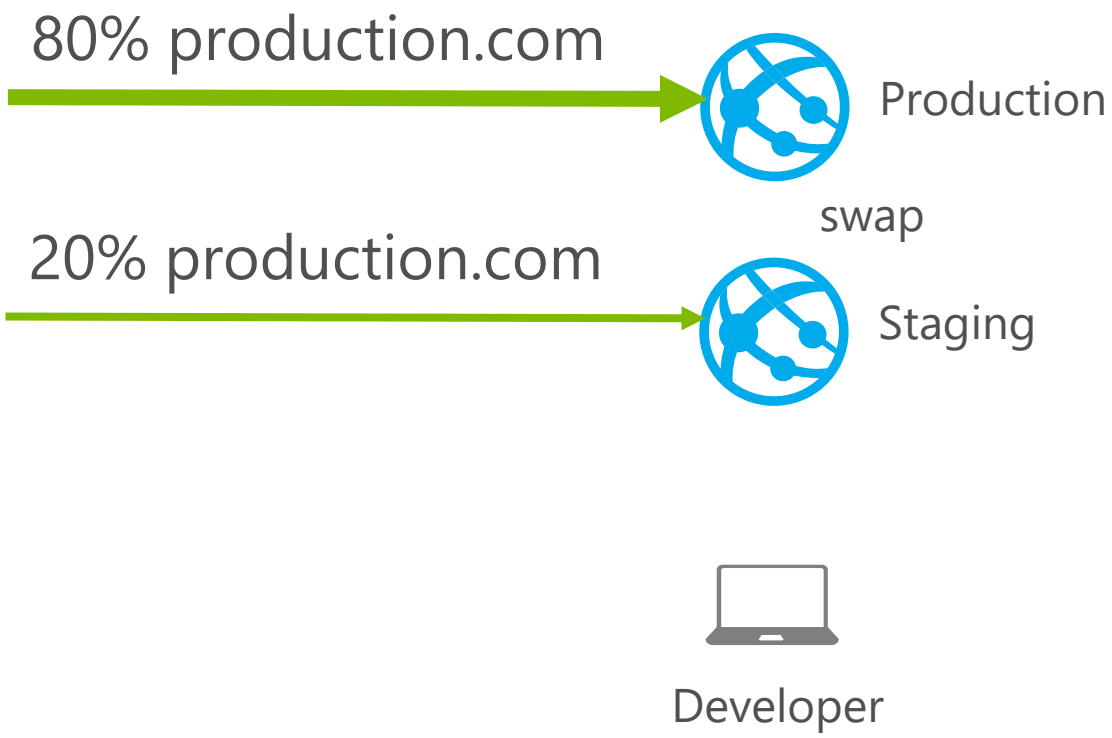
I- Development and Deployment

Testing in Production



I- Development and Deployment

Testing in Production

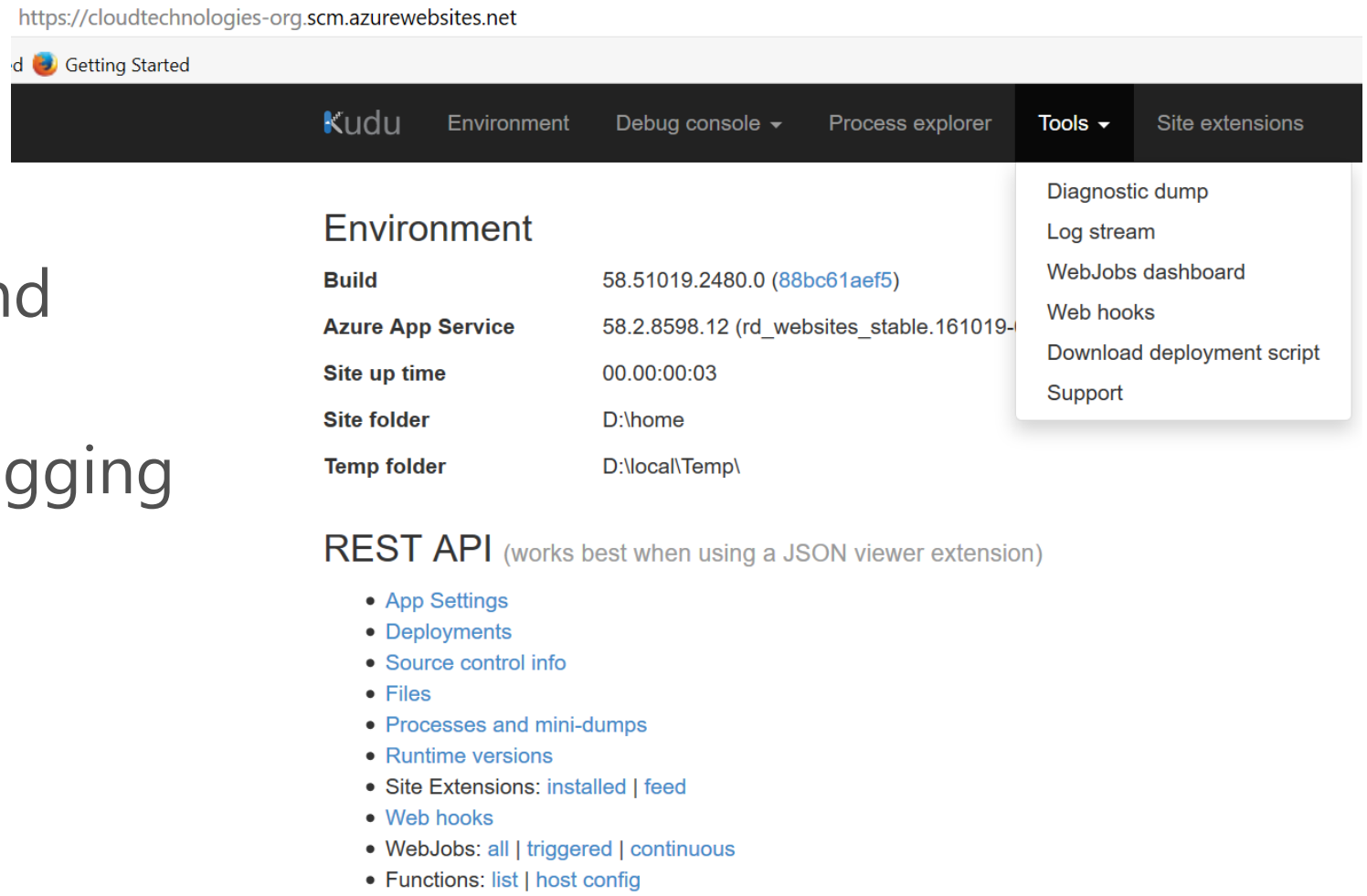


II- Debugging and Investigations

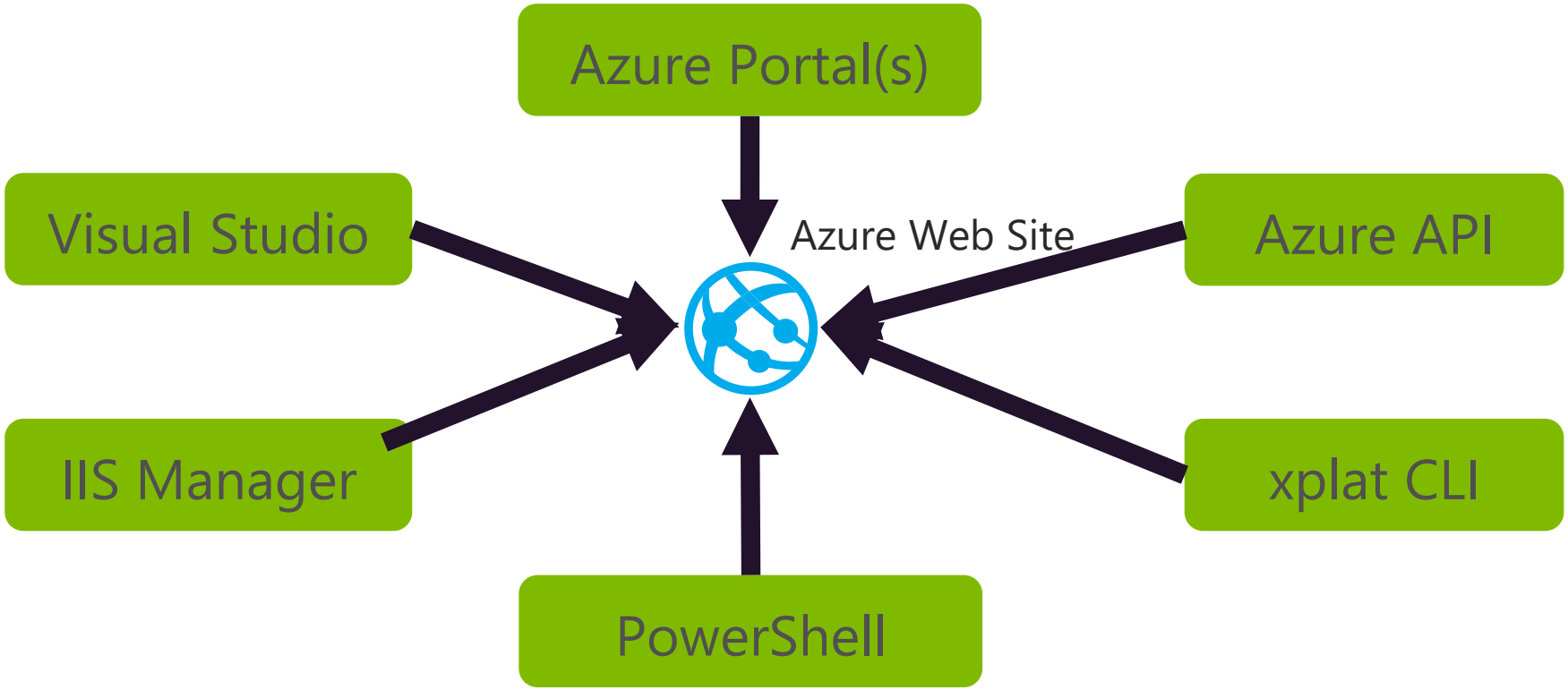
Kudu

https://<mySite>.**scm**.azurewebsites.net/

* Authenticated
* Runs in the same security context as the main site
* Can access the site files and environment variables
* Great for admin and debugging



III- Management and Administration



Создание webjobs

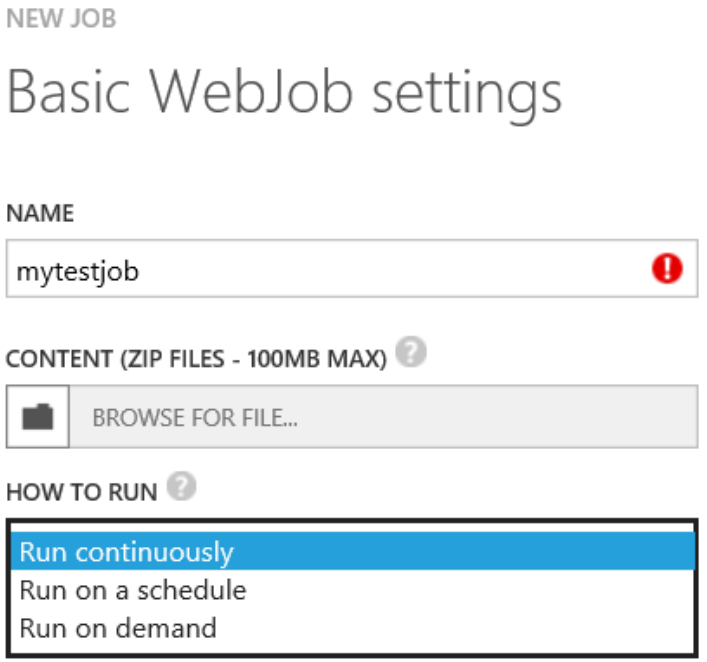
Через ftp :

- Для triggered job копируем все в путь (внутри вебсайта): site/wwwroot/app_data/jobs/triggered/{job name}
- Для continuous job копируем все в путь (внутри вебсайта): site/wwwroot/app_data/jobs/continuous/{job name}

- Используя SDK REST API

(например PUT /api/zip/site/wwwroot/App_Data/jobs/continuous/{job name}/)

- Через портал azure.com – загружаем zip файл



Процесс выполнения webjobs

- Архив распаковывается во временную папку (путь к ней в переменной %WEBJOBS_PATH%)
- Сначала выполняется поиск файла по маске run.{file type extension} (например run.cmd или run.exe)
- Если ничего не найдено для всех типов файлов, то тогда выполняется поиск первого файла с поддерживаемым расширений в указанном порядке: .cmd, .bat, .exe, .ps1, .sh, .php, .py, .js
- Поиск выполняется только в корневой папке
- Реомендованное имя файла для запуска задач = run.cmd

- .cmd, .bat, .exe (using windows cmd)
- .ps1 (using powershell)
- .sh (using bash)
- .php (using php)
- .py (using python)
- .js (using node)
- .jar (using java)

Размещение внутри сайта

- Файловая система: site\wwwroot\App_Data\jobs\{job type}\{job name}
- Снаружи (через POST) **https://{sitename}.scm.azurewebsites.net/api/webjobs/triggered/{webjobname}/run**
- Веб интерфейс: **https://{sitename}.scm.azurewebsites.net/azurejobs/#/jobs**
- Данные веб-сайта доступны: d:\home (%HOME%)
- При запуске WebJobs копируется во временную папку (см %WEBJOBS_PATH%) и там запускается
- Для хранения и обработки данных рекомендуется использовать папку:

  %WEBROOT_PATH%\App_Data\jobs\%WEBJOBS_TYPE%\%WEBJOBS_NAME%

или %WEBJOBS_DATA_PATH%

- Настройки конфигурирования (GET) : **https://{sitename}.scm.azurewebsites.net/api/webjobs**

Настройки

- WEBJOBS_RESTART_TIME – время рестарта в секундах, после завершения предыдещего запуска (независимо от статуса завершения) – только для continuous jobs
- WEBJOBS_IDLE_TIMEOUT – время бездействия в секундах, при превышении этого времении задача будет принудительно остановлена. Бездействие определяется по отсутствию нагрузки на CPU и операциям ввода-вывода на консоль/логи. Только для triggered jobs
- WEBJOBS_HISTORY_SIZE – размер истории запусков, только для triggered jobs
- WEBJOBS_STOPPED – если установлено в 1, то запуск задач запрещен и все задачи (включая уже запущенные) принудительно останавливаются.

Управление остановкой задач

В ряде случаев требуется принудительная остановка задачи и в этом случае существуем механизм сообщения задаче о такой остановке перед тем как она будет удалена.

Это может происходить в случаях:

- Рестарта/остановки/изменение настроек веб-сайта/web.config
- Остановки задачи (через API)
- Обновления файлов webjob

Как это работает:

- Continuous : если требуется принудительная остановка то создается файл %WEBJOBS_SHUTDOWN_FILE% и задаче дается 5 секунд на остановку. Соответственно внутри задачи должен быть мониторинг наличия этого файла как флага принудительной остановки через 5 секунд
- Triggered: по умолчанию 30 секунд ожидания

Управление запуском через settings.job

Этот файл должен находится в корневой папке там же где и выполняемый файл.

- По умолчанию для всех задач (кроме .JS) при каждом запуске выполняется копирование всех файлов в %TEMP%\jobs\{job type}\{job name}\{random name} где и происходит запуск задачи.
- Вторая опция называется in place – при этом запуск производится в том же месте где выложен код webjobs (wwwroot/app_data/job,….) Это устанавливается в settings.job при помощи { "is_in_place": true }
- Дополнительно в этом файле можно менять время ожидания принудительно остановки задач

{ "stopping_wait_time": 60 }

Переменные окружения

Variable name	Description	Example
HOME	Web site root directory	D:\home
WEBJOBS_DATA_PATH	Job data directory	D:\home\data\jobs\type\name.
WEBJOBS_NAME	Job name	
WEBJOBS_PATH	Temporary directory, where job is running	C:\DWASFiles\Sites\~1sitename\Temp\jobs\type\name\lpej4hrk.fks
WEBJOBS_RUN_ID	An unique ID of the job run (an UTC timestamp of the run). There's a data folder created for every run in %WEBJOBS_DATA_PATH%\%WEBJOBS_RUN_ID%	20140909070741 6329
WEBJOBS_TYPE	Job type	triggered or continuous
WEBROOT_PATH	Web site root directory	D:\home\site\wwwroot
WEBSITE_SITE_NAME	Web site name	

Пример

```
@echo off
set LOG=%WEBJOBS_DATA_PATH%\%WEBJOBS_RUN_ID%\session.log

echo WEBJOBS_PATH is %WEBJOBS_PATH%
echo Running script... >> %LOG%
date >>%LOG%
set RESULT=%ERRORLEVEL%
echo Result code is %RESULT%

rem Dumping session log to Azure log (standard output) when it exists
if exist %LOG% (
    echo Session log:
    type %LOG%
)
rem Propagating exit code to Azure
exit %RESULT%
```

Failover – main hosting on-premise, Azure- backup

Azure DNS:

***.cloudtechnologies.org CNAME cloudtechnologies-org.trafficmanager.net**

Azure Traffic Manager profile:

DNS layer: cloudtechnologies-org.trafficmanager.net Priority Routing (failover)

Endpoint list:

default (priority 1) - External (on-premise) hosting: infobox.cloudtechnologies.org (IP = 92.243.94.73)
1st backup (priority 2) – Azure Web App: cloudtechnologies-org.azurewebsites.net
2nd backup (priority 3) – Azure Linux Web App: cloudtech-org-linux-php.azurewebsites.net

Application gateway load balancing

Azure DNS:
gateway.cloudtechnologies.org A record = 52.164.244.33

Azure Application Gateway **(*Web Application Firewall* is ON)**
URL= 52.164.244.33 (68a951a5-e787-4cd5-a2a3-4f8e1ce2bfc1.cloudapp.net)

Backend pool **(appGatewayHttpListener with rule1 (no Cookie affinity, 80)):**
InfoBox hosting: infobox.cloudtechnologies.org (DNS A record IP = 92.243.94.73)