Building and Consuming API with Azure API Apps

SONU SATHYADAS

About Speaker

Sonu Sathyadas

Tech Lead & Training Consultant @ Synergetics India

6+ Years of corporate training experience

Microsoft CERTIFIED Trainer

Microsoft Specialist

Developing Microsoft Azure Solutions

Microsoft Specialist

Implementing Microsoft Azure Infrastructure Solutions

Agenda

1. App Service
2. Azure API Apps.
3. Developing/Migrating API applications
4. Deploying REST API
5. Scaling API application
6. Enabling CORS
7. Securing API
8. Azure API Management Service
9. Conclusion

Azure AppService

Build and host web applications in any programming language without managing infrastructure

- High-productivity development
  - NodeJS, PHP, .NET, Java, Python and more
  - Automated deployments from GitHub, Visual Studio Team Services, or any Git repo.
- Fully-managed platform
  - Auto-scaling and high availability
  - Supports both Windows and Linux platforms
- Enterprise-grade apps
  - High-grade security
  - AD Authentication

Azure App Service

=

WEB APPS

Web apps that scale with your business

MOBILE APPS

Build Mobile apps for any device

LOGIC APPS

Automate business process across SaaS and on-premises

API APPS

Easily build and consume APIs in the cloud

AppService Plans

App Service plans represent the collection of physical resources used to host your apps.

App Service plans define:

- Region (West US, East US, etc.)
- Scale count (one, two, three instances, etc.)
- Instance size (Small, Medium, Large)
- SKU (Free, Shared, Basic, Standard, Premium, PremiumV2, Isolated)

Azure API Apps

- A powerful platform for building and managing APIs
- Build once and consume any type of app, anywhere, at any time
- A rich ecosystem for distributing and monetizing APIs

Powerful platform

- Build on top of Azure App Service Web App
- Support all kinds of REST APIs, new or existing
- Simple yet powerful authentication support.
- Free your API up from complex authentication handling
- Expose API definition for metadata driven clients and SDK code-gen

Built-in API Apps

Connectors

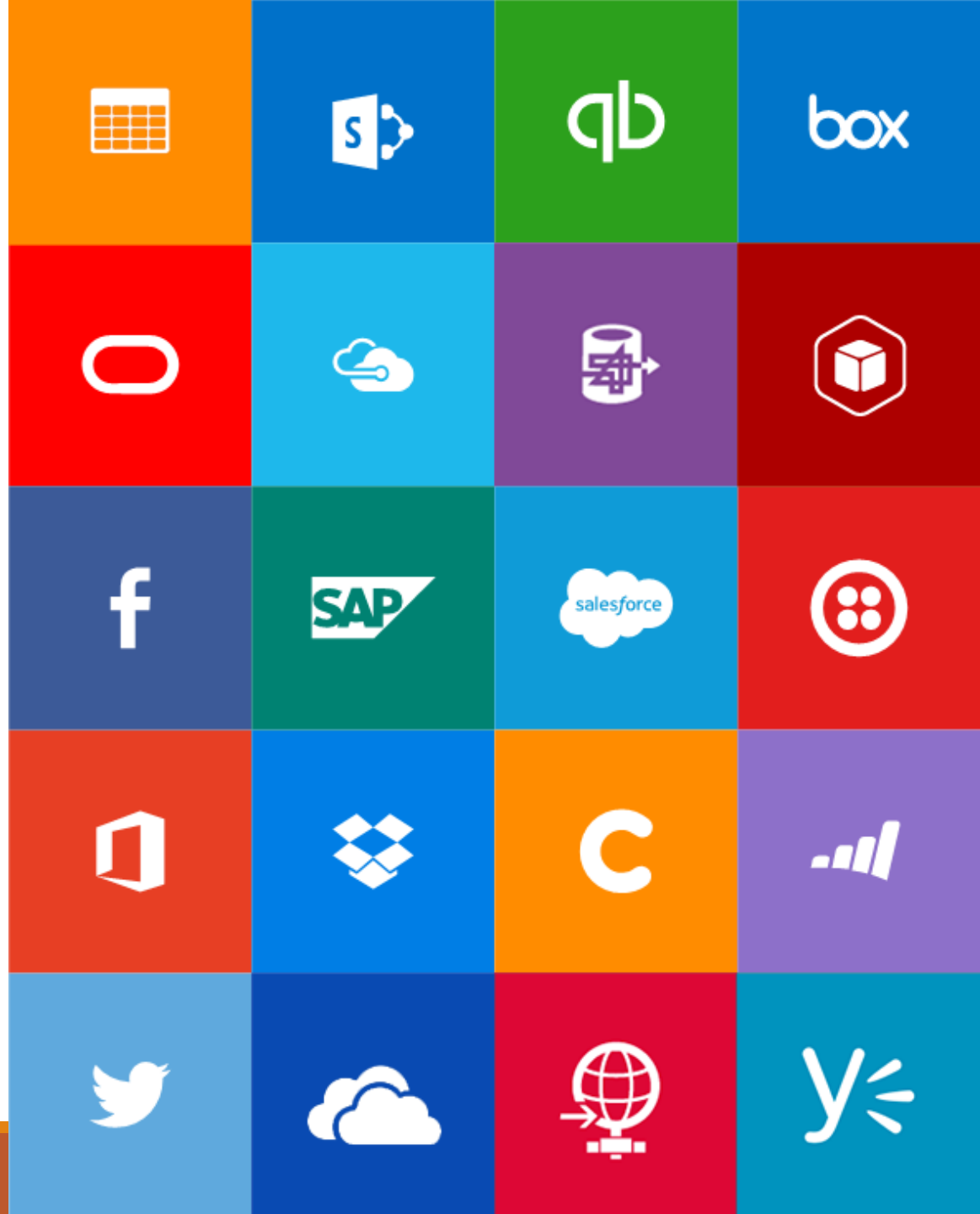
- Box
- Chatter
- Dropbox
- Azure HD Insight
- OneDrive
- SharePoint Server
- SharePoint Online
- SQL Server
- Office 365
- Oracle DB
- QuickBooks
- Salesforce
- SugarCRM
- SAP
- Azure Service Bus
- Azure Storage Blob
- Twilio
- Twitter
- IBM DB2
- Informix
- Websphere MQ
- Azure Web Jobs
- Yammer
- Facebook

Protocols

- HTTP, HTTPS
- File
- FTP
- SFTP
- POP3
- SMTP

BizTalk Services

- Flat File Encoder
- Validate
- Extract (XPath)
- Transform
- X12
- EDIFACT
- AS2
- TPM
- Rules



And more on the way…

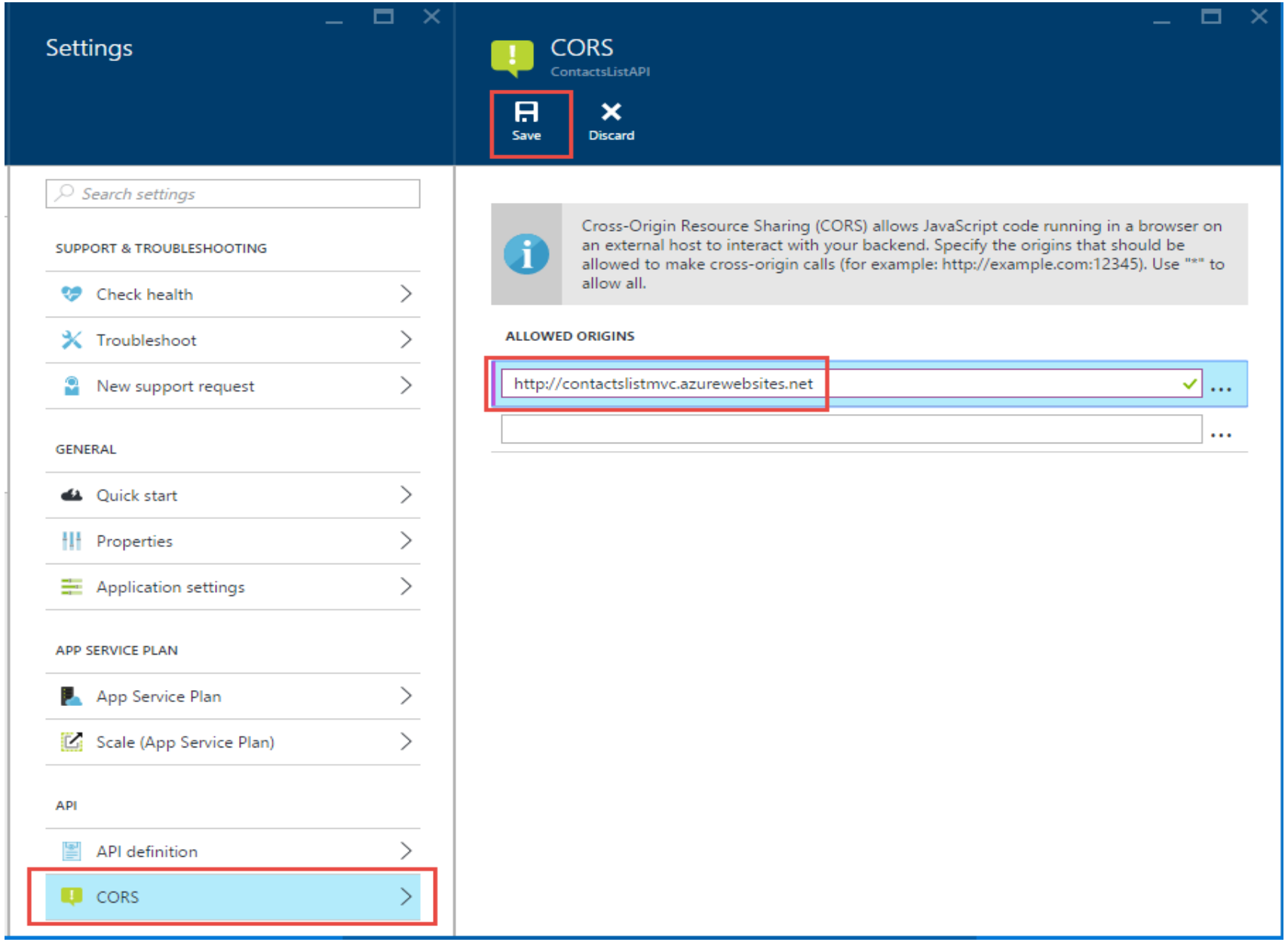
Demo

CREATING AND DEPLOYING API

Enabling CORS

CORS (Cross Origin Resource Sharing)

- Enable cross origin API calls
- Set Access-Control-Allow-Origin header
- Access-Control-Allow-Origin : *


Settings
CORS
ContactsListAPI
Save
Discard
Search settings
SUPPORT & TROUBLESHOOTING
Check health
Troubleshoot
New support request
GENERAL
Quick start
Properties
Application settings
APP SERVICE PLAN
App Service Plan
Scale (App Service Plan)
API
API definition
CORS
Cross-Origin Resource Sharing (CORS) allows JavaScript code running in a browser on an external host to interact with your backend. Specify the origins that should be allowed to make cross-origin calls (for example: http://example.com:12345). Use "*" to allow all.
ALLOWED ORIGINS
http://contactslistmvc.azurewebsites.net

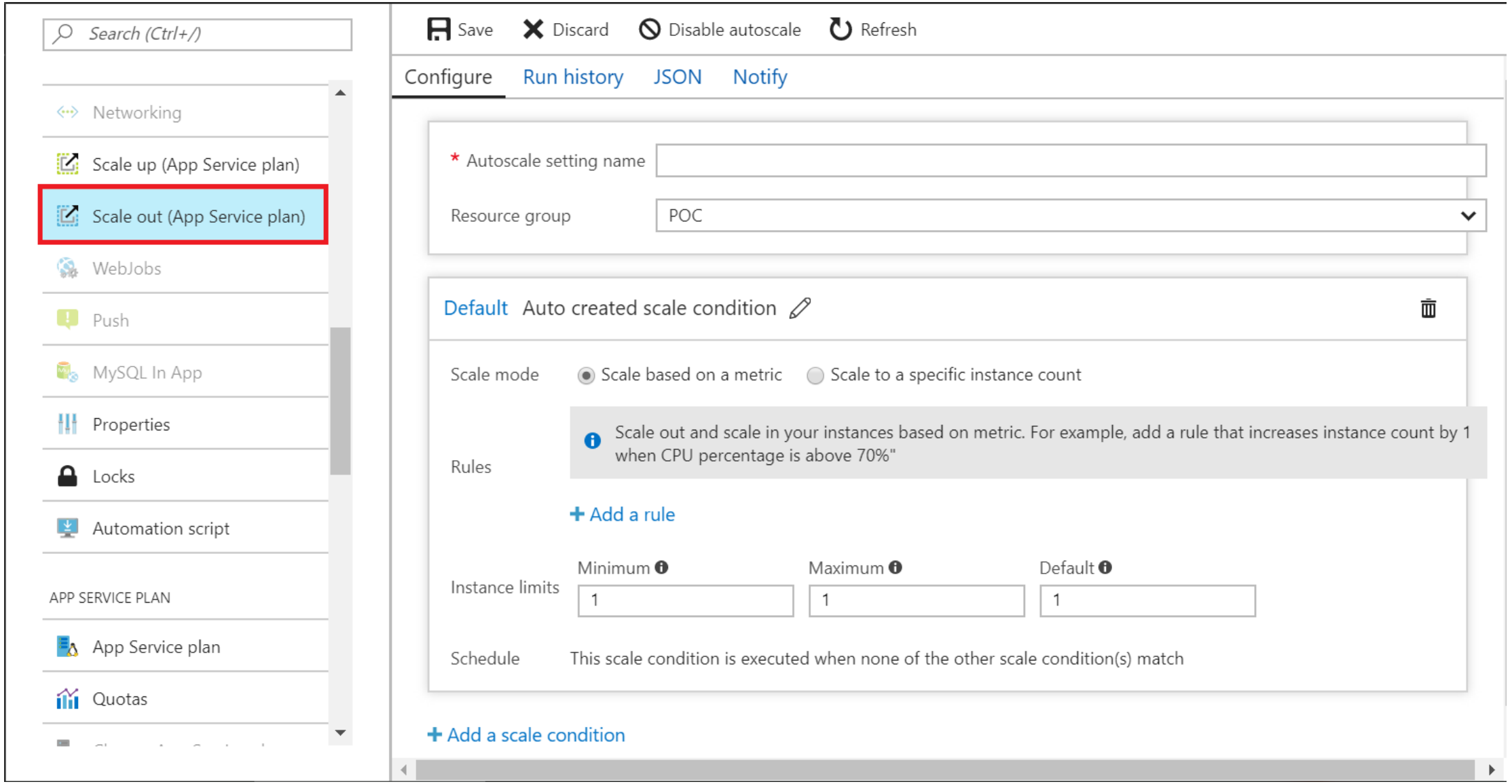
Demo

ENABLING CORS FOR API APPS

Scaling API App

Scale out/Scale up

- Horizontal Scaling and Vertical Scaling
- Vertical Scaling
  - Update the AppService Plan SKU
- Horizontal Scaling
  - Instance Count
  - Automatic scaling


Search (Ctrl+/)
Networking
Scale up (App Service plan)
Scale out (App Service plan)
WebJobs
Push
MySQL In App
Properties
Locks
Automation script
APP SERVICE PLAN
App Service plan
Quotas
Save
Discard
Disable autoscale
Refresh
Configure
Run history
JSON
Notify
* Autoscale setting name
Resource group
POC
Default
Auto created scale condition
Scale mode
Scale based on a metric
Scale to a specific instance count
Rules
Scale out and scale in your instances based on metric. For example, add a rule that increases instance count by 1 when CPU percentage is above 70%"
+ Add a rule
Instance limits
Minimum
Maximum
Default
1
1
1
Schedule
This scale condition is executed when none of the other scale condition(s) match
+ Add a scale condition

DEMO

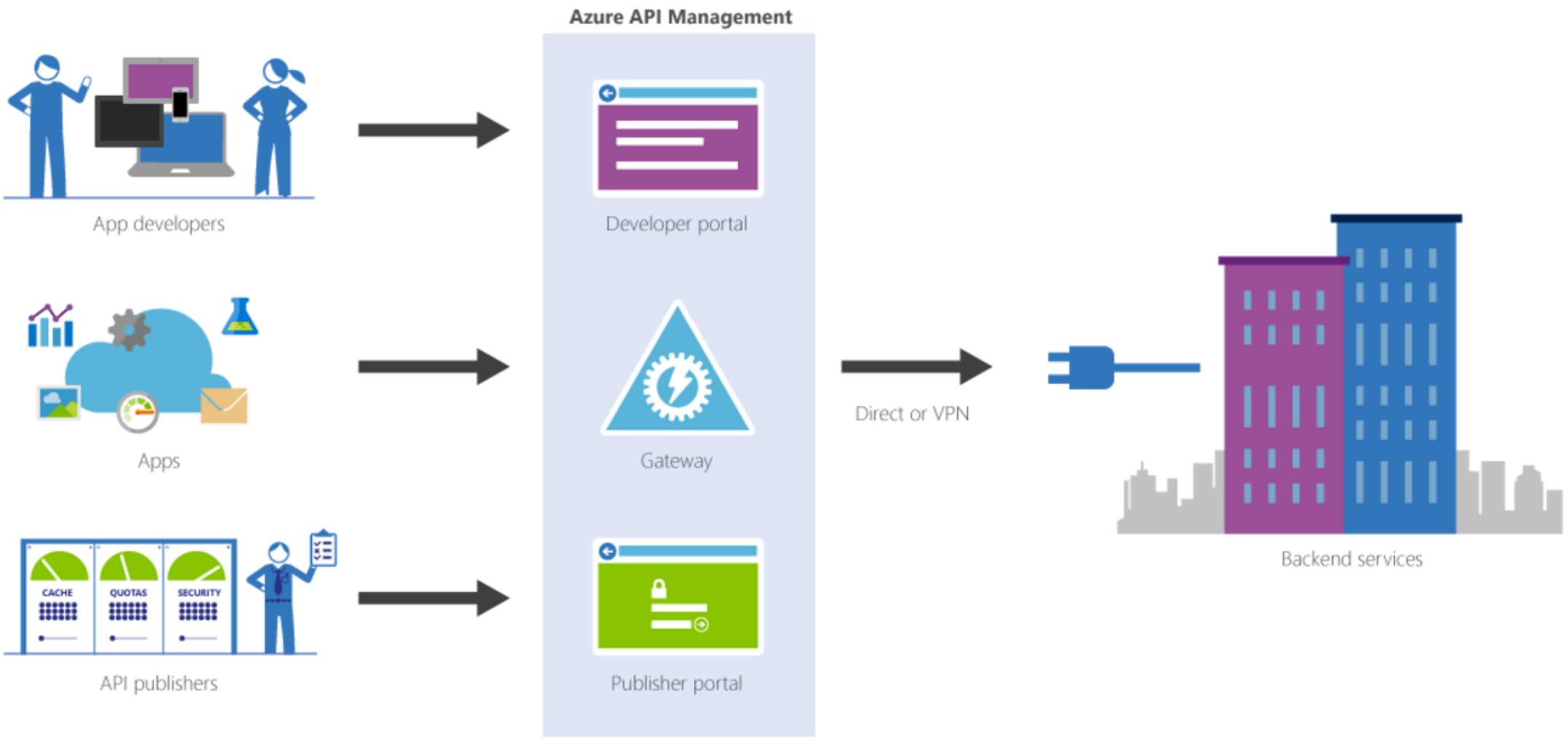
SCALING APPS

Azure API Management App

SECURING AND MANAGING API ON AZURE

Azure API Management App

- Publish APIs more securely, reliably, and at scale.
- Drive API consumption among internal teams, partners and developers
- Business and log analytics available in the admin portal
- End-to-end API management
  - Provisioning user roles
  - Creating usage plans and quotas
  - Applying policies for transforming payloads, throttling
  - Monitoring and alerts.


Azure API Management
App developers
Developer portal
Apps
Gateway
Direct or VPN
Backend services
CACHE
QUOTAS
SECURITY
API publishers
Publisher portal

Provide first-rate developer experience

Self-service API key management

Auto-generated API catalogue, documentation and code samples

OAuth-enabled API console for exploring APIs without writing a line of code

Sign in using popular Internet identity providers and Azure Active Directory

Protect and optimize your APIs

Simplify and optimize requests and responses with transformation policies

Secure APIs with key, JWT token validation and IP filtering

Protect your APIs from overload and overuse with quotas and rate limits

Use response caching for improved latency and scale

Manage all your APIs in one place

Expose all APIs behind a single static IP and domain

View near real-time usage, performance and health analytics

Automate management and integrate using REST API, PowerShell and Git

Provision API Management and scale it on demand in one or more geographical regions

API Management Components

API gateway is the endpoint that:

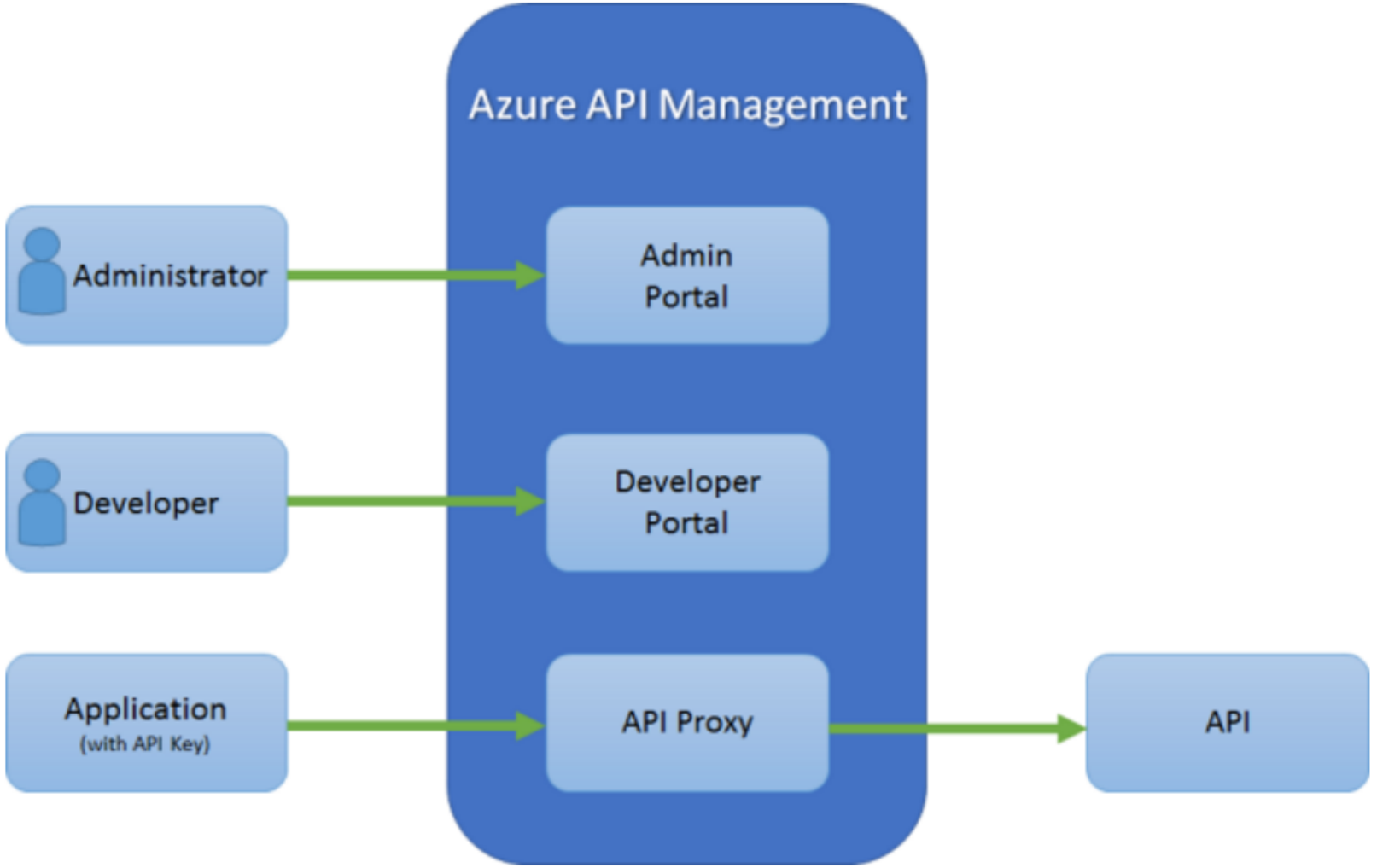
- Accepts API calls and routes them to your backends.
- Verifies API keys, JWT tokens, certificates, and other credentials.
- Enforces usage quotas and rate limits.
- Transforms your API on the fly without code modifications.
- Caches backend responses where set up.
- Logs call metadata for analytics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

Publisher portal is the administrative interface where you set up your API program. Use it to:

- 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 serves as the main web presence for developers, where they can:

- Read API documentation.
- Try out an API via the interactive console.
- Create an account and subscribe to get API keys.
- Access analytics on their own usage.


Azure API Management
Administrator
Admin Portal
Developer
Developer Portal
Application
(with API Key)
API Proxy
API

Configuring Policies

MANAGE API BY CONFIGURING POLICIES

Product or API or Operation level policies

- Allow the publisher to change the behavior of the API through configuration
- Collection of Statements that are executed sequentially on the request or response of an API.
- Applied inside the gateway which sits between the API consumer and the managed API.
- Policies can be used for
  - Format conversion from XML to JSON
  - Call rate limiting to restrict the amount of incoming calls
  - CORS
- Policies such as the Control flow and Set variable policies are based on policy expressions.

DEMO

SETTING UP POLICEIS

API Caching and Security

API Caching

- Reduce API latency, bandwidth consumption, and web service load for data that does not change frequently.
- HTTP response caching using the resource URL as the key.
- Key can be modified by request headers using the vary-by properties.

API caching

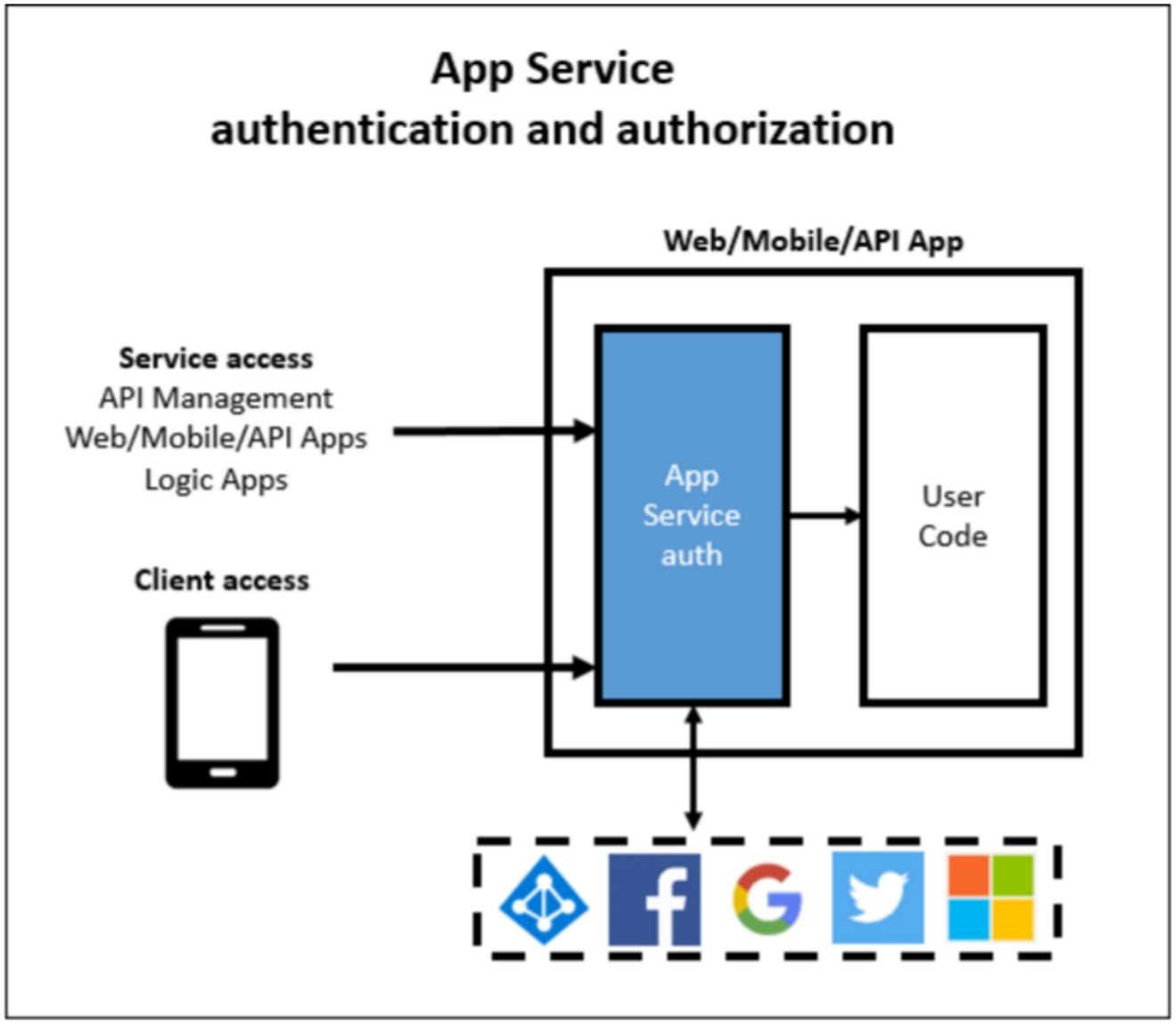
The new cache-lookup-value and cache-store-value policies provide the ability to store and retrieve arbitrary pieces of data from within policy definitions. This ability also adds value to the previously introduced send-request policy because you can now cache responses from external services.

API Management service uses a shared per-tenant data cache so that, as you scale up to multiple units you will still get access to the same cached data.

When working with a multi-region deployment there are independent caches within each of the regions. Due to this, it is important to not treat the cache as a data store, where it is the only source of some piece of information. If you did, and later decided to take advantage of the multi-region deployment, then customers with users that travel may lose access to that cached data.

Securing backend APIs

- Authentication can be enabled for accessing backend APIs
  - Active Directory Authentication
  - Facebook Authentication
  - Microsoft Authentication
  - Google Authentication
  - Twitter Authentication
- Client certificate based authentication


App Service
authentication and authorization
Web/Mobile/API App
Service access
API Management
Web/Mobile/API Apps
Logic Apps
Client access
App
Service
auth
User
Code

Thank You