



# Building an API in Azure

Josef Richberg

Sr. Data Architect

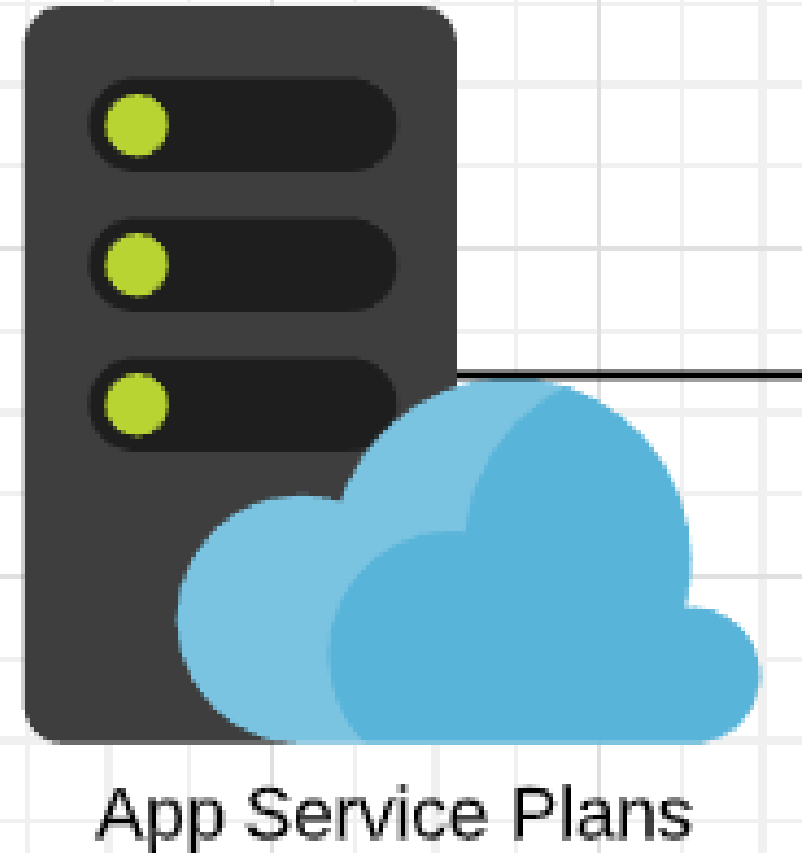
PenguinRandomHouse

<http://youtube.com/josefrichberg>

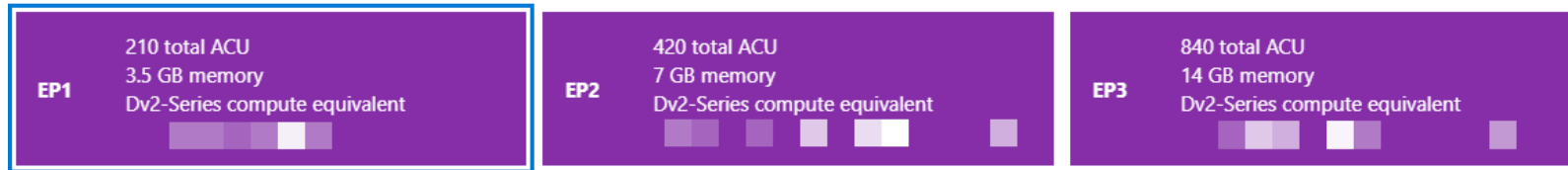
<http://josefrichberg.com>

# API Universe

- App Service Plan
  - This pertains to a static server of a specific size (can be scaled up/out). Serverless can only be scaled out



# APP Service Tiers



## Included features

Every app hosted on this App Service plan will have access to these features:



### Rapid scale

Scale out function apps based on event trigger.



### Virtual Network Integration

Connect to a virtual network.



### High Density

Efficiently share an App Service plan across multiple function apps.



### Custom domains / SSL

Configure and purchase custom domains with SNI and IP SSL bindings



### Traffic manager

## Included hardware

Every instance of your App Service plan will include the following hardware:



### Azure Compute Units (ACU)

Dedicated compute resources used to run applications



### Memory

Memory per instance available to run applications

# APP Service Plan Scale out



## Elastic Scale out

This allows you to control the bounds that your Premium plan can scale within. [Learn more](#)

### Plan Scale out

Minimum Instances ⓘ



Maximum Burst ⓘ



### App Scale out

Adjust the pre-warmed workers for each of your Apps on that App's scale out page. [Learn more](#)

# APP Service Scale out



## Elastic Scale out

This allows you to control the bounds that your Premium plan can scale within. [Learn more](#)

### Plan Scale out

Minimum Instances ⓘ  1

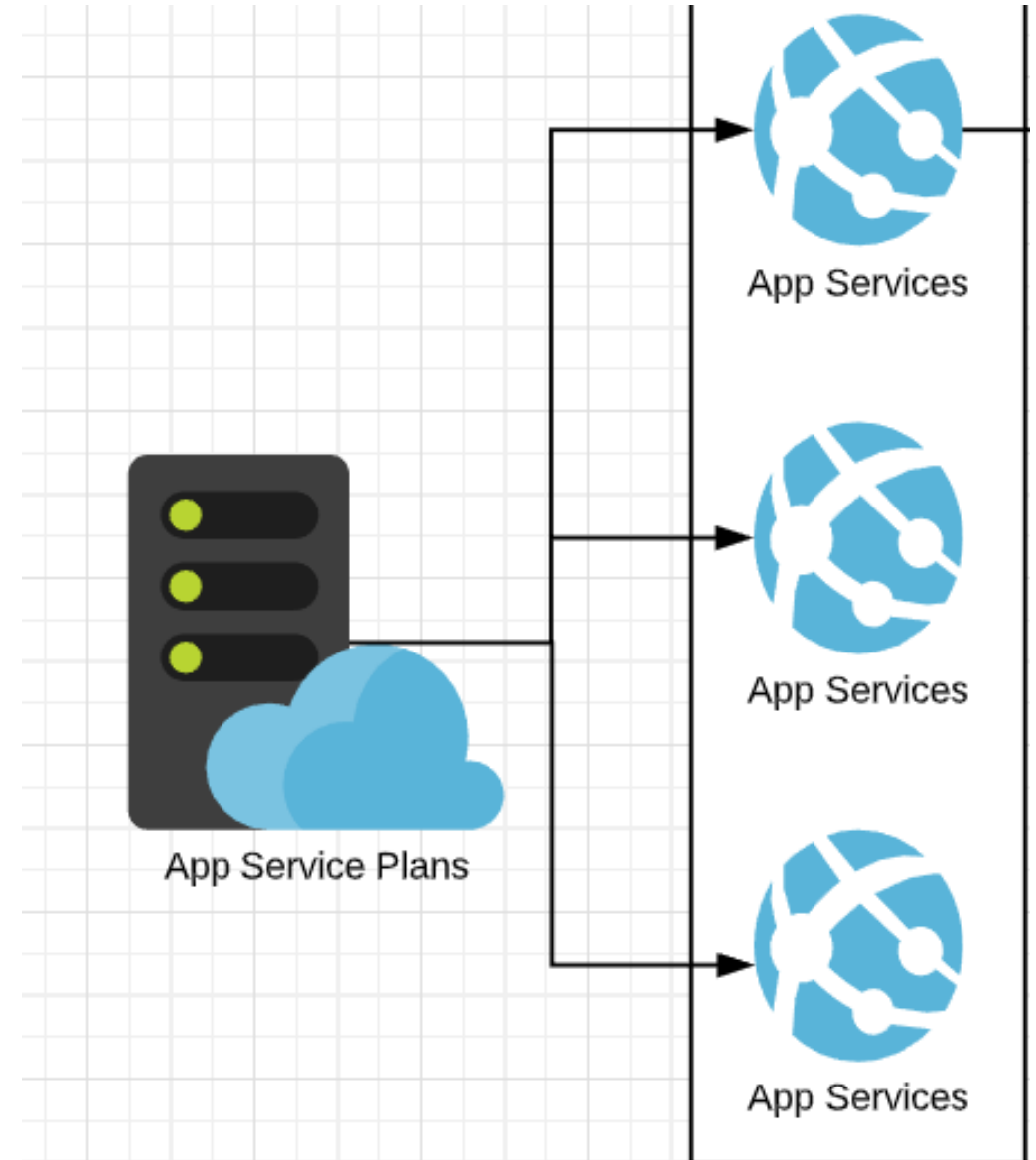
Maximum Burst ⓘ  1

### App Scale out

Pre-Warmed Instances ⓘ  1

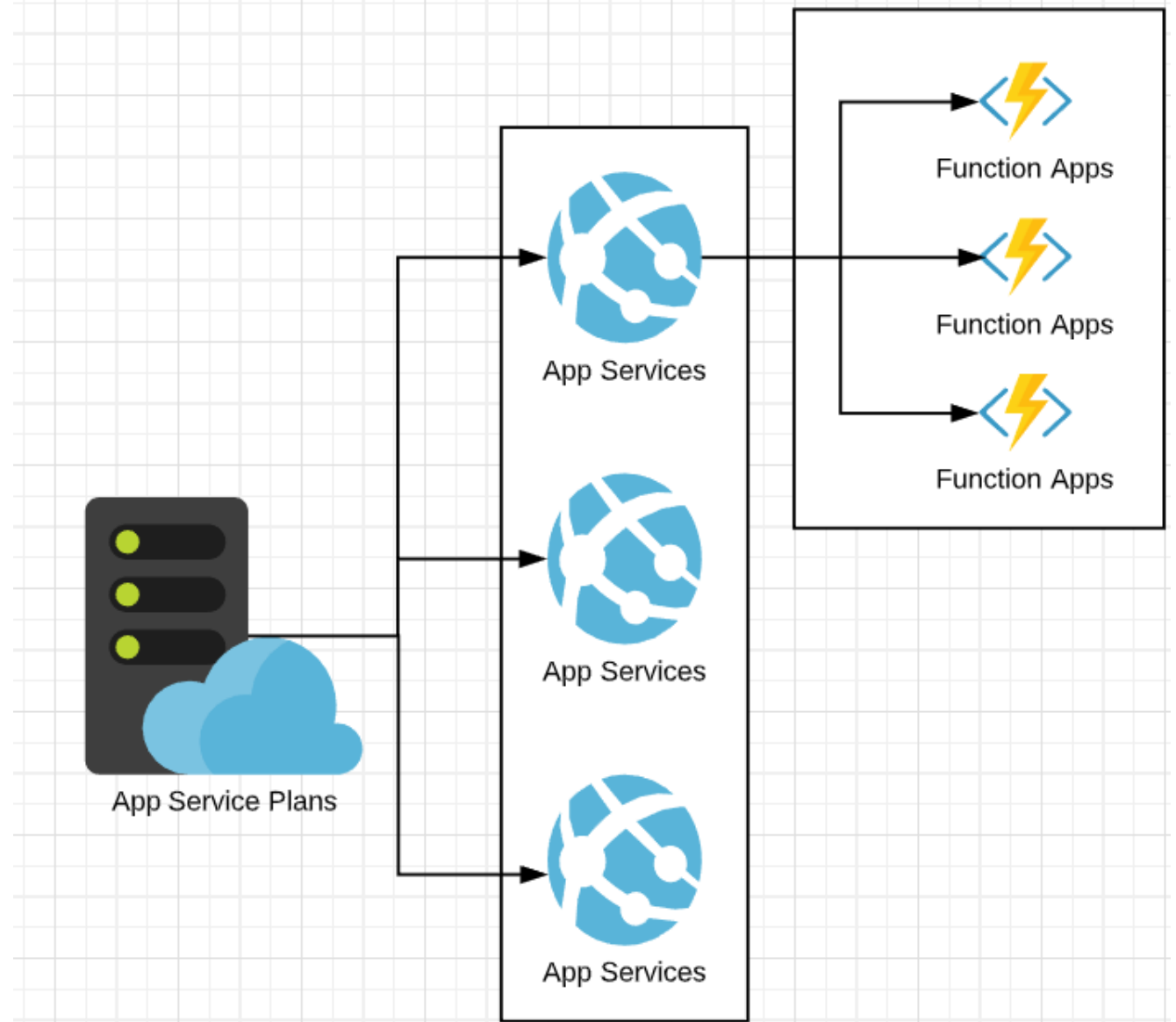
# API Universe

- App Service Plan
  - This pertains to a static server of a specific size (can be scaled up/out). Serverless can only be scaled out
  - Houses 1 or more App Services
- App Services
  - A container that houses function apps



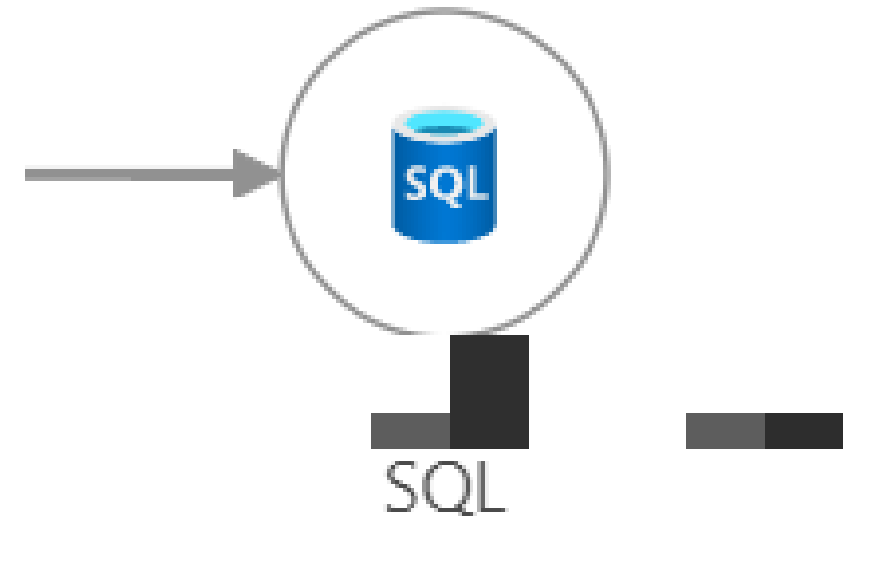
# API Universe

- App Service Plan
  - This pertains to a static server of a specific size (can be scaled up/out). Serverless can only be scaled out
  - Houses 1 or more App Services
- App Services
  - A container that houses function apps
- Function Apps
  - This is a logical grouping of functions.



# The API Train

- Database
  - Contains all logic required to provide raw data. No display manipulation (formatting, ordering, etc.)
  - All interaction is done through stored procedures with a specific limited access id.
  - BONUS– Free Cache!





```

(@UPN varchar(256),
@ListId bigint)
with execute as OWNER
as
select C001_ListName as ListName,
       C011_Description as [Classification],
       lt.C002_CatalogName as [Catalog],
       lt.C002_ListIndexName as [Index],
       lt.C002_SearchField as SearchField,
       case
         when C001_Owner=@UPN
           then cast(1 as bit)
         else cast(0 as bit)
       end isOwner,
       cast(C001_isPrivate as bit) as isPrivate,
       r.C002_RegionName as Region,
       C001_Listid as Listid,
       isnull(C001_ListComments,'') as Comments,
       isnull(C001_OutputColumns,'') as OutputColumns
from mylists.T001_Lists
join mylists.T002_ListTypes lt on (C002_ListTypeID=C001_ListTypeID
and C002_Status=1)
join mylists.T011_Classifications on (C011_ClassificationID=C001_ClassificationID)
join dbo.T002_Regions r on (r.C002_RegionId=lt.C002_RegionId)
where C001_ListId=@ListId
and (C001_Owner=@UPN or C001_isPrivate=0)

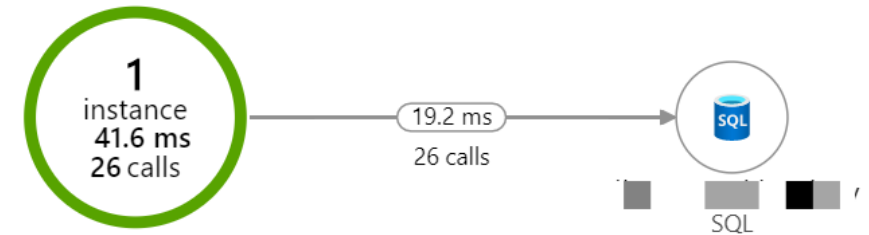
select v.C004_ListValues,
       v.C004_Sequence
from mylists.T004_ListValues v
join mylists.T001_Lists on (C001_ListId=C004_ListId
and (C001_Owner=@UPN or C001_isPrivate=0))
where v.C004_ListId=@ListId

```

# Sample Procedure

# The API Train

- Database
  - Contains all logic required to provide raw data. No display manipulation (formatting, ordering, etc.)
- Function App
  - Makes database calls on behalf of the calling program. Responsible for formatting (including possible ordering).





# Function App – C#



```
local.settings.json*  UpdateOutputColumns.cs  RetrievalIndexes.cs  Foundation.cs  PowerSearchAPI.csproj  UpdateDefaultSearch.cs  SaveList.cs
Schema: <No Schema Selected>
1  {
2    "IsEncrypted": false,
3    "Values": {
4      "AzureWebJobsStorage": "UseDevelopmentStorage=true",
5      "FUNCTIONS_WORKER_RUNTIME": "dotnet",
6      "SQL_SERVER_CON": "Persist Security Info=False;MultiUser=False;ApplicationIntent=ReadOnly;",
7      "SQL_SERVER_CON_RO": "Persist Security Info=False;MultiUser=False;ApplicationIntent=ReadOnly;",
8      "USER": "xxxxxxxxxx",
9      "SecretMessage": "xxxxxxxxxxxxxxxxxxxxxx",
10     "RetrieveList_Proc": "RetrieveList_Proc",
11     "RetrieveListDetails_Proc": "RetrieveListDetails_Proc",
12     "SaveList_Proc": "SaveList_Proc",
13     "ManageDefaultSearch_Proc": "ManageDefaultSearch_Proc",
14     "RetrieveUserDefaults_Proc": "RetrieveUserDefaults_Proc",
15     "RetrieveIndexes_Proc": "RetrieveIndexes_Proc",
16     "ManageDefaultOutput_Proc": "ManageDefaultOutput_Proc",
17   }
18 }
```

# App Service

The screenshot shows the 'Deployment slots' page in the Azure App Service portal. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events (preview), Functions, App keys, App files, Proxies, and Deployment slots (highlighted with a blue arrow). The main content area is titled 'Deployment Slots' and includes a description: 'Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the'. Below this is a table with columns: NAME, STATUS, APP SERVICE PLAN, and TRAFFIC %. The table lists two slots: 'PRODUCTION' (Running, 85% traffic) and 'v2' (Running, 15% traffic). The TRAFFIC % column is highlighted with a blue box.


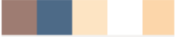

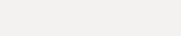

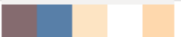



NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
PRODUCTION	Running		85
v2	Running		15

# App Services

App Service plan | Apps

Search (Ctrl+/)

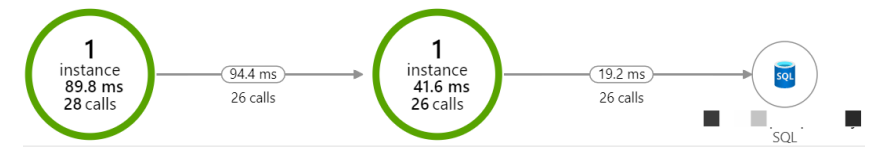
- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events (preview)
- Settings
- Apps

Name	Type	Resource Group	Status
	functionapp		Running
	functionapp		Running
 sv2	slot		Running
 	functionapp		Stopped

Two blue arrows point from the 'functionapp' entries in the table to the 'sv2' slot entry, indicating a relationship or mapping.

# The API Train

- Database
  - Contains all logic required to provide raw data. No display manipulation (formatting, ordering, etc.)
- Function App
  - Makes database calls on behalf of the calling program, including all logic, and formatting.
- API Manager
  - Proxy that manages all function apps. Provides additional capabilities including rate limiting and mocking up APIs



# Publish to API Management Service

The screenshot shows the Azure API Management console interface. The top navigation bar includes the 'App Service' logo and the title 'API Management'. Below this is a search bar and navigation links: 'Go to API Management' and 'Unlink API Management'. The left sidebar contains a 'Development Tools' section with links to 'Console', 'Advanced Tools', 'App Service Editor (Preview)', and 'Extensions'. Below this is an 'API' section with 'API Management' selected, and other options like 'API definition', 'CORS', and 'Monitoring' (with 'Alerts' sub-option). The main content area displays the 'API Management' header with a description: 'Expose your HTTP trigger Functions through Azure API Management - manage, protect, secure, and publish the APIs. [Learn more](#)'. It prompts the user to 'Select API Management instance and API for your App.' and features a dropdown menu for 'API Management \*' with a 'Create new' link below it. Further down, there's an 'API' dropdown, a checked 'Import Functions' checkbox, an unchecked 'Enable Application Insights' checkbox, and a 'Link API' button. The footer of the browser window shows 'Google Chrome' and 'Version 83.0.4103.61', and the operating system is 'Microsoft Windows 10 Enterprise 64-bit Build 6.2.9200'.

App Service | API Management

Search (Ctrl+/) « Go to API Management Unlink API Management

Change App Service plan

**Development Tools**

- Console
- Advanced Tools
- App Service Editor (Preview)
- Extensions

**API**

- API Management
- API definition
- CORS
- Monitoring
  - Alerts

**API Management**

Expose your HTTP trigger Functions through Azure API Management - manage, protect, secure, and publish the APIs. [Learn more](#)

Select API Management instance and API for your App.

API Management \*

Create new

API

☒ Import Functions ⓘ

☐ Enable Application Insights ⓘ

Link API

Google Chrome Version 83.0.4103.61

Microsoft Windows 10 Enterprise 64-bit Build 6.2.9200



# Publish to API Management -- 2

## Create from App Service

Basic | Full

\* App Service

\* Display name

\* Name

Description

API URL suffix

Base URL

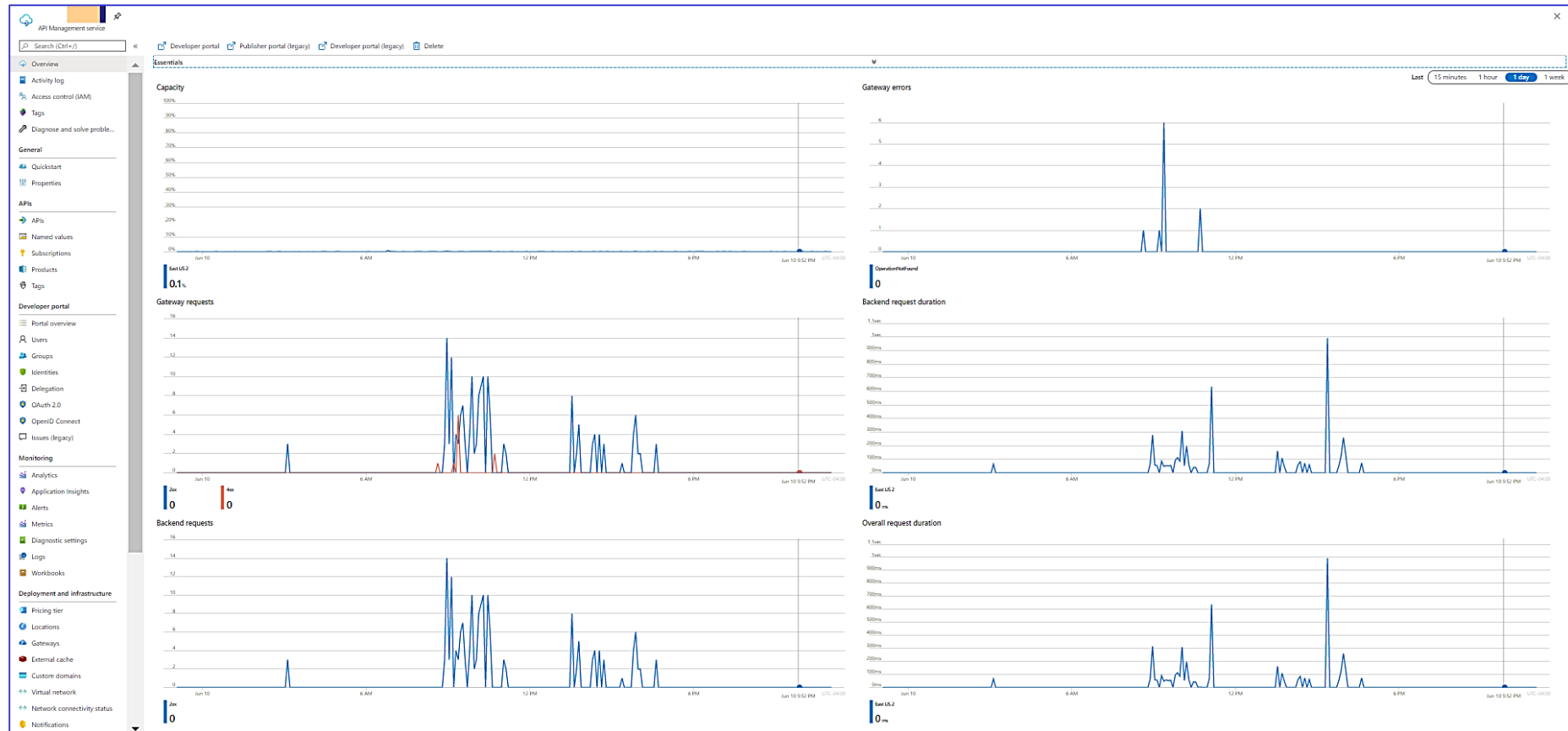
Tags

Products

**i** To publish the API, you must associate it with a product. [Learn more.](#)


Version this API? ☐

# API Management Overview



# Keys to the Kingdom

[Home](#) >

 API Management service

## Subscriptions



[Add subscription](#) [Columns](#) [Refresh](#) [?](#)

API consumers can subscribe to Products to start using your APIs. [Learn more](#)

State

All Pending approval

Scope All

Display name	Primary key	Secondary key	Scope	State	Owner	Allow tracing
	.....	.....	Product: Starter	Active	Administrator	✓
	.....	.....	Product: Unlimited	Active	Administrator	✓
Built-in all-access subscrip...	.....	.....	Service	Active		✓
	.....	.....	API: 	Active		✓
	.....	.....	API: 	Active		✓

# Keys to the Kingdom

Home > App Service | App keys

Search (Ctrl+/) << Refresh

Security

Events (preview)

Functions

Functions

App keys

App files

Proxies

Deployment

### Host keys (all functions)

Use Host keys with your clients to access all your HTTP functions in the app. \_master key grants admin access to Functions Runtime APIs.

+ New host key Show values

Filter host keys

Name	Value	
_master	Hidden value. Click to show value	Renew key value
apim-l	Hidden value. Click to show value	Renew key value

# Behold the API Manager

The screenshot displays the Azure API Management console interface. At the top, the breadcrumb navigation shows 'Home > API Management >'. The main header area includes the 'App Service' logo and the title 'API Management'. Below this is a search bar and a row of action links: 'Go to API Management', 'Download OpenAPI definition', 'Publisher portal (legacy)', 'Developer portal (legacy)', and 'Unlink API Management'.

The left sidebar contains a navigation menu with sections: 'Advanced Tools' (including 'App Service Editor (Preview)' and 'Extensions'), 'API' (including 'API Management', 'API definition', and 'CORS'), and 'Monitoring' (including 'Alerts', 'Metrics', 'Logs', 'Diagnostic settings (preview)', 'App Service logs', 'Log stream', and 'Process explorer'). A 'Support + troubleshooting' section with 'Resource health' is at the bottom.

The main content area is titled 'REVISION 1' with a timestamp 'CREATED May 29, 2020, 7:43:26 PM'. It features tabs for 'Design', 'Settings', 'Test', 'Revisions', and 'Change log'. The 'Design' tab is active, showing a visual flow diagram. On the left, a 'Search operations' box and a 'Filter by tags' dropdown are present, along with a 'Group by tag' checkbox and an '+ Add operation' button. Below these is a list of 'All operations' with columns for HTTP method (all 'POST'), status, and a menu icon.

The flow diagram consists of three main panels: 'Frontend', 'Inbound processing', and 'Backend'. The 'Frontend' panel is empty. The 'Inbound processing' panel contains a description 'Modify the request before it is sent to the backend service.', a 'Policies' section with a 'base' policy box and an '+ Add policy' button, and an 'Outbound processing' section with a description 'Modify the response before it is sent to the client.', a 'Policies' section with a 'base' policy box, and an '+ Add policy' button. The 'Backend' panel contains an 'HTTP(s) endpoint' field with a 'base' policy box and an '+ Add policy' button. Arrows indicate the flow from Frontend to Inbound processing, then to Backend, and finally to Outbound processing.

# Behold the API Manager

The screenshot displays the Azure API Management console interface. At the top, the breadcrumb navigation shows 'Home > API Management >'. The main header area includes the 'App Service' logo and the title 'API Management'. Below this, a search bar and a row of action links are visible: 'Go to API Management', 'Download OpenAPI definition', 'Publisher portal (legacy)', 'Developer portal (legacy)', and 'Unlink API Management'. The left-hand navigation pane is organized into sections: 'Advanced Tools' (containing 'App Service Editor (Preview)' and 'Extensions'), 'API' (containing 'API Management', 'API definition', and 'CORS'), 'Monitoring' (containing 'Alerts', 'Metrics', 'Logs', 'Diagnostic settings (preview)', 'App Service logs', 'Log stream', and 'Process explorer'), and 'Support + troubleshooting'. The main content area is titled 'REVISION 1' with a creation timestamp of 'May 29, 2020, 7:43:26 PM'. It features a tabbed interface with 'Design', 'Settings', 'Test', 'Revisions', and 'Change log'. The 'Settings' tab is active, showing a list of operations on the left and a grid of policy cards on the right. The policy cards include: 'Filter IP addresses' (ip-filter), 'Limit call rate' (rate-limit-by-key), 'Mock responses' (mock-response), 'Set query parameters' (set-query-parameter), 'Set headers' (set-header), 'Allow cross-origin resource sharing (CORS)' (cors), 'Cache responses' (cache-lookup/store), and 'Set usage quota by key' (quota-by-key). Each card provides a brief description and a 'Learn more' link. The 'Mock responses' card is highlighted with a blue border. At the bottom, there are links for 'Validate JWT' and 'Other policies'.

# Behold the API Manager

The screenshot displays the Azure API Management console interface. On the left, a navigation pane lists various services and tools, including Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The 'APIs' section is expanded, showing a list of APIs. The main content area is titled 'APIs' and shows a list of APIs. The selected API is highlighted, and its details are shown on the right. The 'Diagnostics Logs' section is active, showing configuration options for logging. The 'Enable' checkbox is checked. The 'Destination' is set to 'Application Insights'. The 'Sampling (%)' is set to 50. The 'Always log errors' checkbox is checked. The 'Verbosity' is set to 'Information'. The 'Additional settings' section shows 'Headers to log' as 'Accept-Language' and 'Number of payload bytes to log (up to 8192)' as 0. The 'Advanced Options' link is visible at the bottom.

Home > [User] | APIs  
API Management service

Search (Ctrl+/) << Publisher portal (legacy) Developer portal (legacy) Developer portal

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems

General  
Quickstart  
Properties

APIs  
APIs  
Named values  
Subscriptions  
Products  
Tags

Developer portal  
Portal overview  
Users  
Groups  
Identities  
Delegation  
OAuth 2.0

Search APIs  
Filter by tags  
Group by tag  
Add API

All APIs

REVISION 1 CREATED May 29, 2020, 7:43:26 PM

Design Settings Test Revisions Change log

User authorization ☒ None ☐ OAuth 2.0 ☐ OpenID connect

Diagnostics Logs

Application Insights Azure Monitor

Enable ☒

Destination  Manage

Sampling (%)  50

For high traffic APIs, please read this [documentation](#) to understand performance implications and log sampling.

Always log errors ☒

Verbosity

Additional settings

Advanced Options →

# The API Train

- Database
  - Contains all logic required to provide raw data. No display manipulation (formatting, ordering, etc.)
- Function App
  - Makes database calls on behalf of the calling program, including all logic, and formatting.
- API Manager
  - Proxy that manages all function apps. Provides additional capabilities including rate limiting and mocking up APIs
- API Client
  - The calling application/resource

