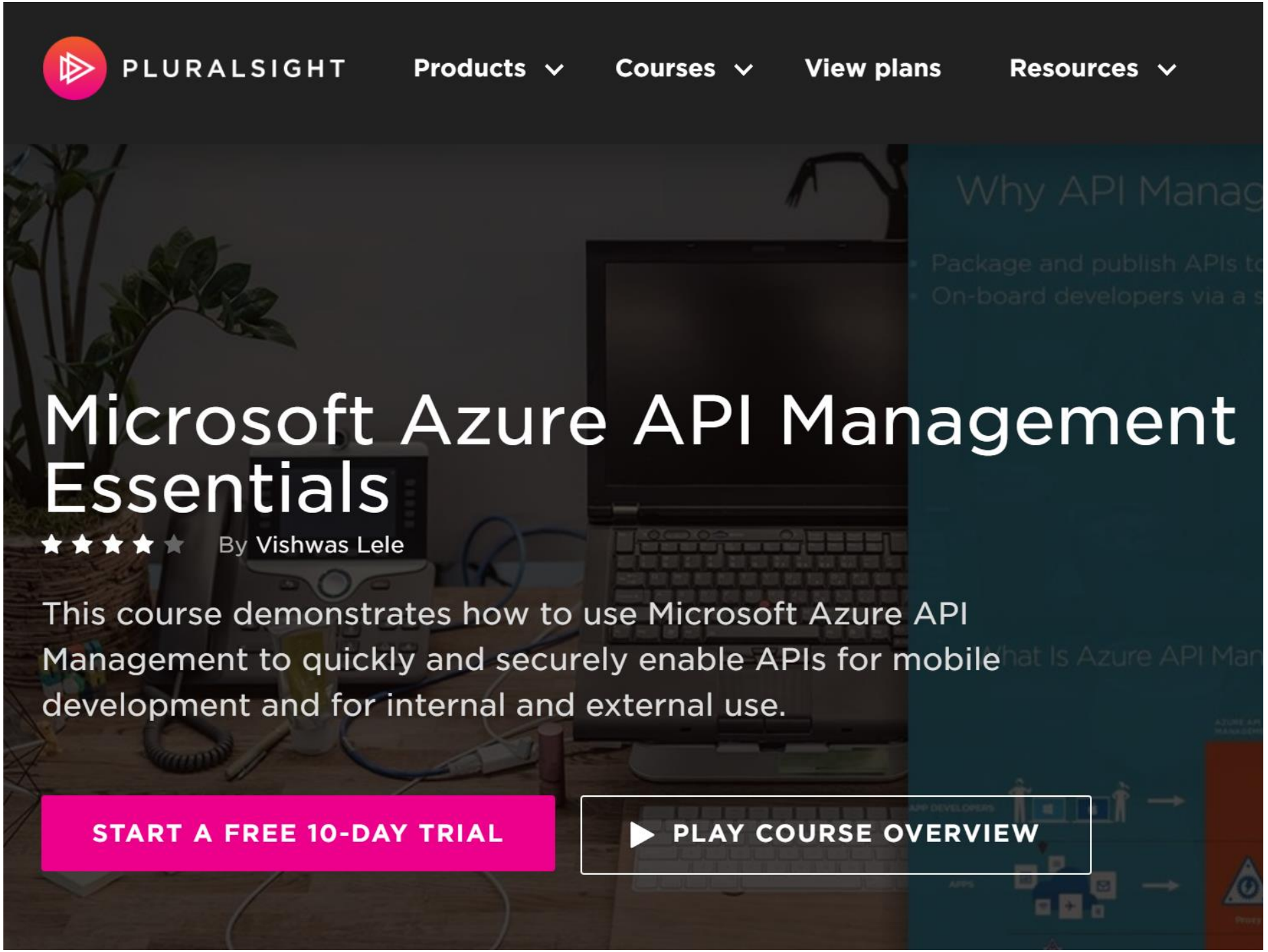
Azure API Management


PLURALSIGHT
Products
Courses
View plans
Resources
Why API Manag
Package and publish APIs to
On-board developers via a s
Microsoft Azure API Management Essentials
By Vishwas Lele
This course demonstrates how to use Microsoft Azure API Management to quickly and securely enable APIs for mobile development and for internal and external use.
START A FREE 10-DAY TRIAL
PLAY COURSE OVERVIEW

API Management

The hub for enterprise APIs

Discover
Learn
On-board
Try
Get support
SDKs and samples

1st and 3rd party apps

Abstract
Secure & protect
Evolve
Monitor
Analyze
Productize
Monetize

App developers

Consume

Developer portal

Mediate

Gateway

Publish

Azure portal

API managers and developers

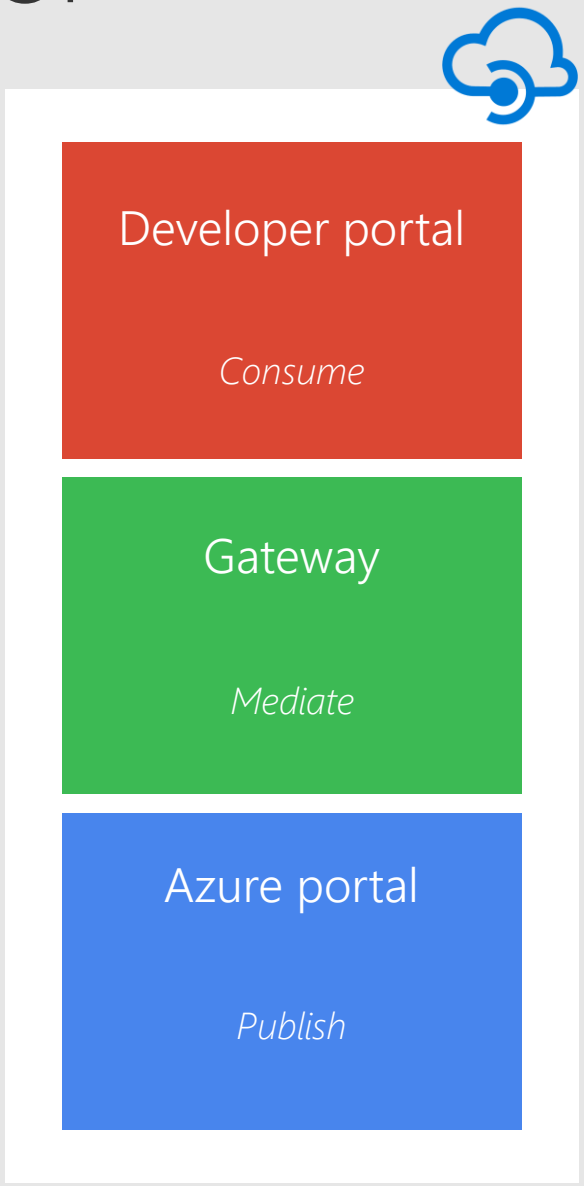
APIs on Azure, on premises

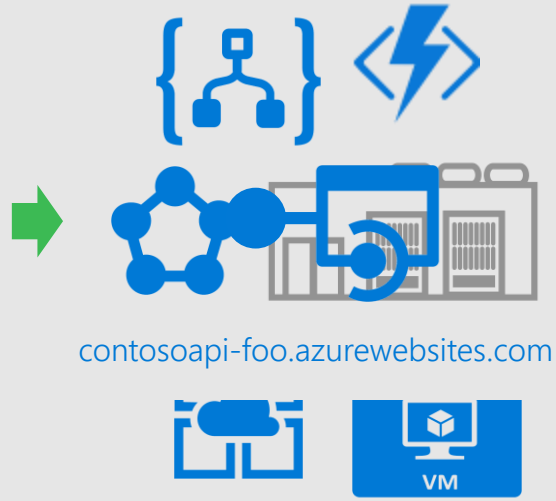
Façade and front door

App developers

1st and 3rd party apps

API managers and developers





contosoapi-foo.azurewebsites.com

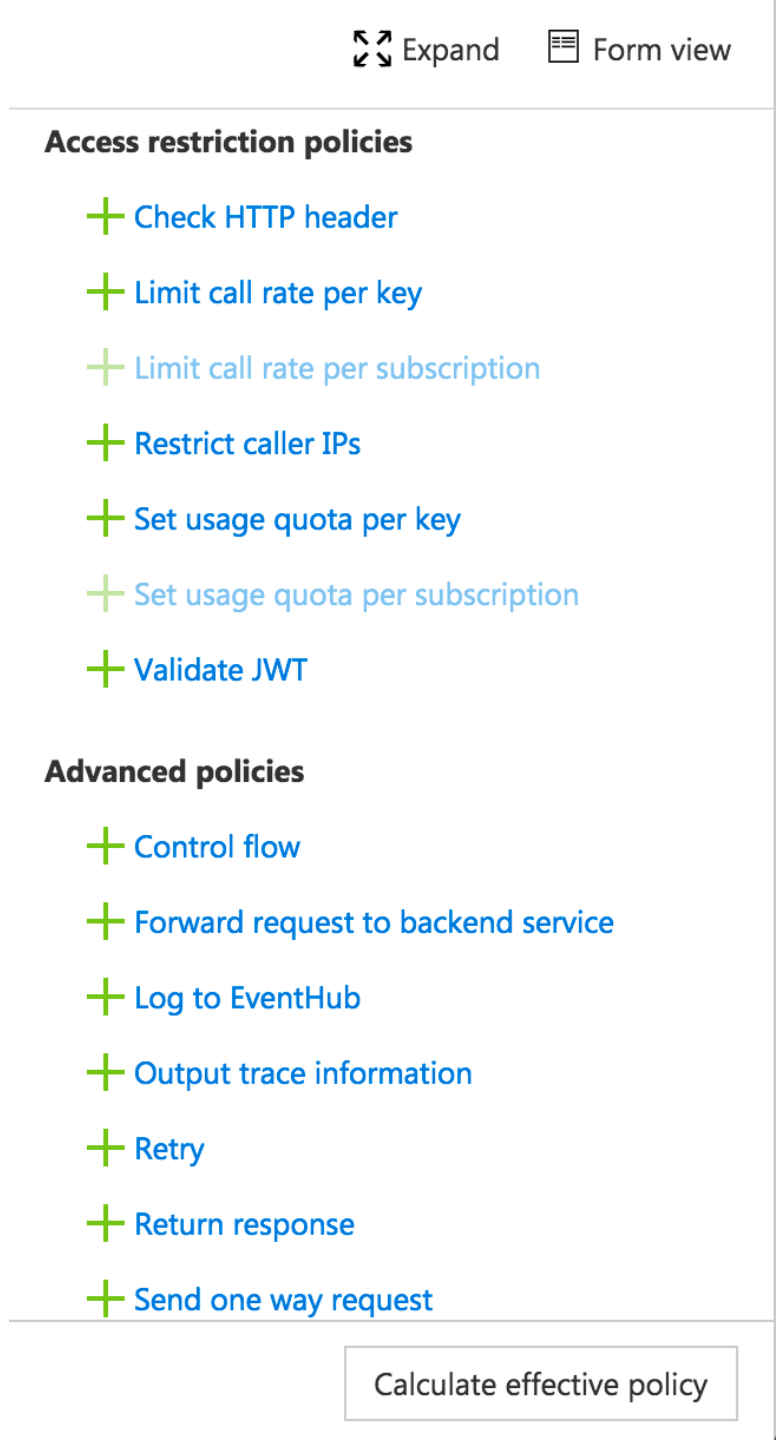
APIs on Azure and outside

What are policies?

- Pre/Post processing and validation of incoming requests
- Can be defined at multiple levels:
  - Operation
  - API
  - Product
- Each policy is defined as an XML document
  - Dynamic expressions are wrapped in @()
- Defined in Publisher Portal (or Azure Portal)

Policies

- Encapsulate common API management functions
  - Access control, Protection, Transformation, Caching, …
- Chained together into a pipeline
- Mutate request context or change API behavior
- Set in the inbound and outbound directions
- Can be triggered on error
- Applied at a variety of scopes



Policies - Example

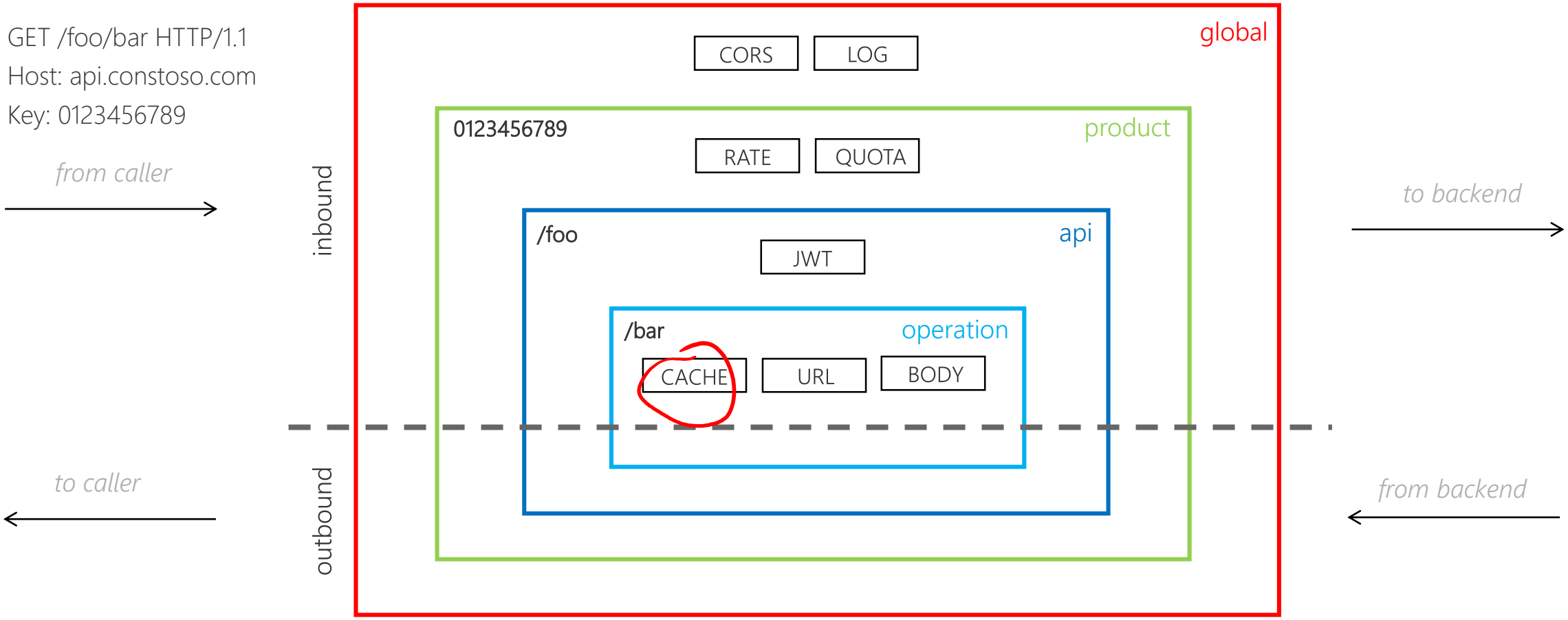
- Limit to 5 calls / minute
- Maximum 100 calls / week

```
<policies>
    <inbound>
        <rate-limit renewal-period="60" calls="5" />
        <quota renewal-period="604800" calls="100" />
        <base />
    </inbound>
    <backend>
        <base />
    </backend>
    <outbound>
        <base />
    </outbound>
    <on-error>
        <base />
    </on-error>
</policies>
```

Policy expressions

```
<policies>
    <inbound>
        <base/>
        <set-variable name="bodySize" value="@(context.Request.Headers["Content-Length"][0])"/>
        <choose>
            <when condition="@(int.Parse(context.Variables.GetValueOrDefault<string>("bodySize")) < 256*1024)">
            </when>
            <otherwise>
                <rewrite-uri template="{{alternate-path-and-query}}"/>
                <set-backend-service base-url="{{alternate-host}}"/>
            </otherwise>
        </choose>
    </inbound>
    <outbound>
        <base/>
    </outbound>
</policies>
```

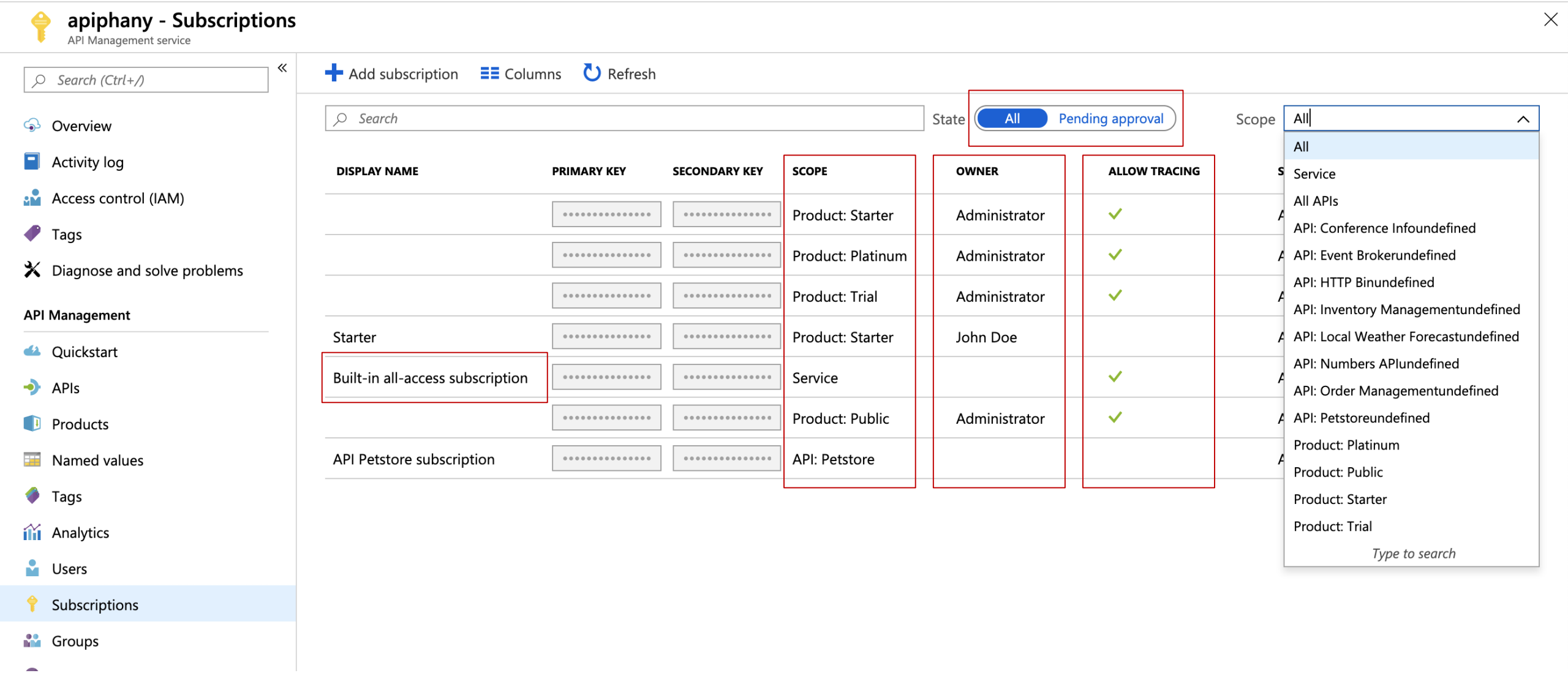
Policy scopes



<<<Demo Steps>>>

- Create an API
- Test using the portal
- Subscribe
- API Catalog
- Dev Portal

Subscriptions



Versioning is a highly debated subject

Version or not?

Semantic versioning?

What is a breaking change?

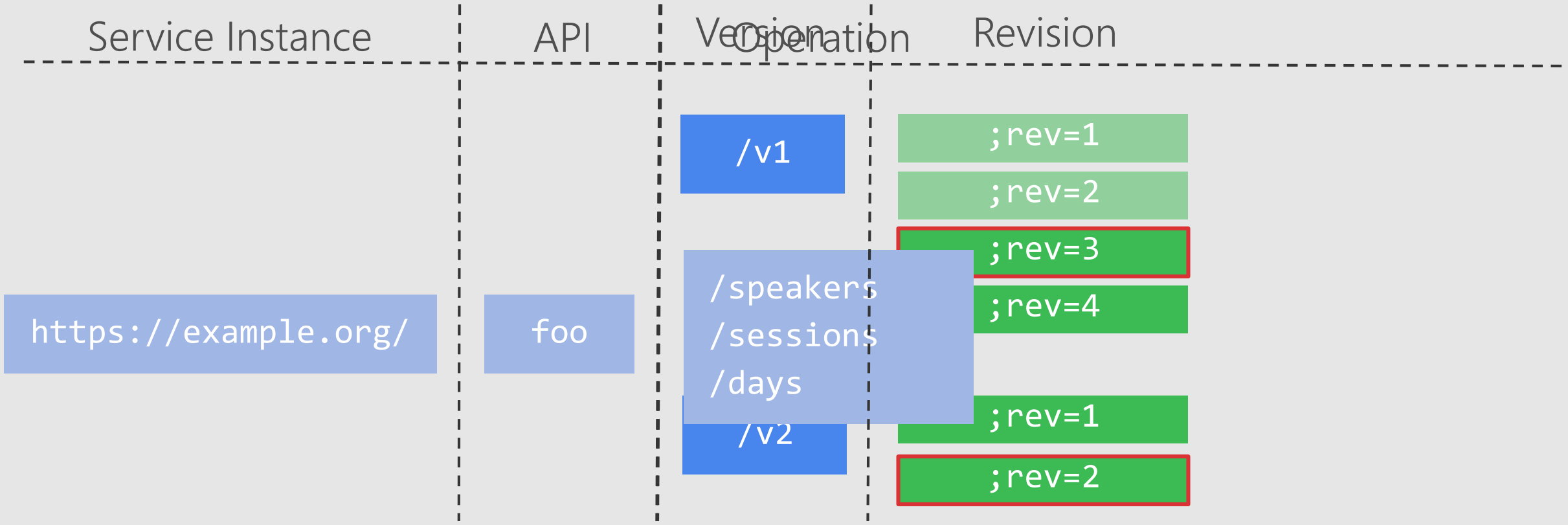
Where to place version information?

- Path? Query? Header? Media type?

What format to use?

- Number? Date? Name?

Versions and revisions in API Management



There is no one-size-fit-all solution

Teams

Team structure

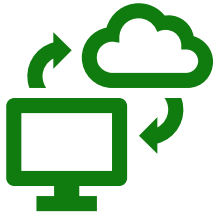
Organizational culture

Governance model

Tools

Azure Pipelines, Jenkins

Azure Repo, GitHub



Deployment Options

Portal

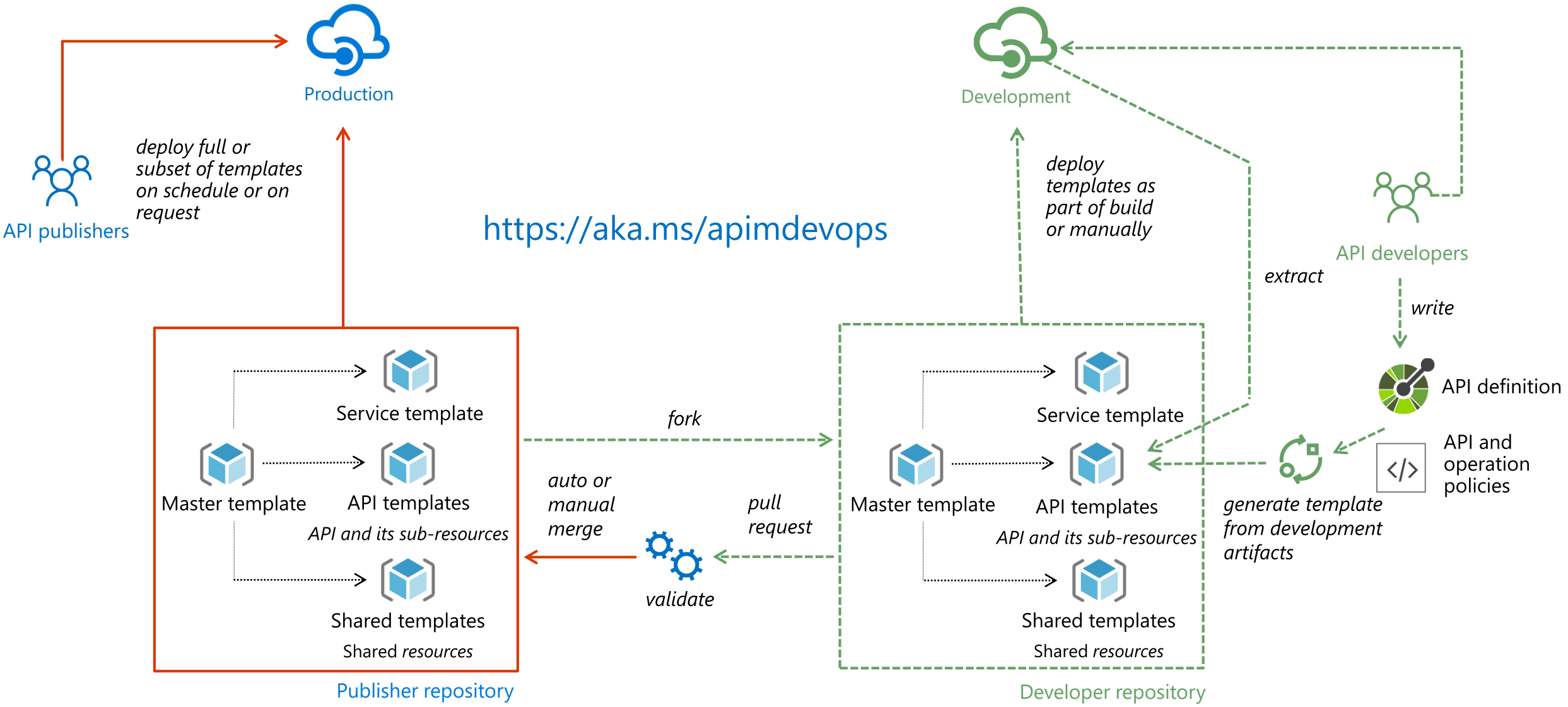
Management API

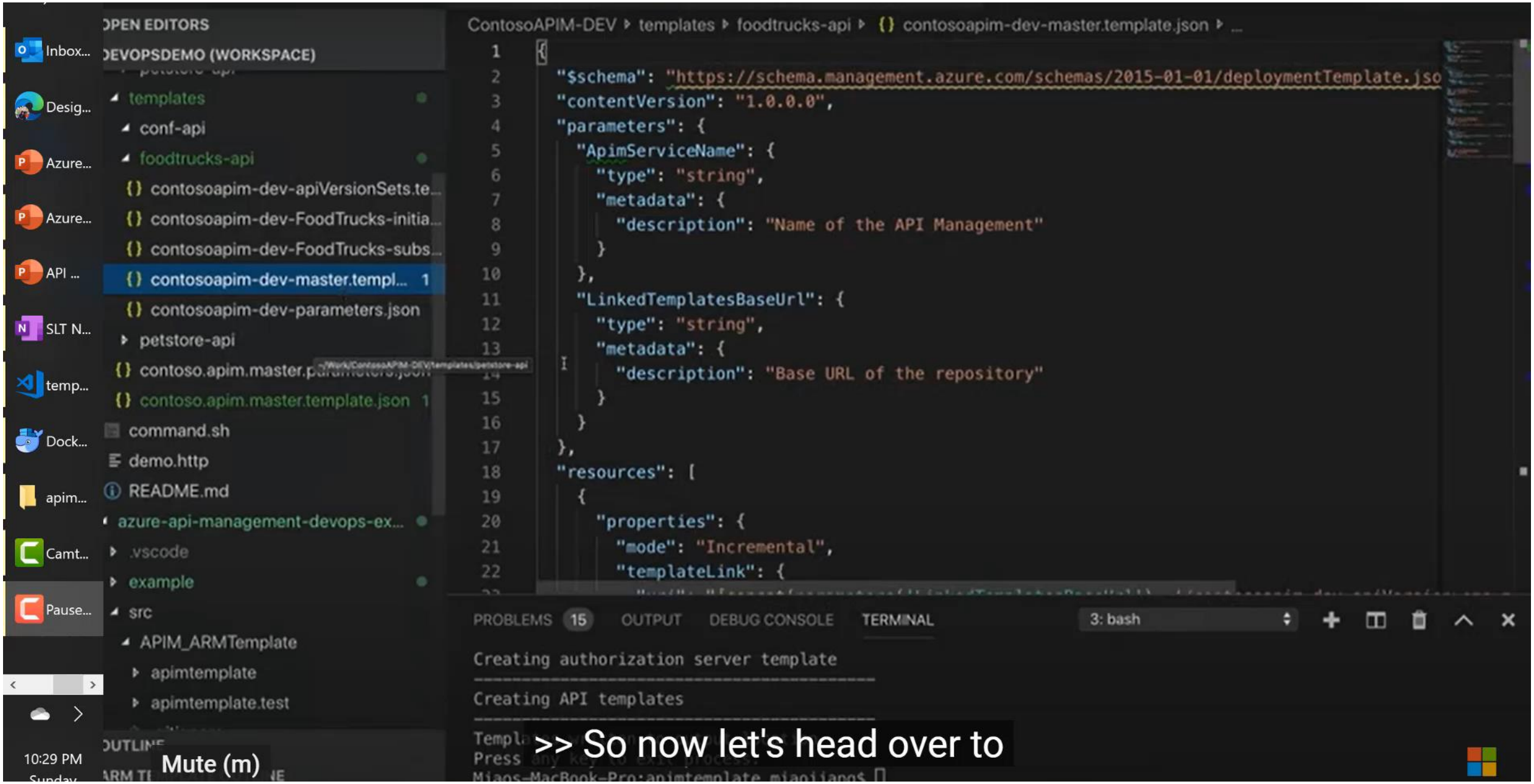
PowerShell

SDKs

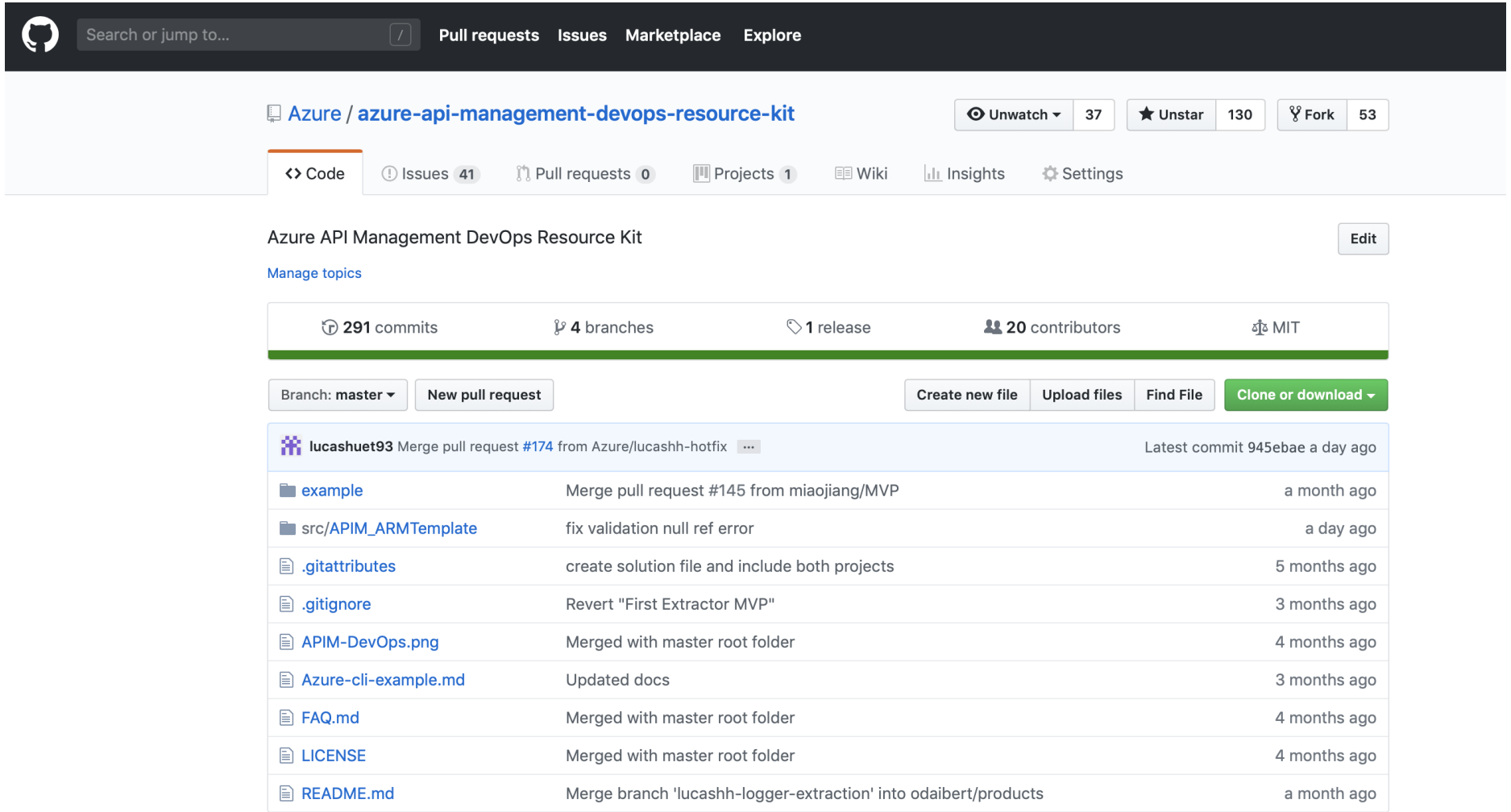
Classic Git deployment

Resource Manager Templates


Production
Development
API publishers
deploy full or subset of templates on schedule or on request
https://aka.ms/apimdevops
deploy templates as part of build or manually
API developers
extract
write
API definition
API and operation policies
generate template from development artifacts
Service template
Master template
API templates
API and its sub-resources
Shared templates
Shared resources
fork
auto or manual merge
pull request
validate
Publisher repository
Developer repository


Inbox...
Desig...
Azure...
Azure...
API ...
SLT N...
temp...
Dock...
apim...
Camt...
Pause...
10:29 PM
Sunday
Mute (m)
OPEN EDITORS
DEVOPSDEMO (WORKSPACE)
templates
conf-api
foodtrucks-api
contosoapim-dev-apiVersionSets.te...
contosoapim-dev-FoodTrucks-initia...
contosoapim-dev-FoodTrucks-subs...
contosoapim-dev-master.templ... 1
contosoapim-dev-parameters.json
petstore-api
contoso.apim.master.p...
contoso.apim.master.template.json 1
command.sh
demo.http
README.md
azure-api-management-devops-ex...
.vscode
example
src
APIM_ARMTemplate
apimtemplate
apimtemplate.test
OUTLINE
ARM TE...
ContosoAPIM-DEV ▸ templates ▸ foodtrucks-api ▸ {} contosoapim-dev-master.template.json ▸ ...
{
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.jso
"contentVersion": "1.0.0.0",
"parameters": {
"ApimServiceName": {
"type": "string",
"metadata": {
"description": "Name of the API Management"
}
},
"LinkedTemplatesBaseUrl": {
"type": "string",
"metadata": {
"description": "Base URL of the repository"
}
}
},
"resources": [
{
"properties": {
"mode": "Incremental",
"templateLink": {
PROBLEMS 15
OUTPUT
DEBUG CONSOLE
TERMINAL
3: bash
Creating authorization server template
Creating API templates
Press any key to exit process...
Miaos-MacBook-Pro:apimtemplate miaojiapo$
>> So now let's head over to

DevOps Resource Kit



https://aka.ms/apimdevops

<<<Demo Steps>>>

- XML to REST
- Routing
- Transaction Management
- SOAP to REST

Routing

```
<choose>
    <when condition='@(context.Request.Url.Query.GetValueOrDefault("version") == "2020-03")'>
        <set-backend-service base-url="{{usual-host}}" />
    </when>
    <otherwise>
        <set-backend-service base-url="{{alternate-host}}"/>
    </otherwise>
</choose>
```

```
<choose>
    <when condition="{{broadcast-is-true}}">
      <send-one-way-request mode="new">
        <set-url>https://broadcast.message.com/services</set-url>
        <set-method>POST</set-method>
        <set-body>@{
                return new JObject(
                        new JProperty("messageId","bJtrpFi1fO1JMCcwLx8uZyAg"),
                        new JProperty("message", "message details"))
                        ).ToString();
            }</set-body>
      </send-one-way-request>
    </when>
</choose>
```

Encryption

```
<choose>
    <when condition="@(context.Request.Certificate == null || !context.Request.Certificate.Verify()
                    || context.Request.Certificate.Issuer != "trusted-issuer"
                    || context.Request.Certificate.SubjectName.Name != "expected-subject-name")" >
        <return-response>
            <set-status code="403" reason="Invalid client certificate" />
        </return-response>
    </when>
</choose>
```

Importing a SOAP API

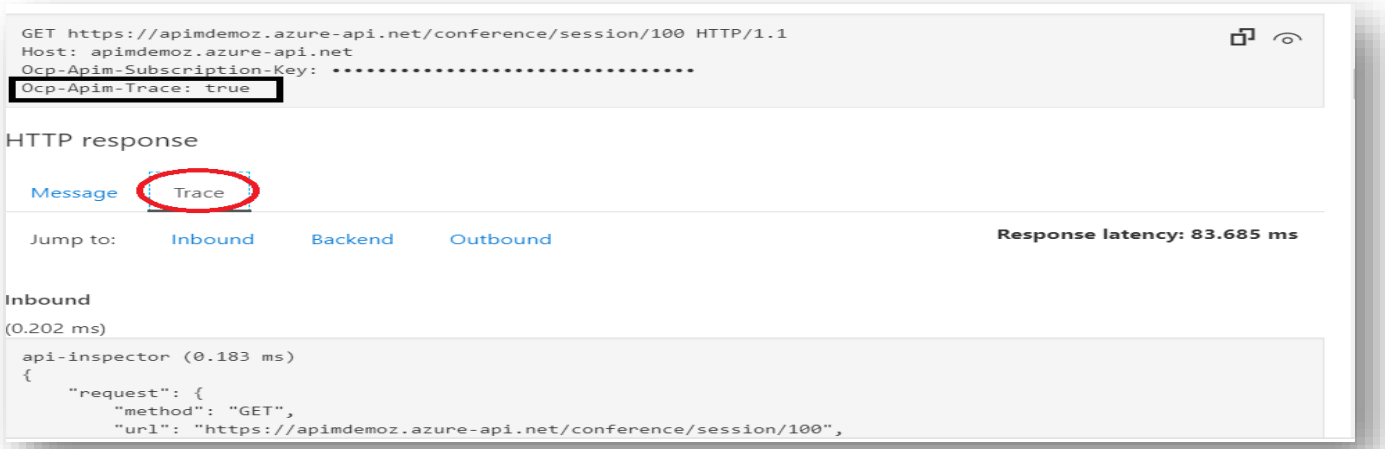
SOAP passthrough

- Gateway receives SOAP
- SOAP sent to service

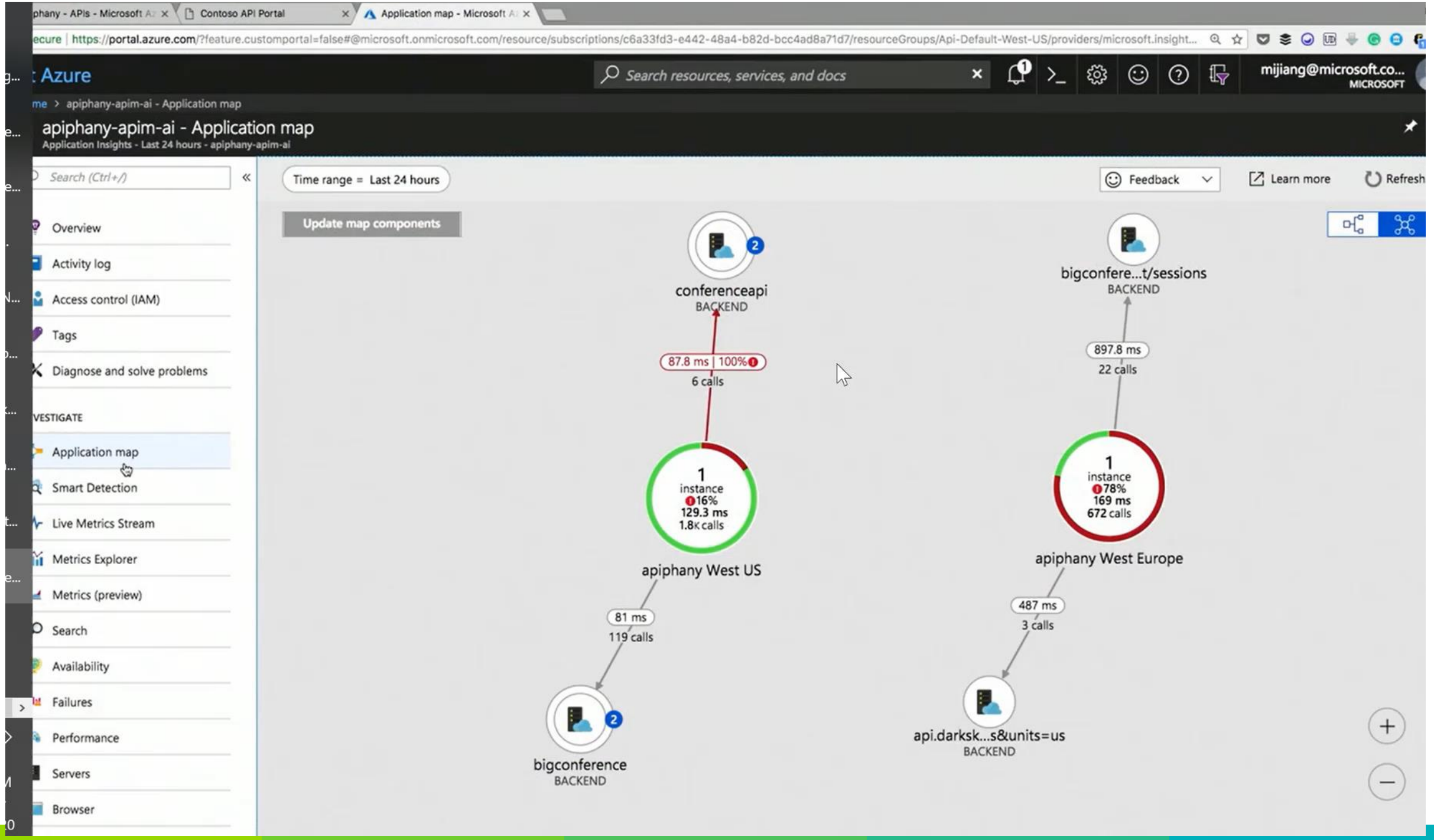
SOAP to REST

- Gateway receives REST
- SOAP sent to service

Tracing and Error Handling

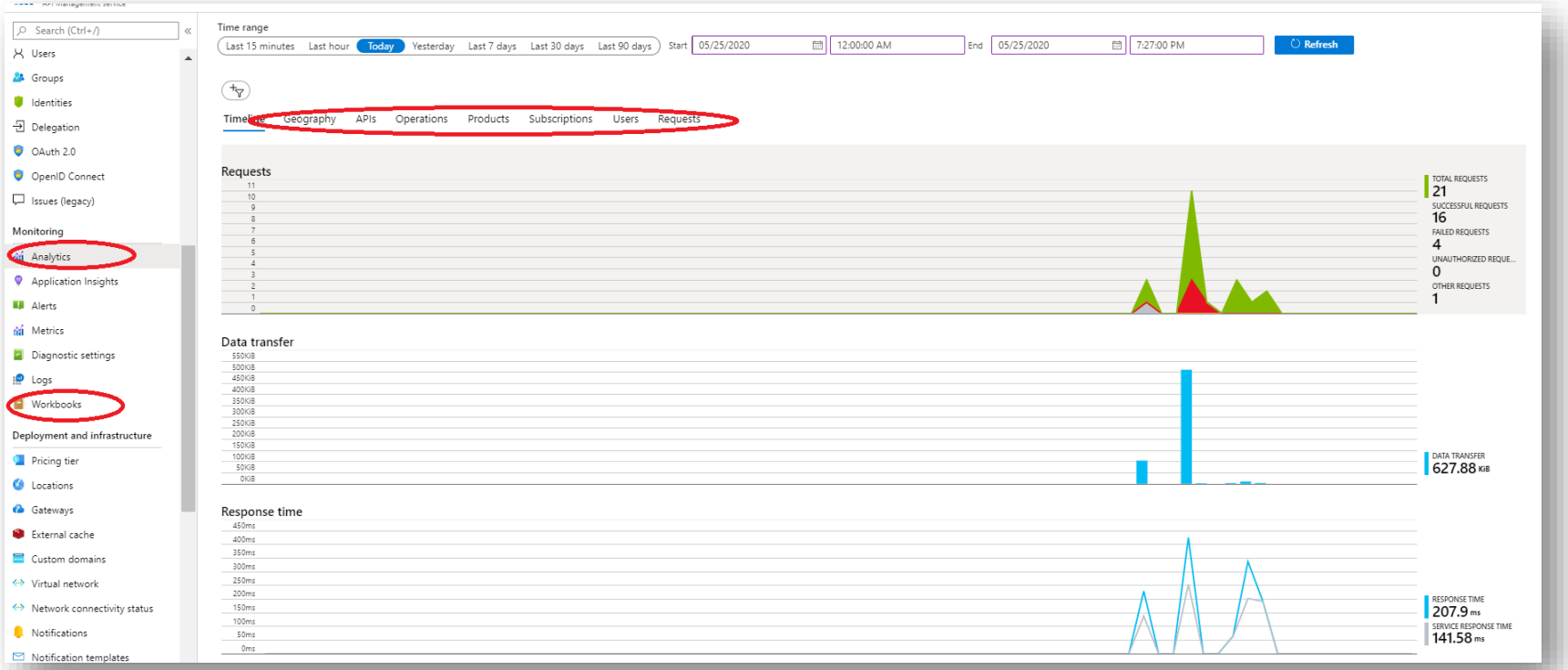



```
    <on-error>
        <!-- statements to be applied if there is an error
             condition go here -->
  </on-error>
</policies>
```


phany - APIs - Microsoft Az
Contoso API Portal
Application map - Microsoft Az
ecure | https://portal.azure.com/?feature.customportal=false#@microsoft.onmicrosoft.com/resource/subscriptions/c6a33fd3-e442-48a4-b82d-bcc4ad8a71d7/resourceGroups/Api-Default-West-US/providers/microsoft.insight...
Azure
Search resources, services, and docs
mijiang@microsoft.co...
MICROSOFT
me > apiphany-apim-ai - Application map
apiphany-apim-ai - Application map
Application Insights - Last 24 hours - apiphany-apim-ai
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
VESTIGATE
Application map
Smart Detection
Live Metrics Stream
Metrics Explorer
Metrics (preview)
Search
Availability
Failures
Performance
Servers
Browser
Time range = Last 24 hours
Feedback
Learn more
Refresh
Update map components
conferenceapi
BACKEND
87.8 ms | 100%
6 calls
1
instance
16%
129.3 ms
1.8K calls
apiphany West US
81 ms
119 calls
bigconference
BACKEND
bigconfere...t/sessions
BACKEND
897.8 ms
22 calls
1
instance
78%
169 ms
672 calls
apiphany West Europe
487 ms
3 calls
api.darksk...s&units=us
BACKEND

Analytics

- Built in dashboard in Azure Portal.

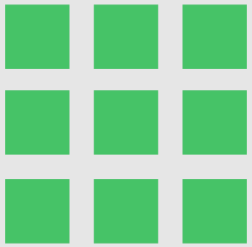


- Create custom workbooks

Security and protection

App developers

- Username/Password
- Microsoft account
- Google account
- Facebook account
- Twitter account
- Azure AD (Premium)
- Azure AD B2C (Premium)
- Delegated



1st and 3rd party apps

- Key
- OAuth 2
- OpenID Connect
- Client certificate
- IP filter
- Rate limits and quotas

API managers and developers

- Azure account
- RBAC

Developer portal

Consume

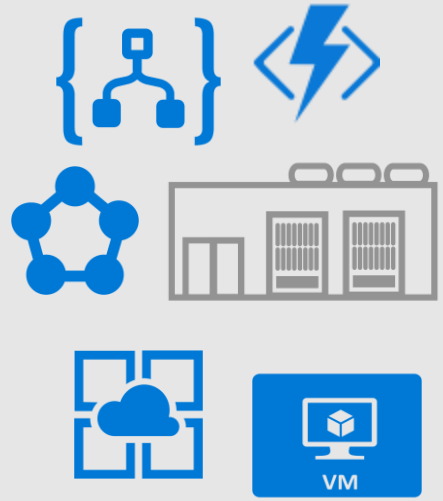
Gateway

Mediate

Azure portal

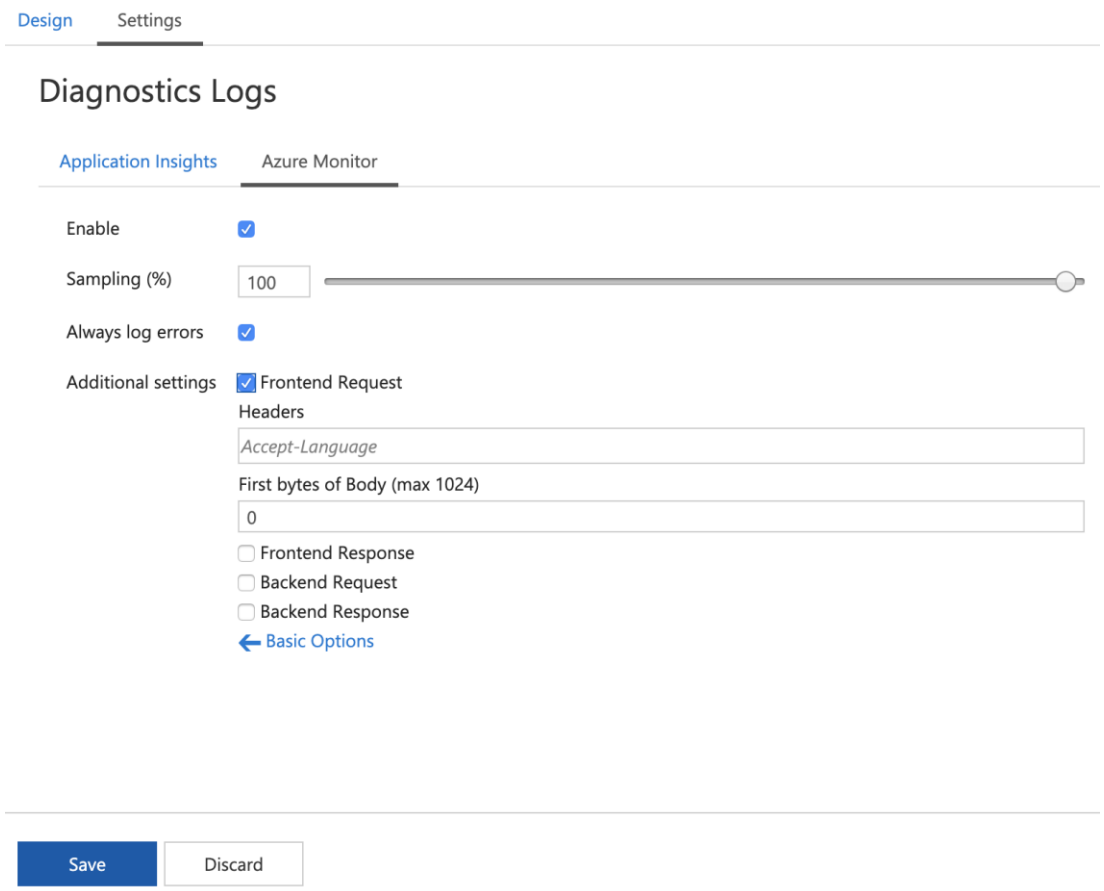
Publish

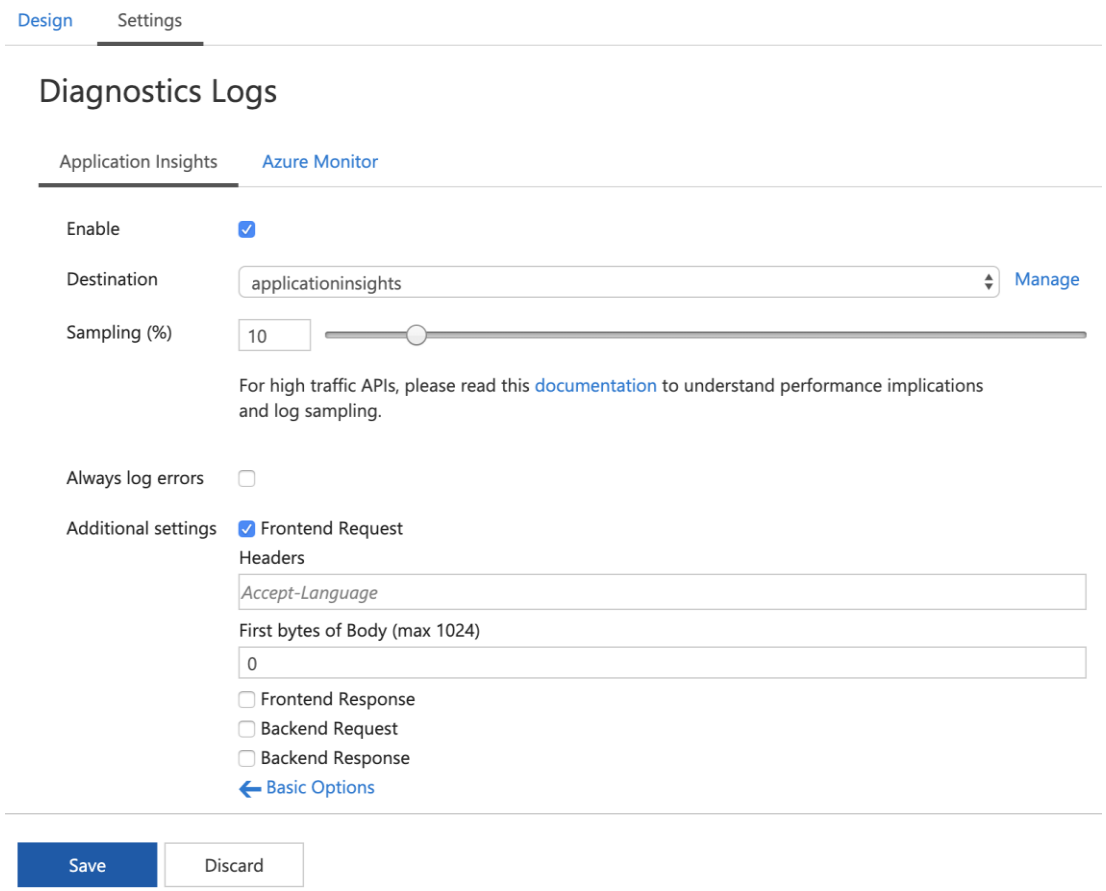
- HTTP Basic
- Mutual certificate
- Shared secret
- IP filter
- VNET/NSG



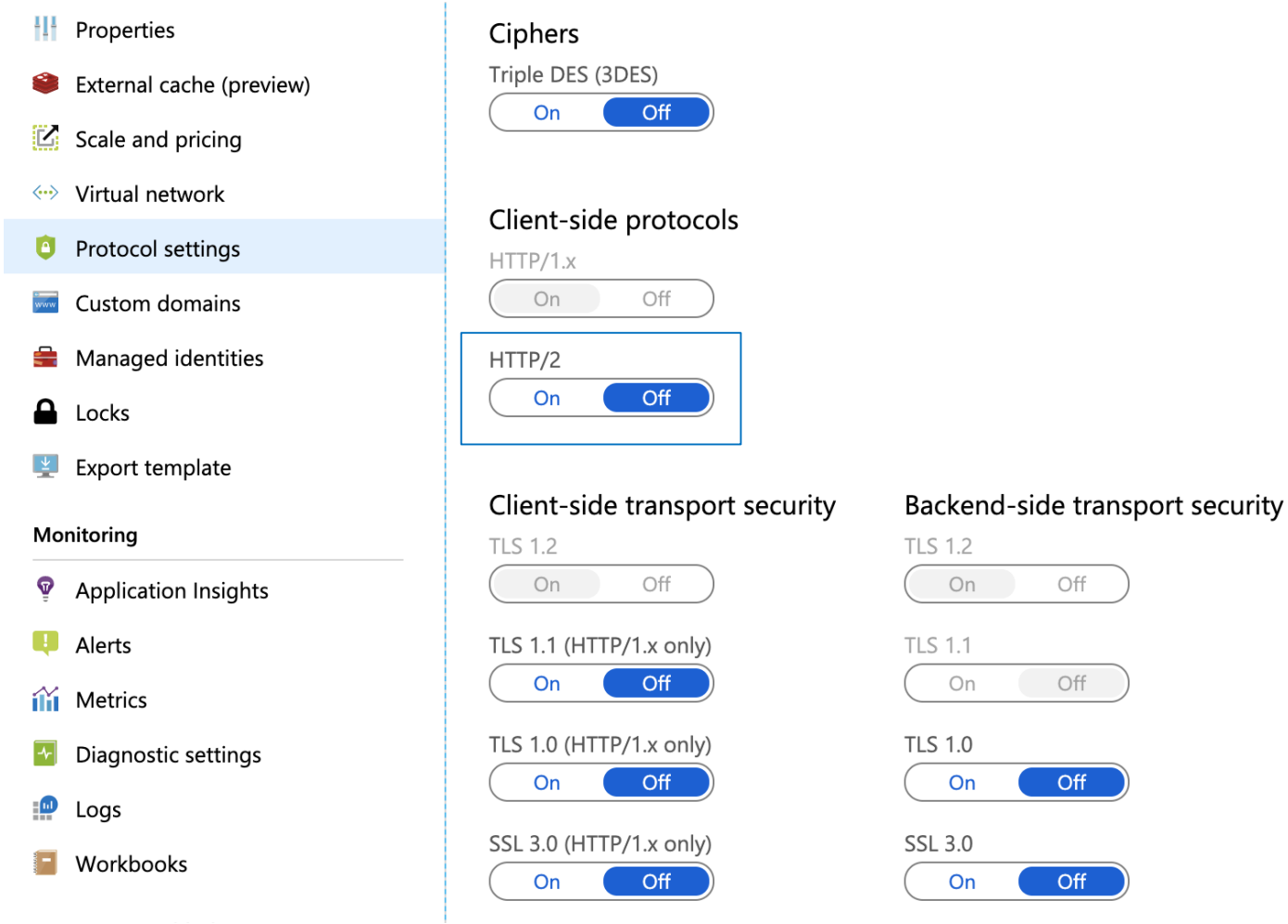
APIs on Azure and outside

Observability

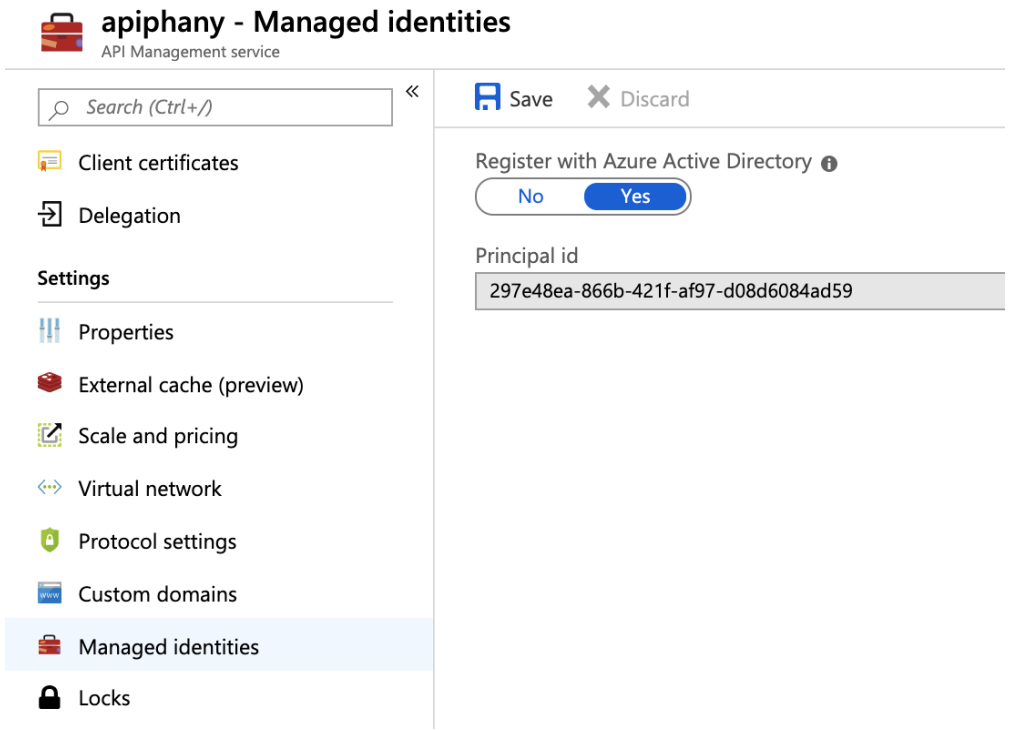




Protocol settings



Managed identity



Turn on managed identity

```
<authentication-managed-identity
    resource={{resource-url}}
    output-token-variable-name="token"
    ignore-error="false" />
```

Use the policy to add access token

JWE

```
<validate-jwt header-name="Authorization" require-scheme="Bearer" output-token-variable-name="jwt">
    <issuer-signing-keys>
        <key>{{jwt-signing-key}}</key>
    </issuer-signing-keys>
    <decryption-keys>
        <key>{{jwt-encryption-key}}</key>
    </decryption-keys>
    <audiences>
        <audience>@(context.Request.OriginalUrl.Host)</audience>
    </audiences>
    <issuers>
        <issuer>contoso.com</issuer>
    </issuers>
</validate-jwt>
<choose>
    <when condition="@(context.Request.Method == "POST" &&
                       !((Jwt)context.Variables["jwt"]).Claims["group"].Contains("contributor"))">
        <return-response>
            <set-status code="403" reason="Forbidden" />
        </return-response>
    </when>
```

JWE

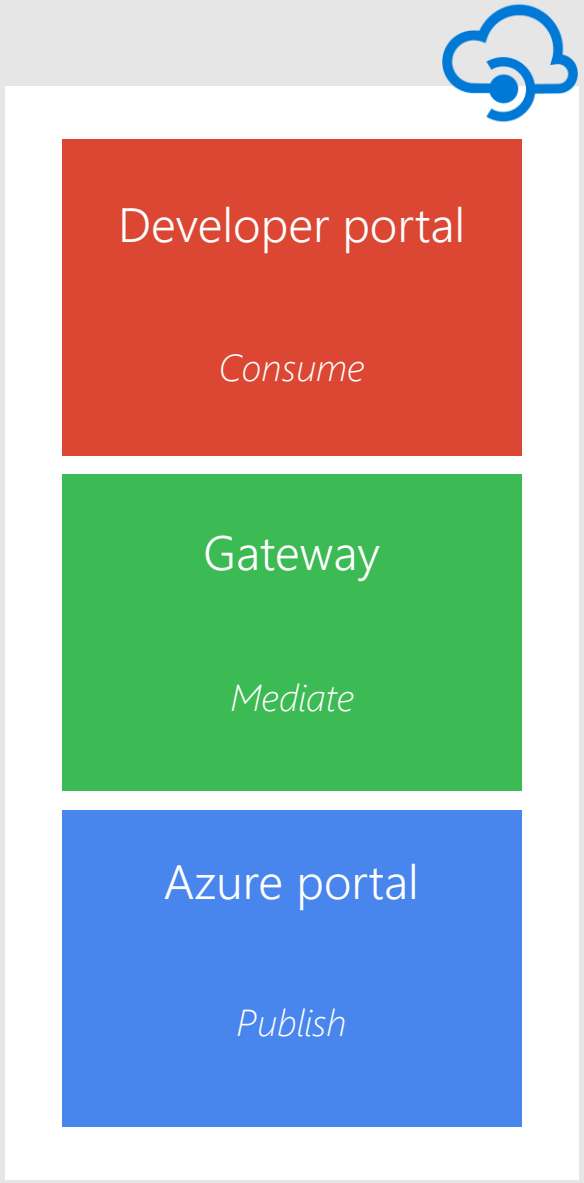
```
<validate-jwt header-name="Authorization" require-scheme="Bearer">
    <issuer-signing-keys>
        <key>{{jwt-signing-key}}</key>
    </issuer-signing-keys>
    <decryption-keys>
        <key>{{jwt-encryption-key}}</key>
    </decryption-keys>
    <audiences>
        <audience>@(context.Request.OriginalUrl.Host)</audience>
    </audiences>
    <issuers>
        <issuer>contoso.com</issuer>
    </issuers>
</validate-jwt>
```

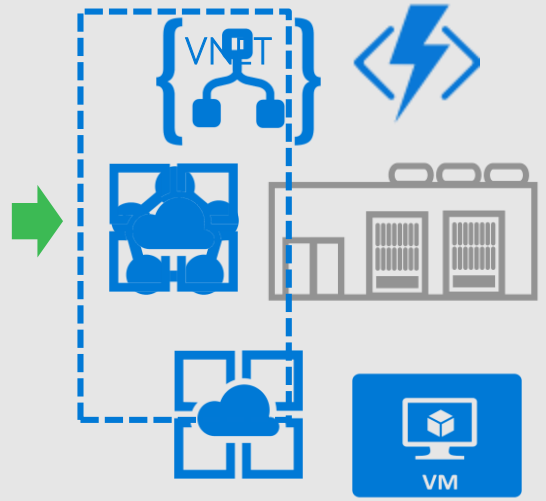
VNETs and Hybrid

App developers

1st and 3rd party apps

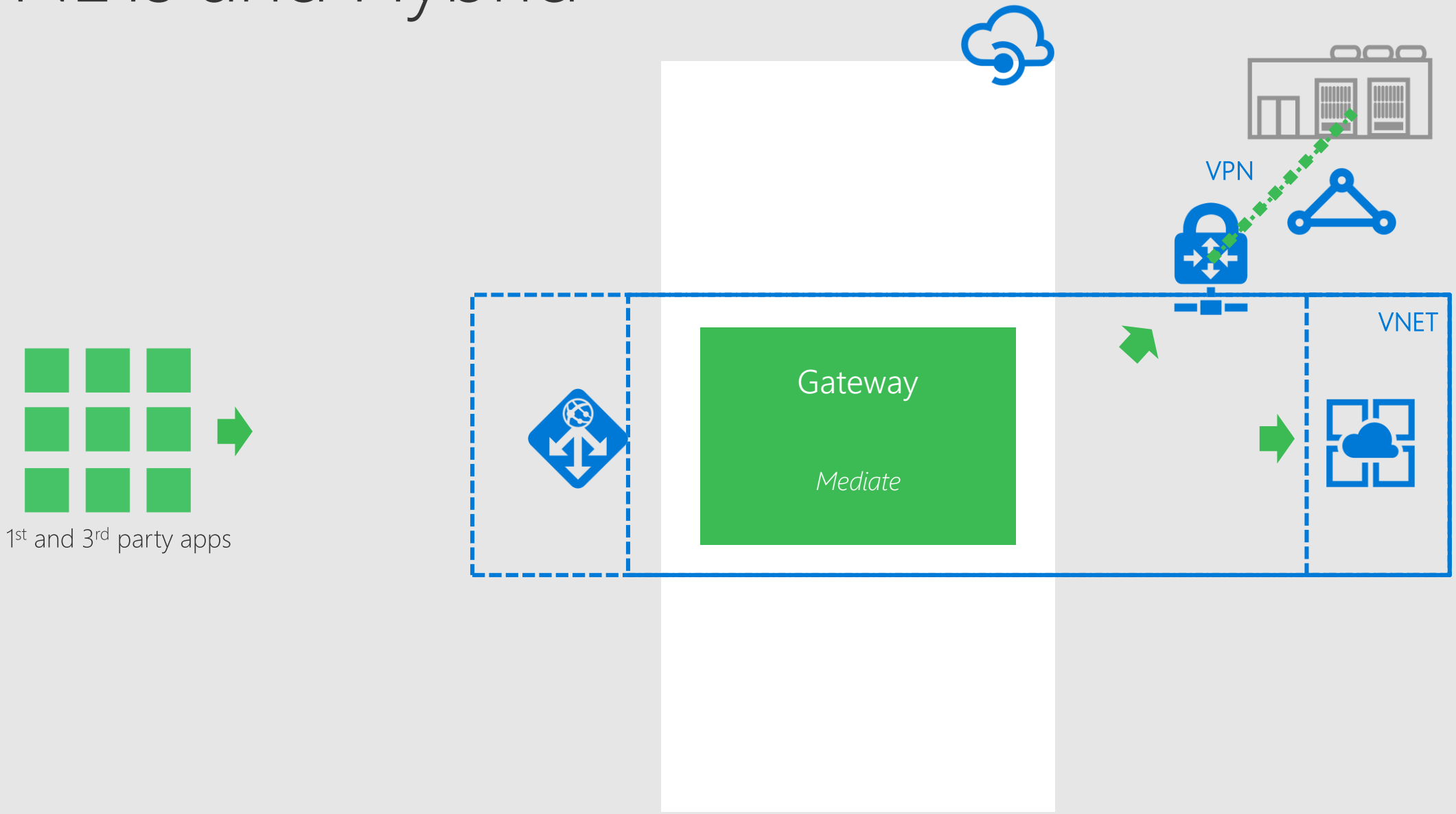
API managers and developers





APIs on Azure and outside

VNETs and Hybrid

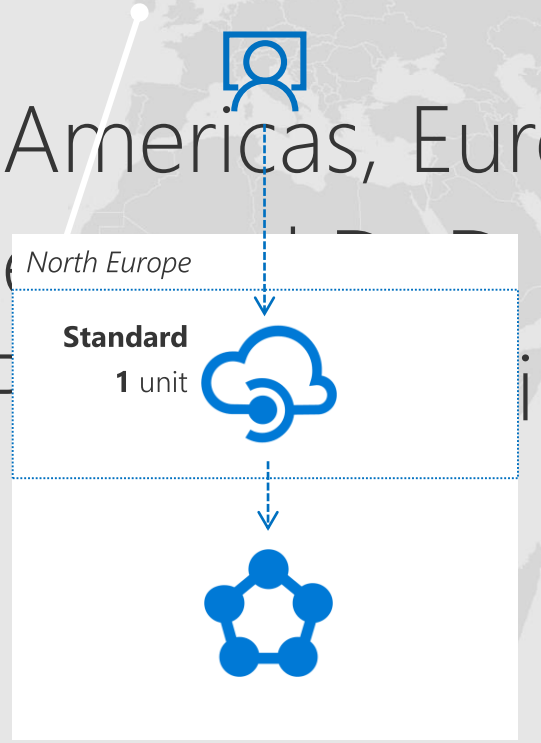


Multi-region and scaling

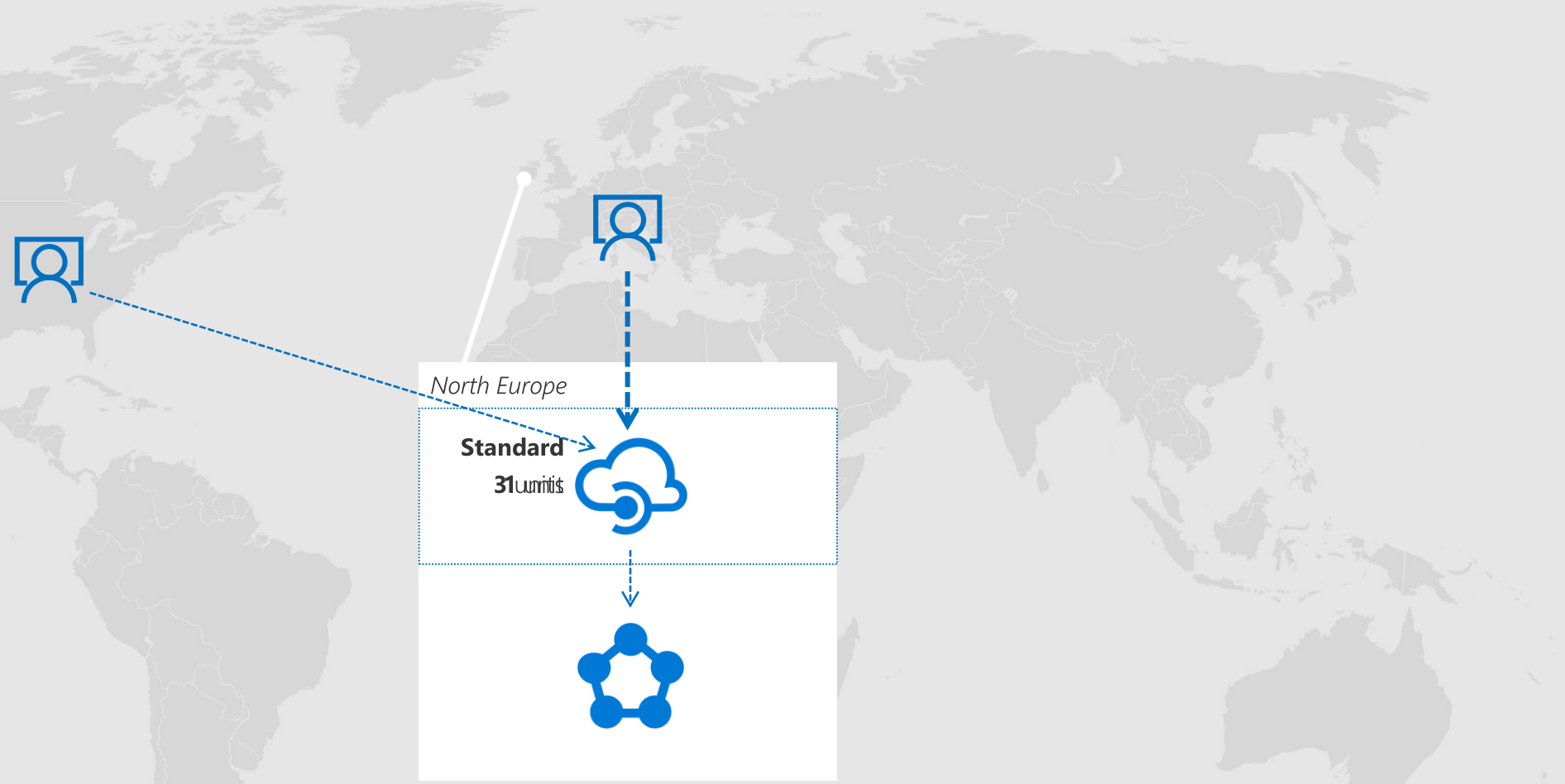
27 public regions in Americas, Europe, Asia and Australia

6 US governm regions (preview)

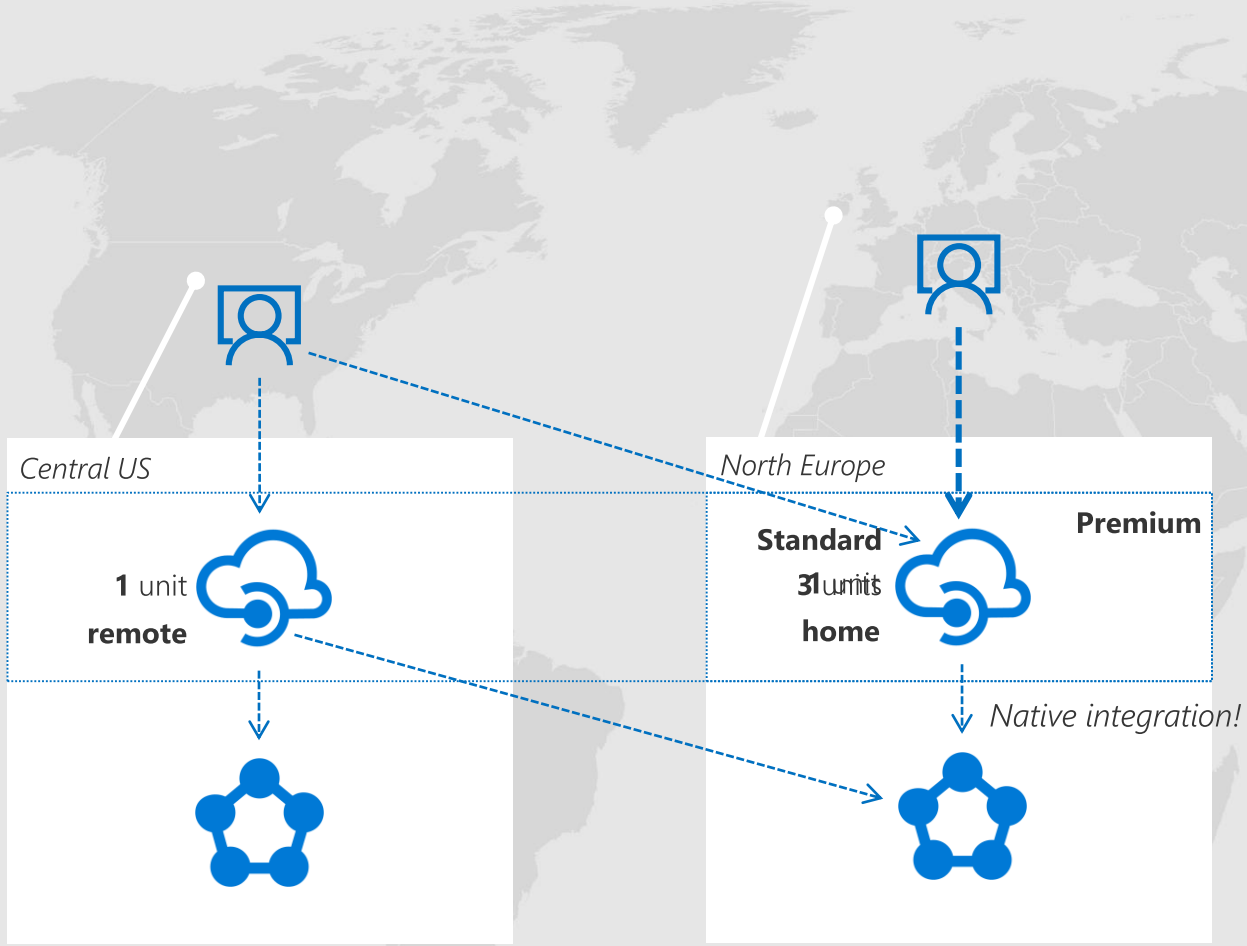
P ina!



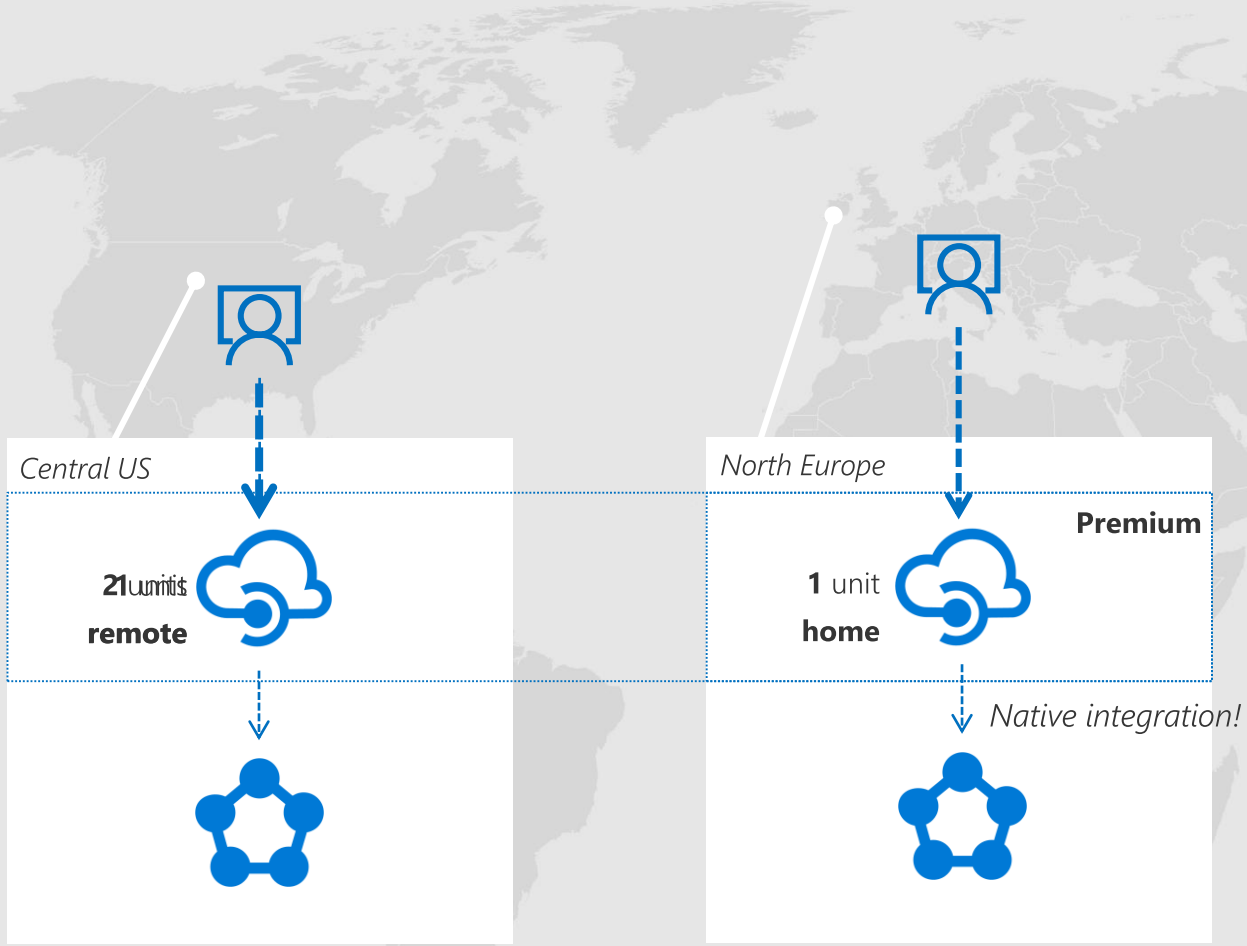
Multi-region and scaling



Multi-region and scaling



Multi-region and scaling



Self-hosted API Management gateway

Can be hosted on premises or in any cloud

- Packaged as a Docker container image (Linux only)
- Functionally equivalent to the managed gateway

Federated with an API Management service instance

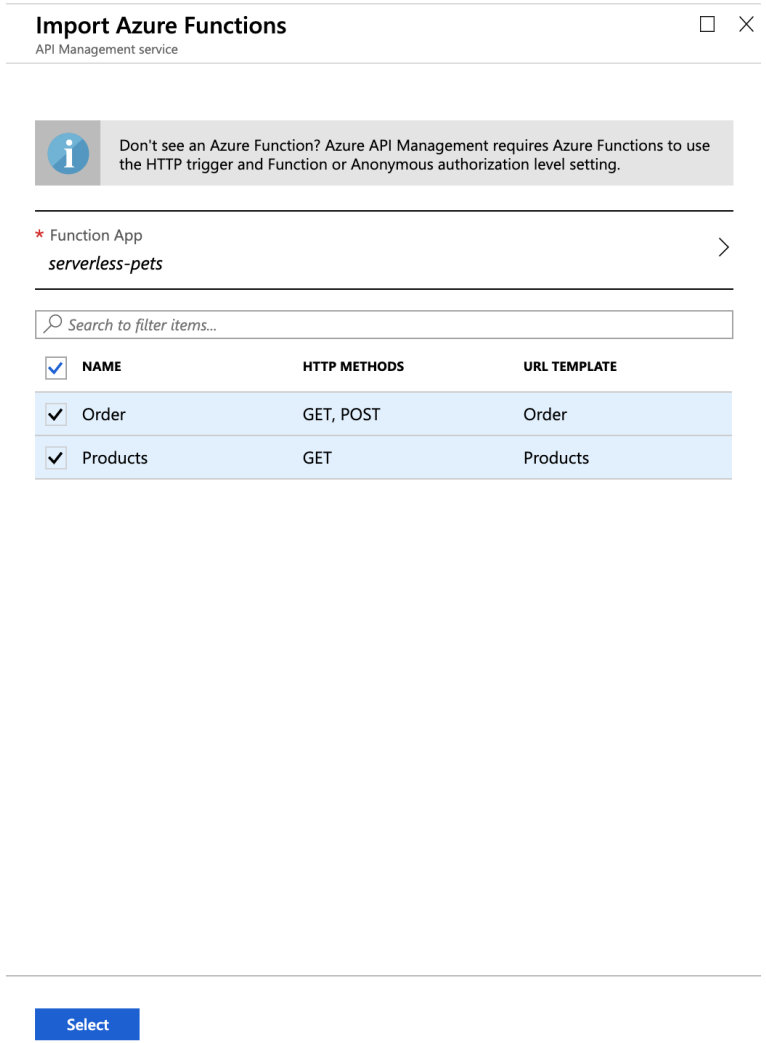
- Gateway pulls down configuration and pushes up telemetry
- Gateway needs only an outgoing connection to Azure

Kubernetes is the recommended hosting environment

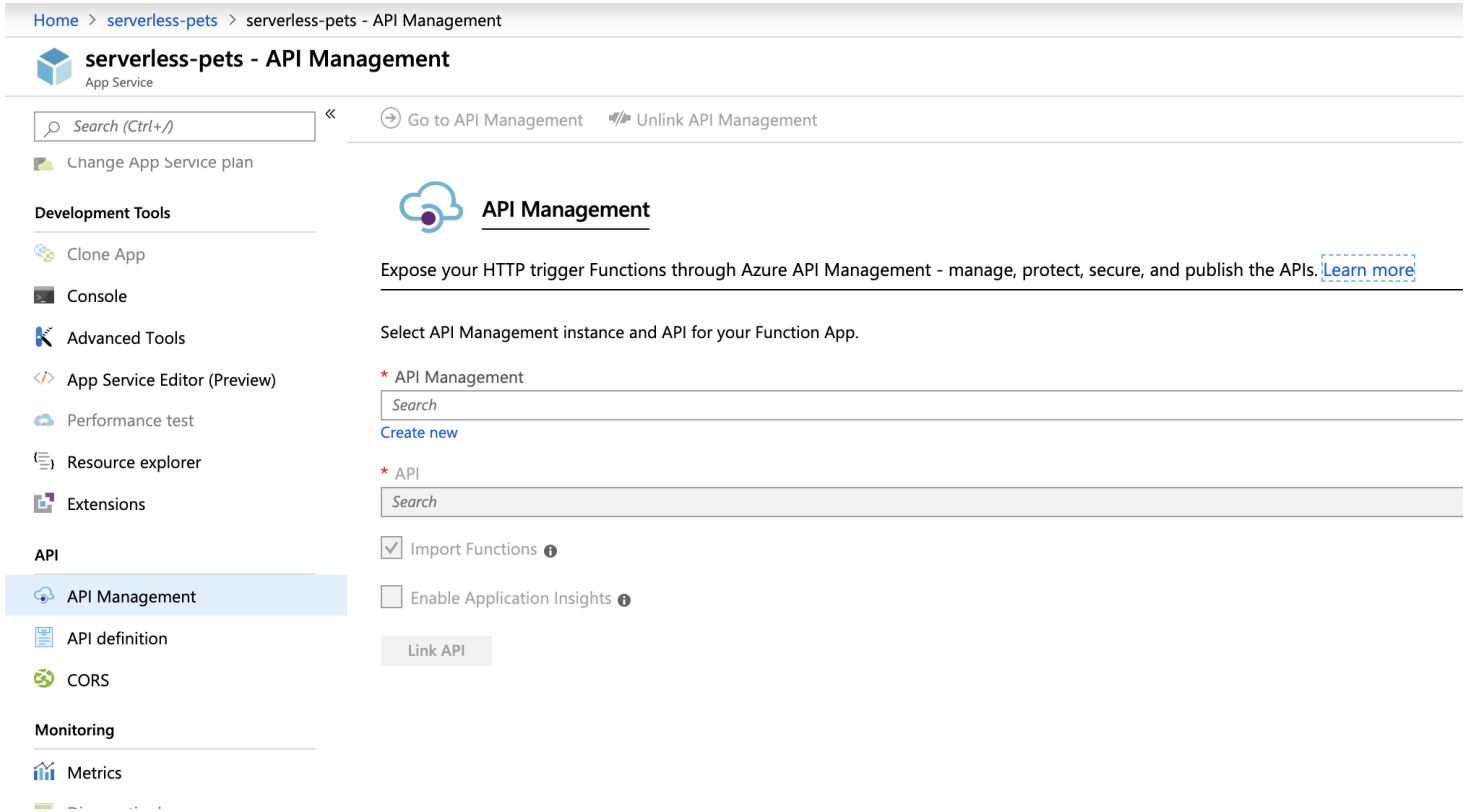
- Simplifies deployment, scaling, updates, availability
- Other container orchestrators will work too

Preview is coming in the late summer or early fall!

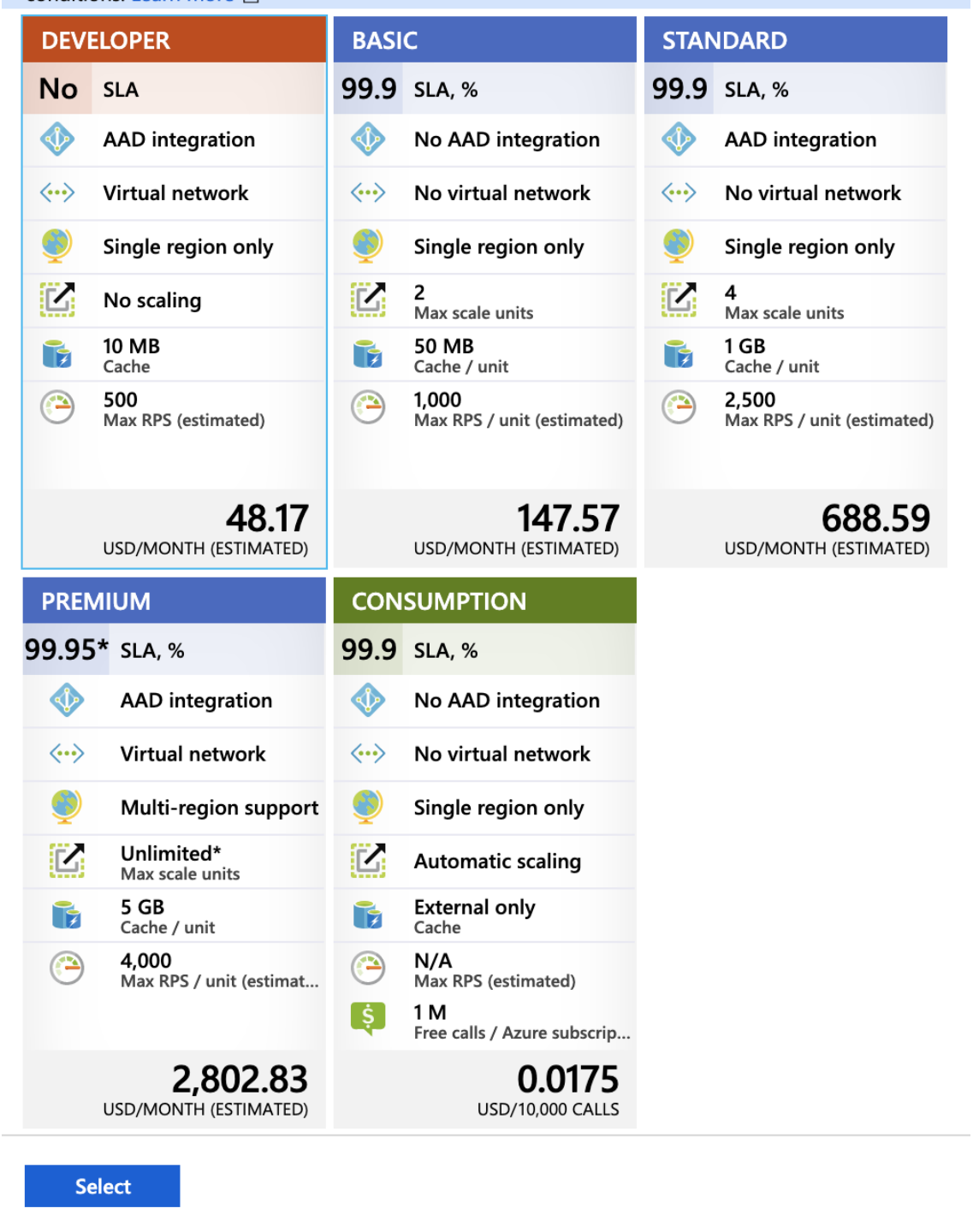
API Management



Functions



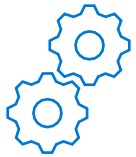
Consumption tier



Consumption tier

Generally available

North Central US, West US, West Europe, North Europe, Southeast Asia, and Australia East. Additional regions coming soon.



Serverless API Management

On-demand activation, auto-scale out and back to zero, consumption-based micro billing

Façade for serverless endpoints and container-based microservices

Functions, Logic Apps, Kubernetes, Service Bus, Event Hubs, Storage, etc.

Curated set of features and usage limits

E.g. no developer portal or built in cache

VS Code extension

Resource explorer

- List, create, delete APIM instances
- List, create, delete sub-resources
  - APIs, operations, products, named values, loggers

Policy editing

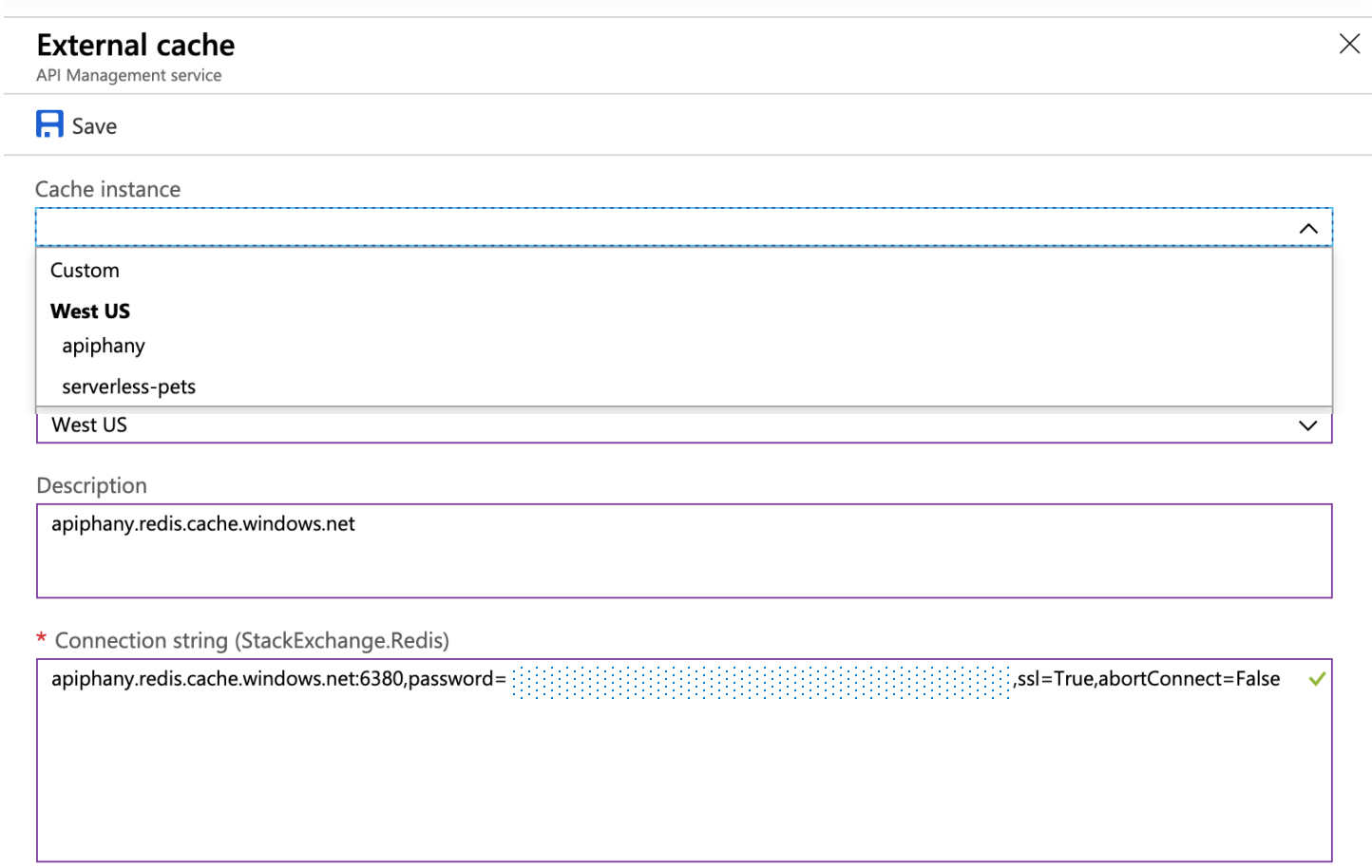
- XML and C# syntax check
- IntelliSense (code completion, hover for parameter description)
- Policy snippets for common scenarios

Remote and local testing

Integrated with automation tools

Please reach out to me if you want to provide feedback or see additional scenarios

Bring your own cache



Bring your own cache

```
<cache-lookup caching-type="prefer-external" vary-by-developer="true" vary-by-developer-groups="true" downs
    <vary-by-header>header n
    <vary-by-query-parameter
</cache-lookup>
```

```
external
internal
prefer-external
```

```
<cache-lookup-value caching-type="prefer-external" key="" default-value="" variable-name="" />
```

```
external
internal
prefer-external
```

Transformation and Routing

- Transformation through policies
  - XML to JSON and vice-versa.

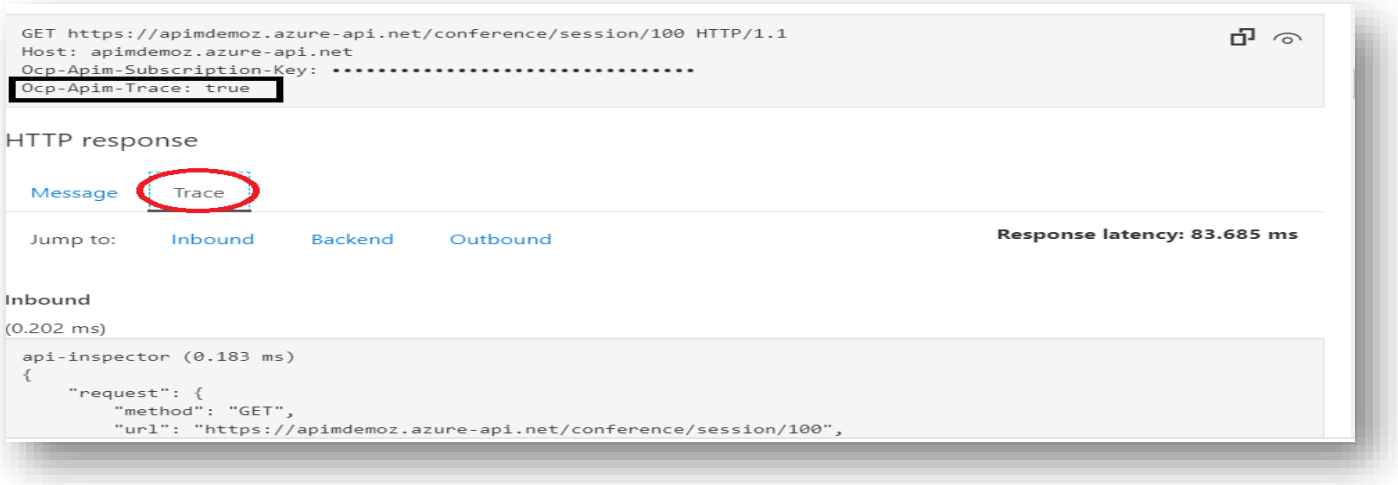
```
<json-to-xml apply="always | content-type-json" consider-accept-header="true | false" parse-date="true | false"/>
<xml-to-json kind="javascript-friendly | direct" apply="always | content-type-xml" consider-accept-header="true |
```

- SOAP to REST
  - Supports SOAP to REST during the API Import itself. Custom transformation policies could be written as well.

- Basic Routing during API/Operation definitions, Advanced and conditional routing through policies.

Tracing and Error Handling

- Tracing
  - Inspect request processing across different stages (Inbound, Backend & Outbound)



- Error Handling
  - Processing immediately jumps to the on-error policy section
  - Can configure custom behavior or creating a new response to return to the caller.
  - Rich set of Error objects populated with details about the error such as source, reason, message and scope etc

```
    <on-error>
        <!-- statements to be applied if there is an error
             condition go here -->
  </on-error>
</policies>
```

Logging and Monitoring

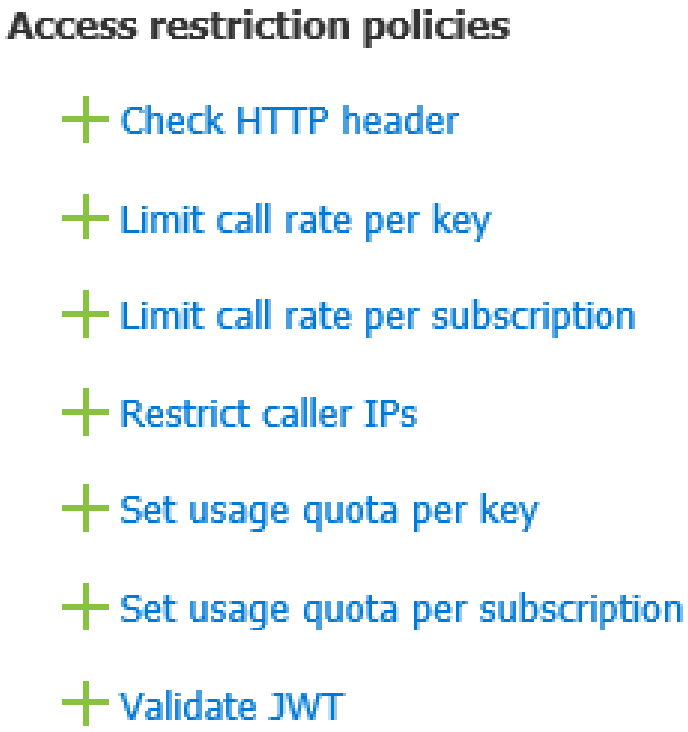
- Configuration
  - Integration with Application Insights and Azure Monitors.
  - Can be setup at API level or the Operation level
  - API Gateway specific metrics
  - Can be archived to Log Analytics, Azure storage or Event Hub.
- Logs Types
  - API Metrics
    - Gateway Requests, Capacity, Events, Network status etc.
  - Resource Logs
    - Provide rich information about API requests and errors.
    - Batched hourly, logs every API requests.
  - Activity Logs
    - Audit log or Operational log
    - Get insight of all operations performed out such as modifying API, adding/removing users etc.
- Alerts
  - receive alerts based on metrics and activity logs.
  - Notify or take actions using Action groups.

Caching Policies

- Cache full responses:
  - Store to cache / Get from cache
- Cache partial values
  - Store value in cache / Get value from cache

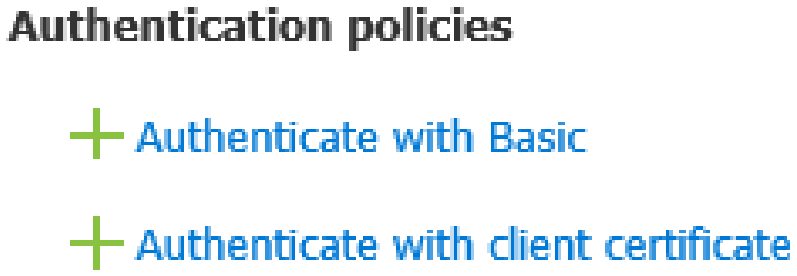
Access restriction Policies

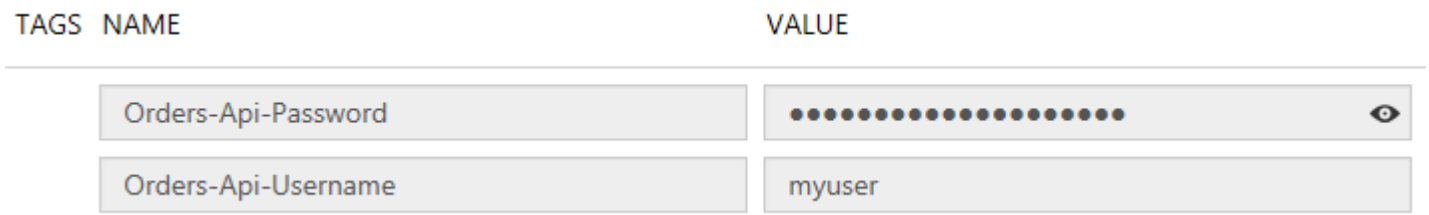
- Check HTTP header:
  - Validate request contains header with value
- Call rates:
  - Per subscription or per key
- Quota
  - Per subscription
- Restrict caller Ips
  - Restricts where api can be called from
- Validate JWT
  - Validate JWT tokens before calling API



Authentication Policies

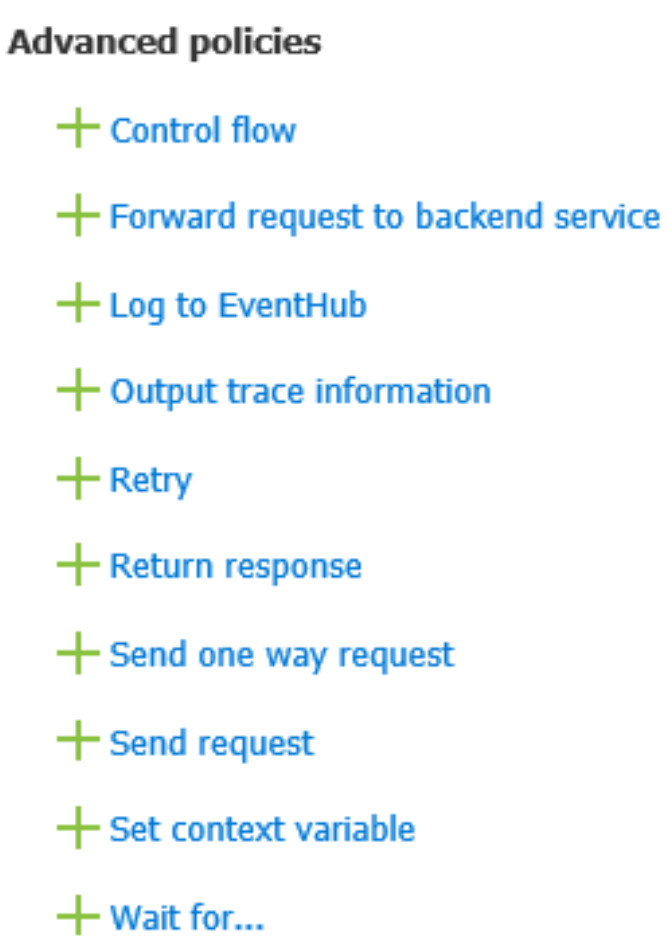
- Secure connection between API gateway and API
  - Basic authentication
  - TLS Mutual Authentication
- Use Named Values to avoid hardcoding sensitive values in policy






```
<backend>
    <authentication-basic username="{{Orders-Api-Username}}" password="{{Orders-Api-Password}}" />
    <base />
</backend>
```

Advanced Policies



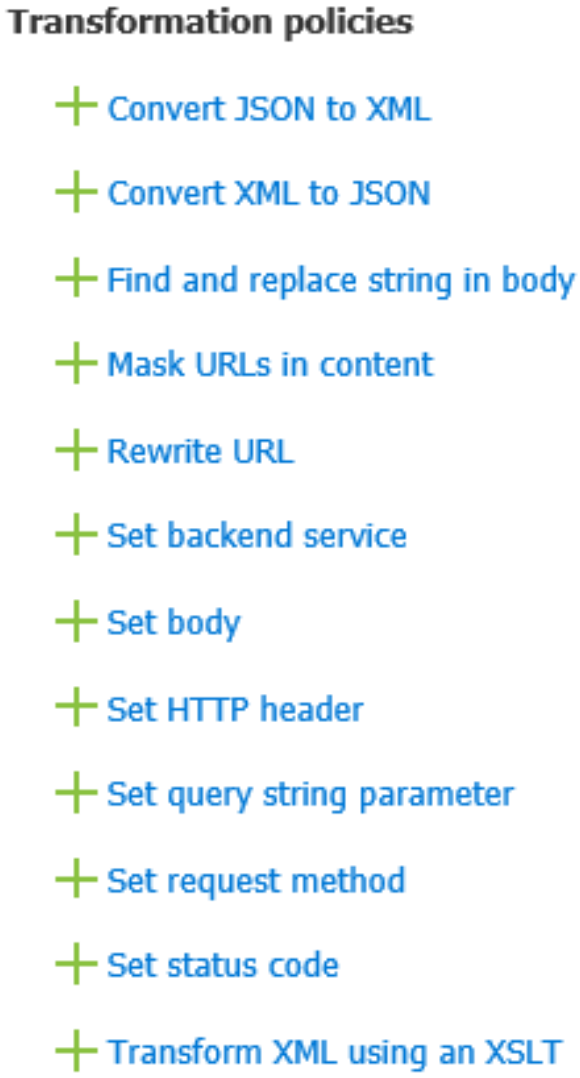
- Control Flow == When/Case *
- Forward Request == send request to backend
- Retry == implement retry policy
- Return an HTTP response

Others can be used to implement fan-out, logging / audit and other patterns

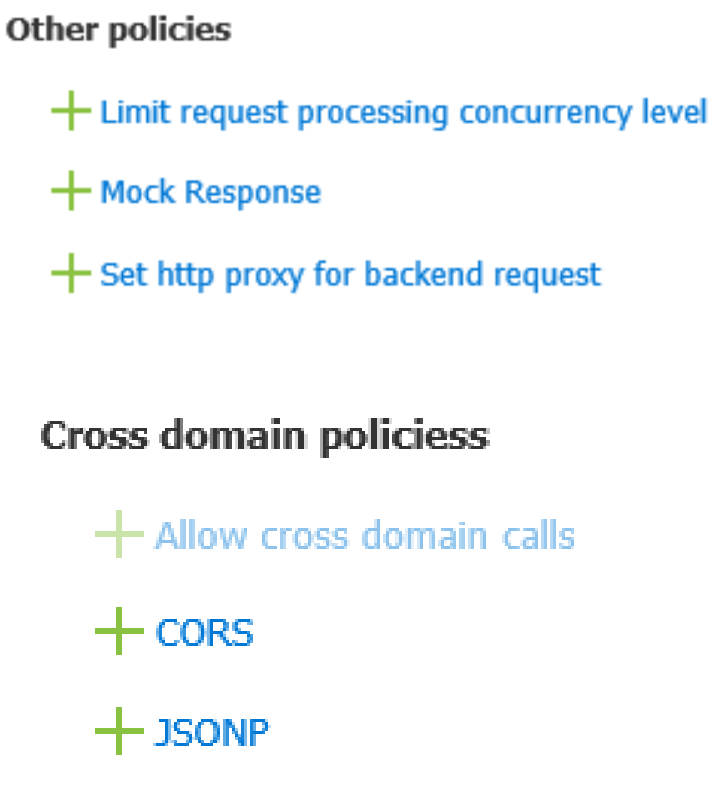
* Combined with Set-Backend-Url can be used for Dynamic routing

Transformation Policies

- Content Conversion
  - XML <-> JSON
- Modifying request or responses
  - Masking, replace strings
- Rewriting URLs
  - Pretty URLs exposed and converted internally
- Change backend URL
- Change request / response
- Apply XSLT



Other Policies



- Concurrency
- Mocking service responses (during dev)
- Proxy URL
- CORS policies

Effective Policy

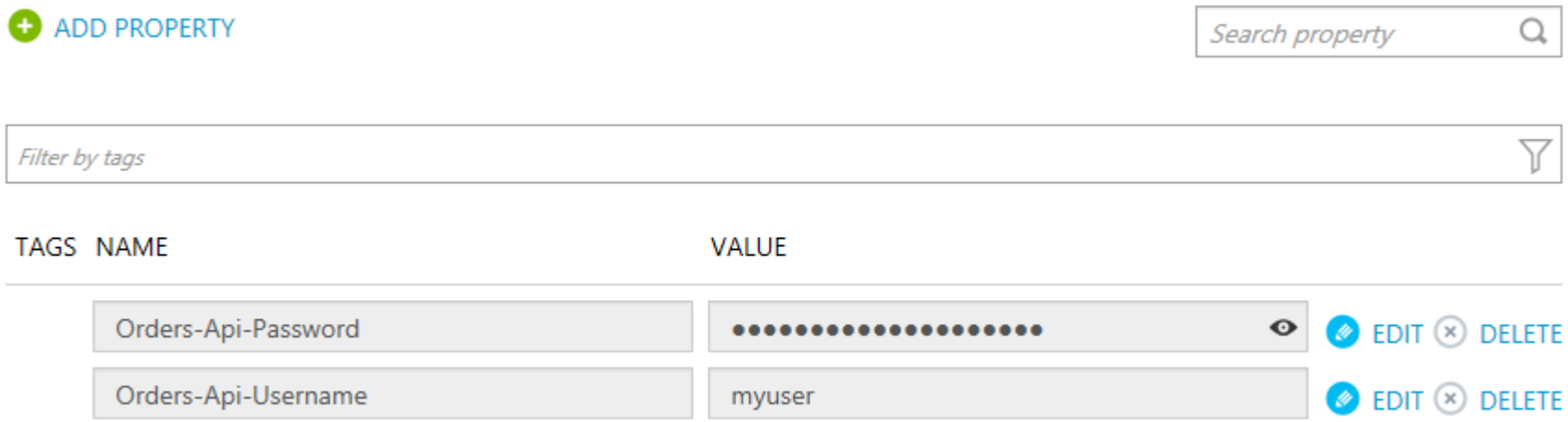
- Combination of Product / Api / Operation policies
- Keep it simple
- Some policies only make sense in certain scopes
  - Example: Subscription quotas only make sense at the Product level

Effective Policy

- Combination of Product / Api / Operation policies
- Keep it simple
- Some policies only make sense in certain scopes
  - Example: Subscription quotas only make sense at the Product level

Properties

- Key / Value Store
- Provides reusable values across policies
- Values can be secret (not displayed in UI)
- Used with “{{propertyName}}” in policies



Policy Expressions

- C# 6.0 expressions embedded in the XML policy
- Has implicit context variable
- Can use a subset of .NET Framework Types

```
@(expression)
```

Single statement
Expression

```
@{expression}
```

Multi-statement
Expression

Policy expressions

```
<policies>
    <inbound>
        <base/>
        <set-variable name="bodySize" value="@(context.Request.Headers["Content-Length"][0])"/>
        <choose>
            <when condition="@(int.Parse(context.Variables.GetValueOrDefault<string>("bodySize")) < 256*1024)">
            </when>
            <otherwise>
                <rewrite-uri template="{{alternate-path-and-query}}"/>
                <set-backend-service base-url="{{alternate-host}}"/>
            </otherwise>
        </choose>
    </inbound>
    <outbound>
        <base/>
    </outbound>
</policies>
```

Policy Expressions - Samples

```
@(true)
```

```
@((1+1).ToString())
```

```
@("Hi There".Length)
```

```
@(Regex.Match(context.Response.Headers.GetValueOrDefault("
Cache-Control",""), @"max-
age=(?<maxAge>\d+)").Groups["maxAge"]?.Value)
```

```
@(context.Variables.ContainsKey("maxAge") ?
int.Parse((string)context.Variables["maxAge"]) : 3600)
```

Policy Expressions - Samples

```
@{
  string value;
  if (context.Request.Headers.TryGetValue("Authorization", out value))
  {
    return Encoding.UTF8.GetString(Convert.FromBase64String(value));
  }
  else
  {
    return null;
  }
}
```

Policy Expressions - Samples

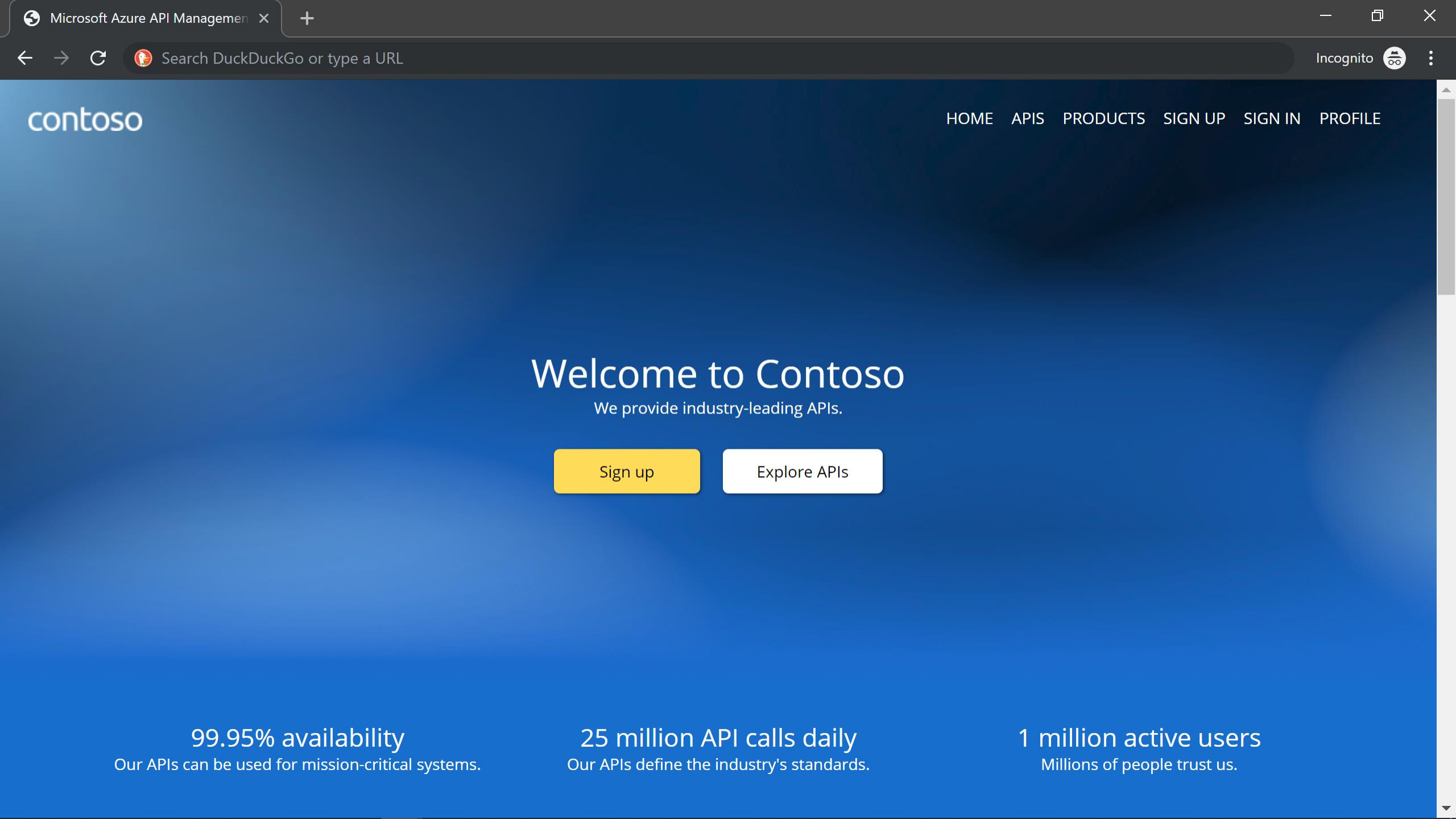
```
@(true)
```

```
@((1+1).ToString())
```

```
@("Hi There".Length)
```

```
@(Regex.Match(context.Response.Headers.GetValueOrDefault("
Cache-Control",""), @"max-
age=(?<maxAge>\d+)").Groups["maxAge"]?.Value)
```

```
@(context.Variables.ContainsKey("maxAge") ?
int.Parse((string)context.Variables["maxAge"]) : 3600)
```


Microsoft Azure API Managemen
Search DuckDuckGo or type a URL
Incognito
contoso
HOME
APIS
PRODUCTS
SIGN UP
SIGN IN
PROFILE
Welcome to Contoso
We provide industry-leading APIs.
Sign up
Explore APIs
99.95% availability
Our APIs can be used for mission-critical systems.
25 million API calls daily
Our APIs define the industry's standards.
1 million active users
Millions of people trust us.