Integration Technical Conference 2019

V2018 DataPower Gateway Architectural Deep Dive

Krithika Prakash, STSM - krithika.p@ibm.com
Jeremy Geddes, Technical Lead - jgeddes@us.ibm.com
API Connect & Gateways

IBM **Cloud**

IBM

Important Disclaimers

- **IBM Confidential**. Unless specifically advised otherwise, you should assume that all the information in this presentation (whether given in writing or orally) is IBM Confidential and restrict access to this information in accordance with the confidentiality terms in place between your organization and IBM.
- **Content Authority**. The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- **Performance**. Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- **Customer Examples**. Any customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- **Availability**. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates

Trademark Acknowledgements

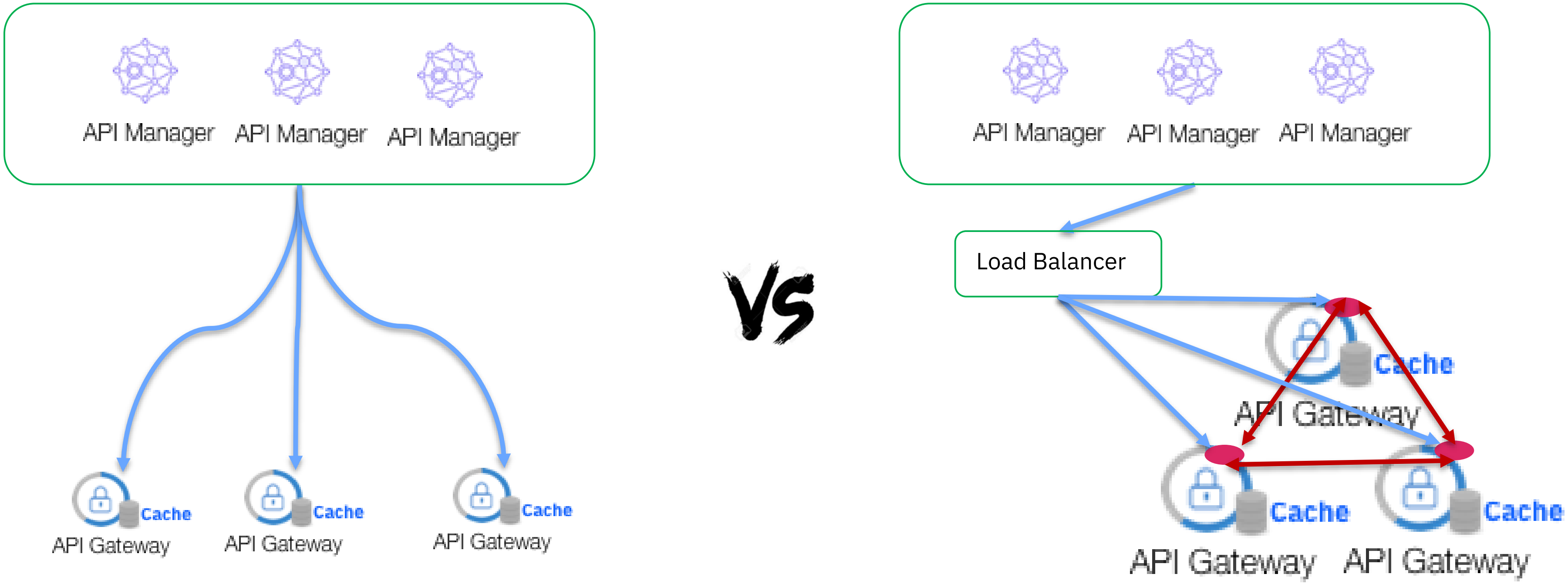
Agenda

- Gateway Service Management
  - Dataflow, High Availability, Scaling, Upgrade, Rate Limiting
- Gateway Service Flows
  - Configuration, Registration, Data Transmission
- Extending Gateway Services
  - Gateway Extensions, User Defined Policies, Global Policies
- Security – Architecture Deep Dive
  - Re-architected OAuth Provider
  - New Policies
- Trouble Shooting

Gateway Service Management

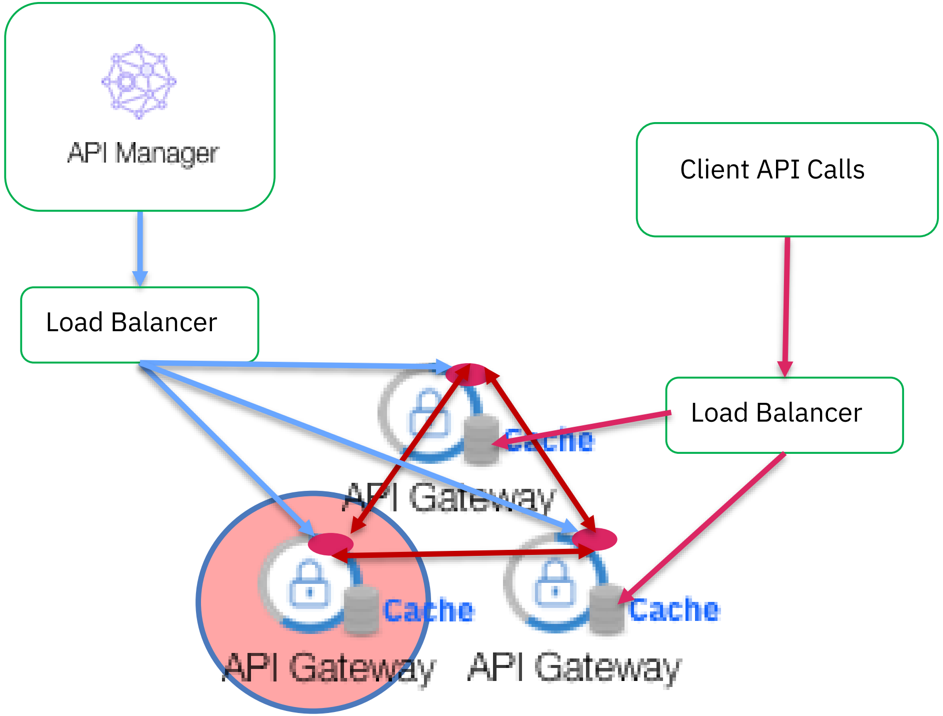
- Dataflow
- High Availability
- Scaling
- Upgrade
- Plan Rate Limiting

Gateway Service Dataflow



- APICv5 each gateway in the gateway service would communicate directly with API Manager, putting a lot of load on the server as total gateways increased.
- Now API Manager communicates through the Gateway Service Object on a per gateway service basis, rather than each individual gateway.
  - This reduces load on the Management Server.
  - Faster sync time between gateways in a gateway service.

Gateway Service High Availability



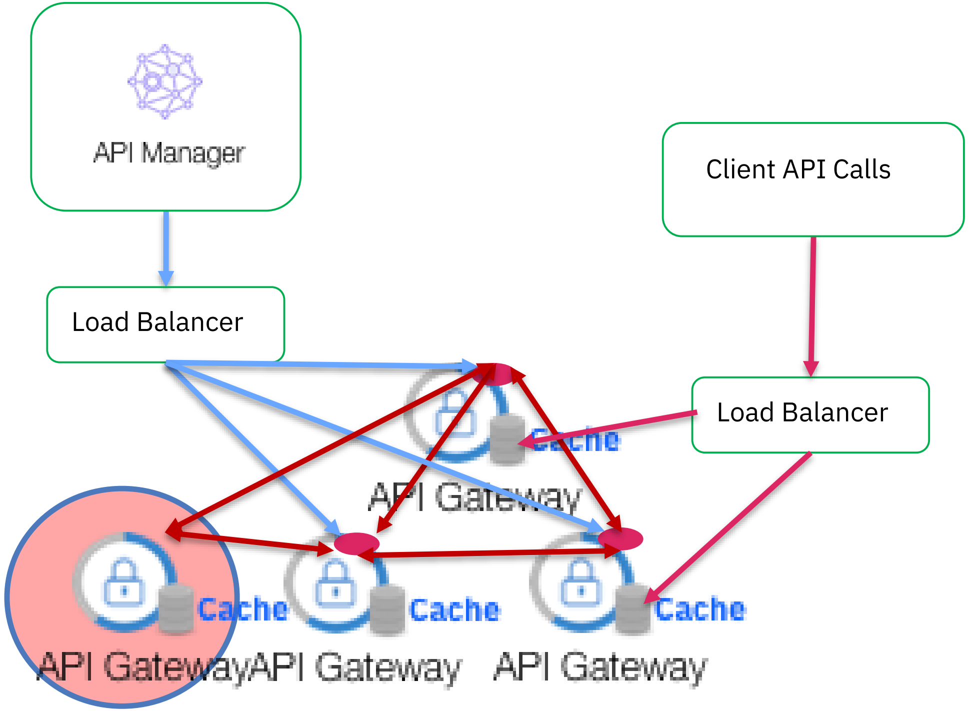
Gateway Peering requires minimum three members for HA (Quorum)

- Two members will provide API load balancing, but an outage of the primary will not result in failover

Gateway Peering persistence set to memory with one member will force resync with APIm (setting for helm charts we ship via apicup)

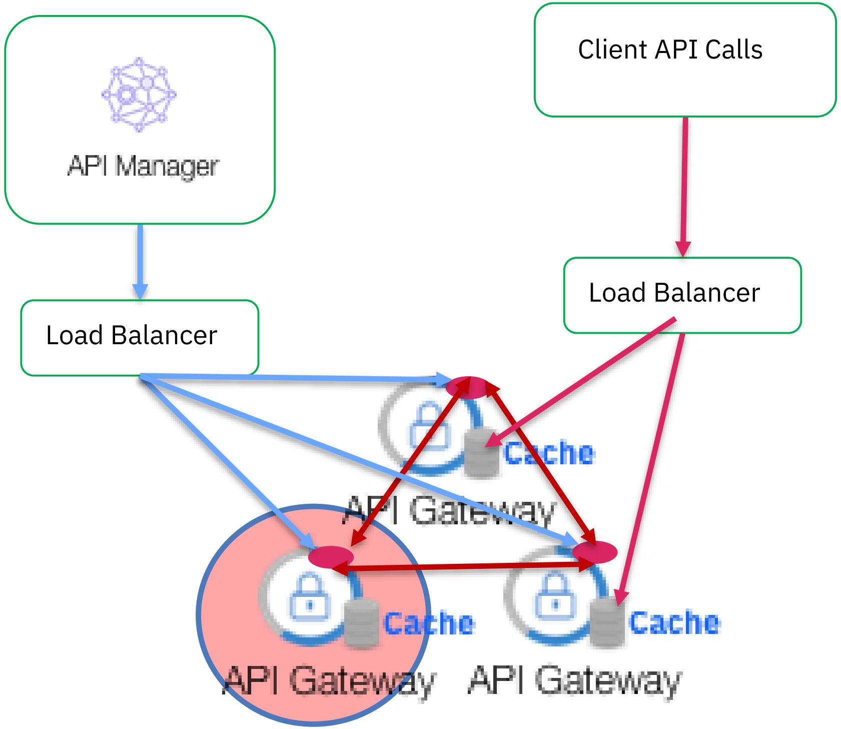
Gateway Peering persistence set to Local or Raid on appliance will cause the data to persist even in single member.

Gateway Service Scaling



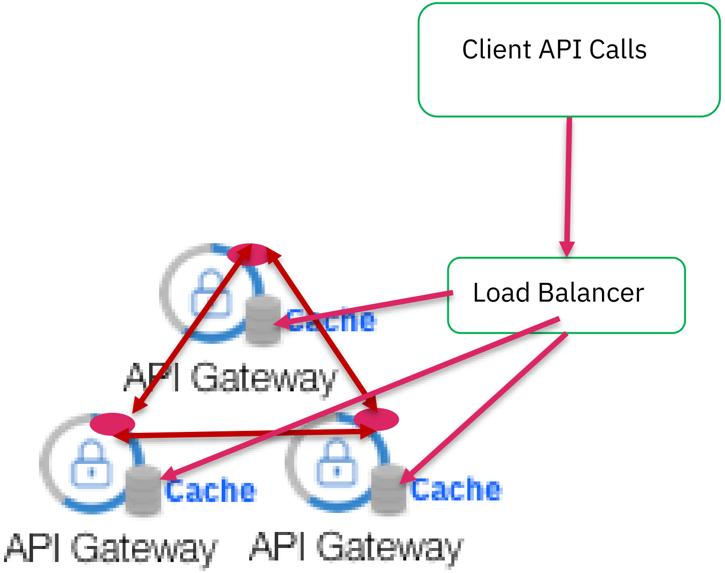
- Faster sync time between gateways in a gateway service.
  - Each individual gateway in a gateway service shares a common persistence layer.
  - Newly added gateways have access to the shared data immediately upon joining.

Gateway Service Upgrade



- All APIConnect components should be kept at same level.
- APImanager upgraded first
- Remove Gateway being upgraded from load balancer for client APIs
- New Gateway Service function available when all members upgraded.

Gateway Service Plan Rate Limiting



- Plan rate limiting was done via Multicast SLM in APICV5
- Plan rate limiting defaulted to Unicast SLM in APICv2018 for the V5C gateway to enable use in docker environment where multicast is not enabled
- Plan rate limiting in APICv2018 for API Gateway is now handled via Gateway Peering limiting required configurations.

Gateway Service Flows

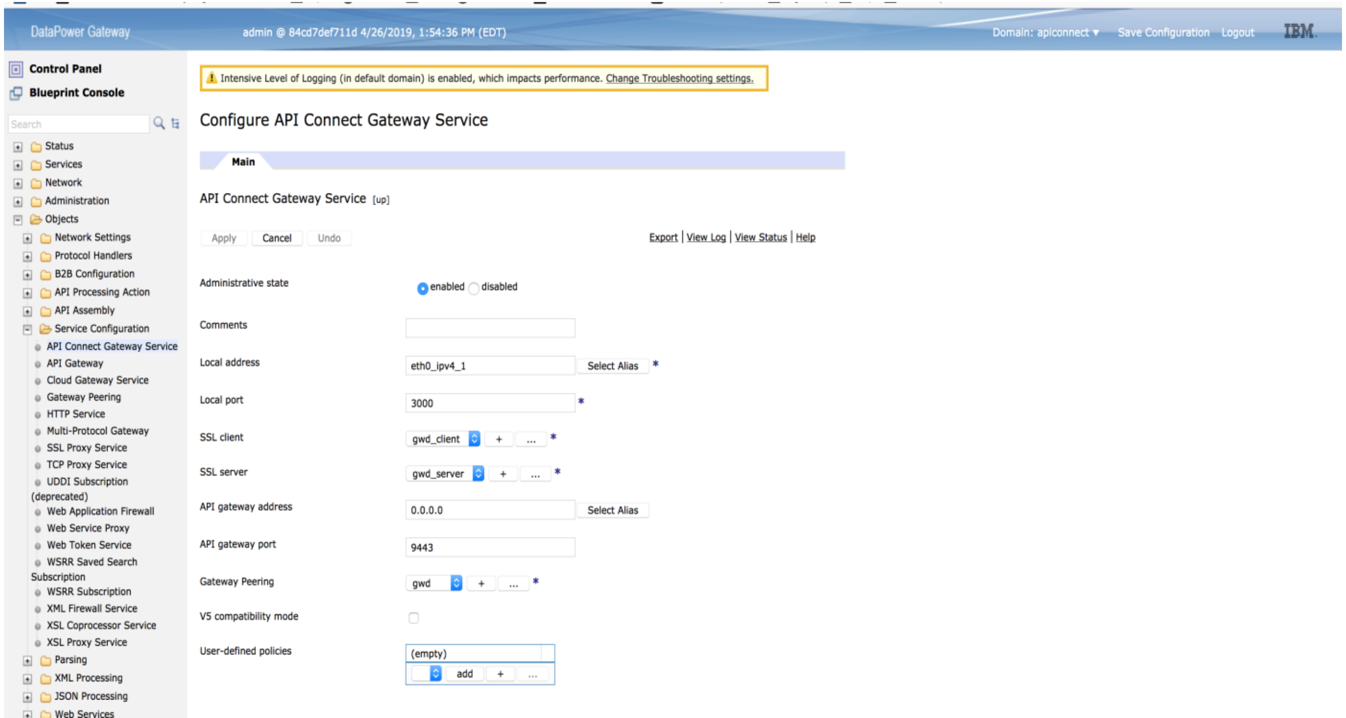
- Gateway Configuration
- APIm Registration
- Data Transmission

Gateway Configuration

– Gateway Service Object

- Is the actual Gateway Service Object deployed in the sidecar of a gateway. It enables APIC functionality on the gateway.

```
apic-gw-service
  v5-compatibility-mode off
  admin-state enabled
  ssl-client gwd_client
  ssl-server gwd_server
  local-address eth0_ipv4_1
  local-port 3000
  gateway-peering gwd
exit
```



⚠ Intensive Level of Logging (in default domain) is enabled, which impacts performance. Change Troubleshooting settings.

Configure API Connect Gateway Service

API Connect Gateway Service [up]

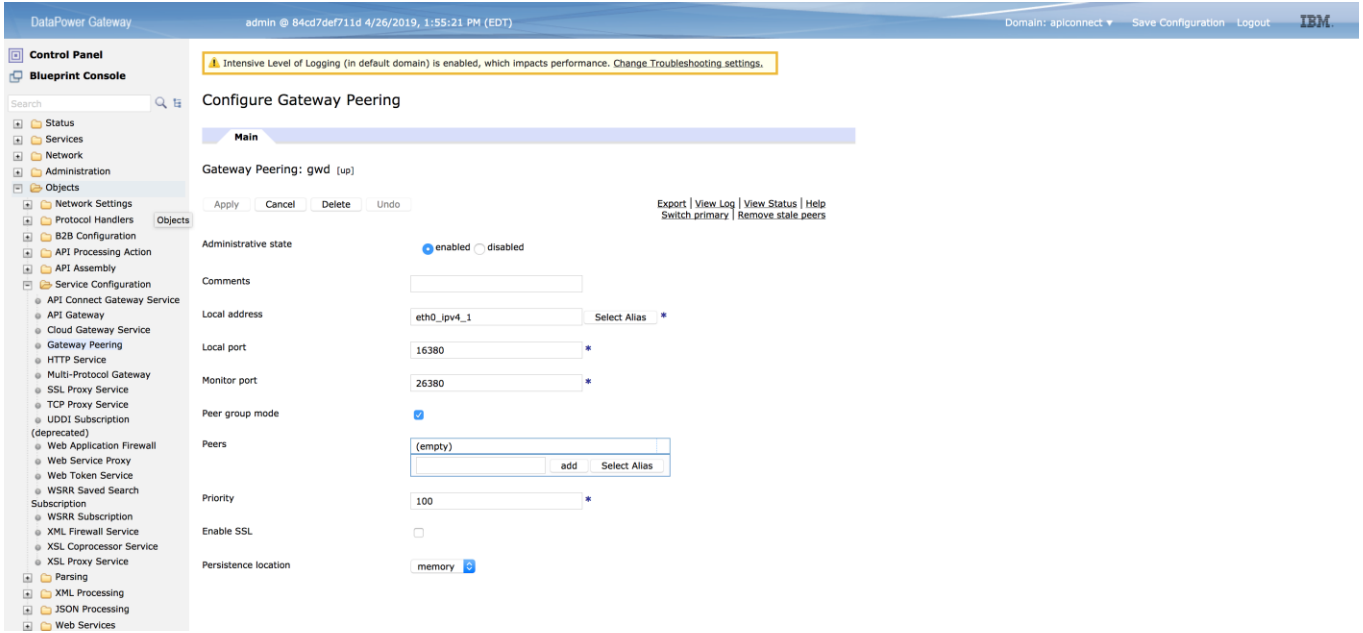
Administrative state	● enabled ○ disabled
Comments	
Local address	eth0_ipv4_1 *
Local port	3000 *
SSL client	gwd_client *
SSL server	gwd_server *
API gateway address	0.0.0.0
API gateway port	9443
Gateway Peering	gwd *
V5 compatibility mode	□
User-defined policies	(empty)

Gateway Configuration

– Gateway Peering Object

- Used for setting up the persistence layer, used by the Gateway Service Object to store all the data, API configurations, and used for linking individual Gateway Directors into a cluster.

```
gateway-peering gwd
  admin-state enabled
  local-address eth0_ipv4_1
  local-port 16380
  monitor-port 26380
  enable-peer-group on
  priority 100
  enable-ssl off
exit
```



⚠ Intensive Level of Logging (in default domain) is enabled, which impacts performance. Change Troubleshooting settings.

Configure Gateway Peering

Gateway Peering: gwd [up]

Administrative state ● enabled ○ disabled

Comments

Local address eth0_ipv4_1 Select Alias *

Local port 16380 *

Monitor port 26380 *

Peer group mode ☑

Peers (empty)

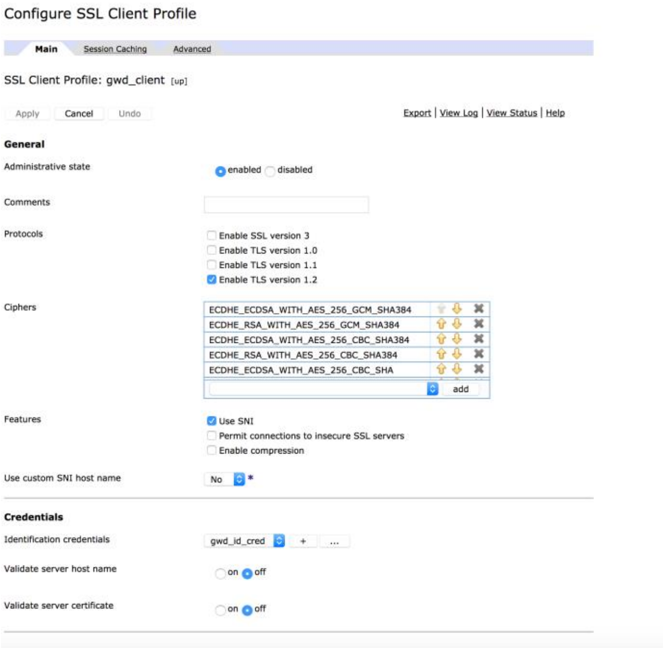
Priority 100 *

Enable SSL □

Persistence location memory

Gateway Configuration

- Profiles used for:
  - Communication to APIm
  - Communication with Gateway Peering



Configure SSL Client Profile

SSL Client Profile: gwd_client [up]

General

Administrative state: ● enabled ○ disabled

Comments:

Protocols:
□ Enable SSL version 3
□ Enable TLS version 1.0
□ Enable TLS version 1.1
☑ Enable TLS version 1.2

Ciphers:
ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE_ECDSA_WITH_AES_256_CBC_SHA

Features:
☑ Use SNI
□ Permit connections to insecure SSL servers
□ Enable compression

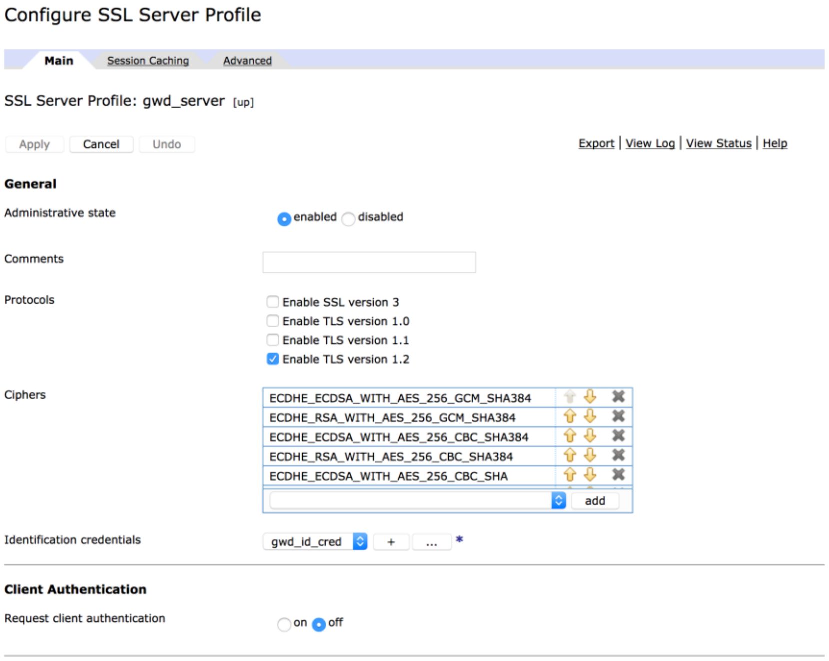
Use custom SNI host name: No *

Credentials

Identification credentials: gwd_id_cred + ...

Validate server host name: ○ on ● off

Validate server certificate: ○ on ● off



Configure SSL Server Profile

SSL Server Profile: gwd_server [up]

General

Administrative state: ● enabled ○ disabled

Comments:

Protocols:
□ Enable SSL version 3
□ Enable TLS version 1.0
□ Enable TLS version 1.1
☑ Enable TLS version 1.2

Ciphers:
ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE_ECDSA_WITH_AES_256_CBC_SHA

Identification credentials: gwd_id_cred + ... *

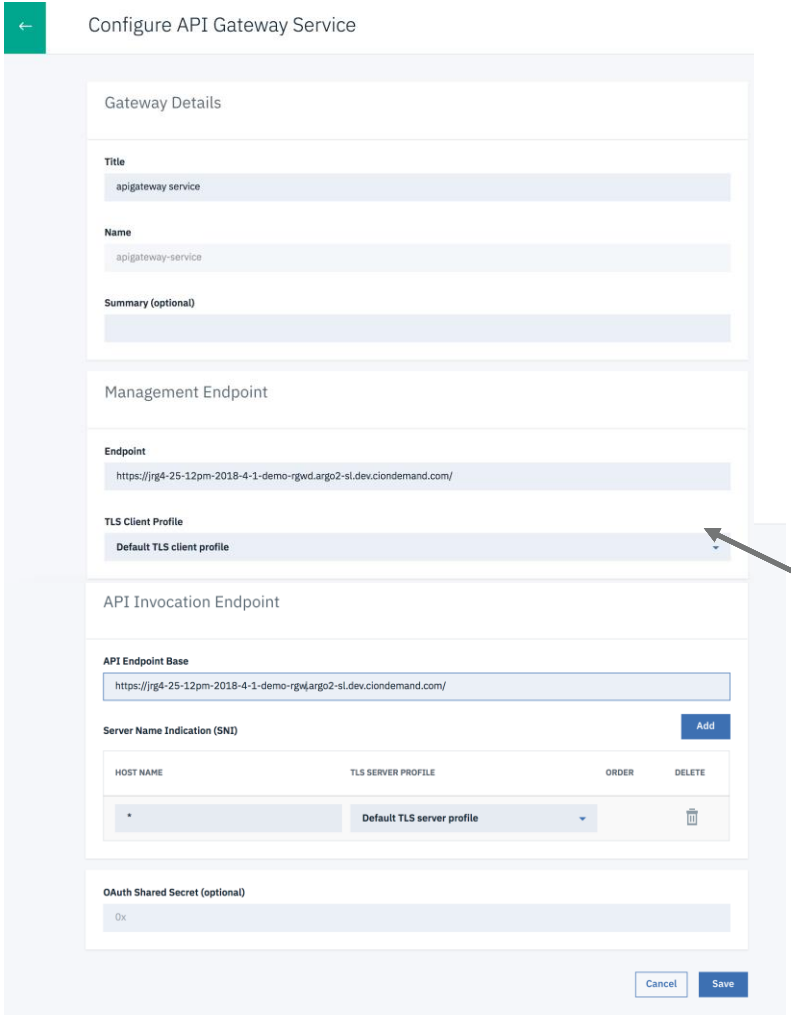
Client Authentication

Request client authentication: ○ on ● off

APIm Registration

– The Gateway Service Object is the interface between the Management Server (Cloud Manager and API Manager) and a gateway service.

– It handles the initial registration and configuration of a gateway service with the Management Server.



Configure API Gateway Service

Gateway Details

Title: apigateway service

Name: apigateway-service

Summary (optional):

Management Endpoint

Endpoint: https://jrg4-25-12pm-2018-4-1-demo-rgwd.argo2-sl.dev.ciondemand.com/

TLS Client Profile: Default TLS client profile

API Invocation Endpoint

API Endpoint Base: https://jrg4-25-12pm-2018-4-1-demo-rgw.argo2-sl.dev.ciondemand.com/

Server Name Indication (SNI)

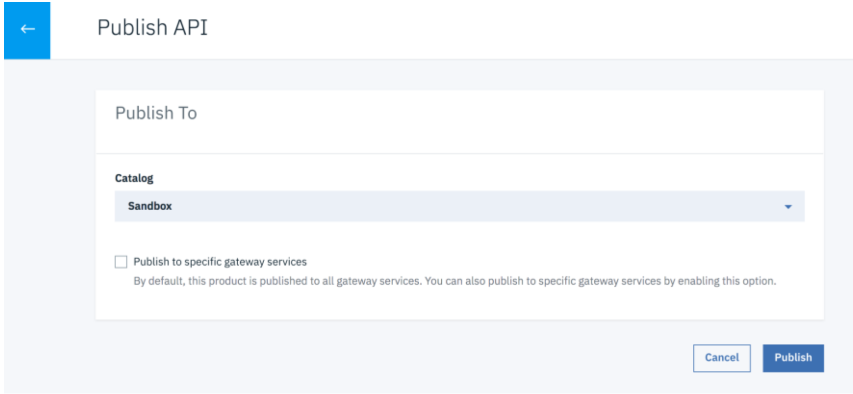
HOST NAME	TLS SERVER PROFILE	ORDER	DELETE
*	Default TLS server profile		🗑

OAuth Shared Secret (optional): 0x

REST call sent to the Gateway Service Object. Response from Gateway Service Object informs management server which policies it supports and which events the gateway service wants to be sent.

Data Transmission

- Handles all catalog updates coming from the API Manager, aggregates them, and refreshes them to the gateways.
- It seeks to keep the gateways in sync with the Management Server and maintain consistency. It can take actions to restore consistency if it detects that the gateways are out of sync.



Publish API

Publish To

Catalog

Sandbox

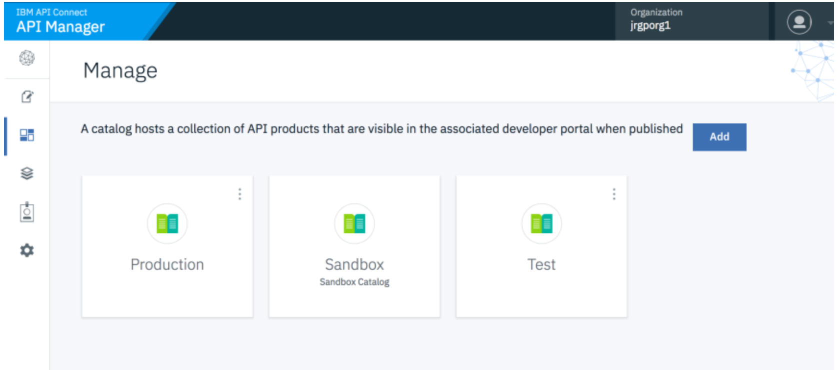
□ Publish to specific gateway services

By default, this product is published to all gateway services. You can also publish to specific gateway services by enabling this option.

Example: REST call sent to the Gateway Service Object to Publish product.

Data Transmission - What Types of Data?

- Events sent from the Management Server to the Gateway Service Object are referred to as “webhooks”. Webhook events are sent via REST calls.
- There are two primary types of webhooks - Cloud and Catalog
  - Cloud Webhooks are events that impact the operation of the gateway for all catalogs associated with it.

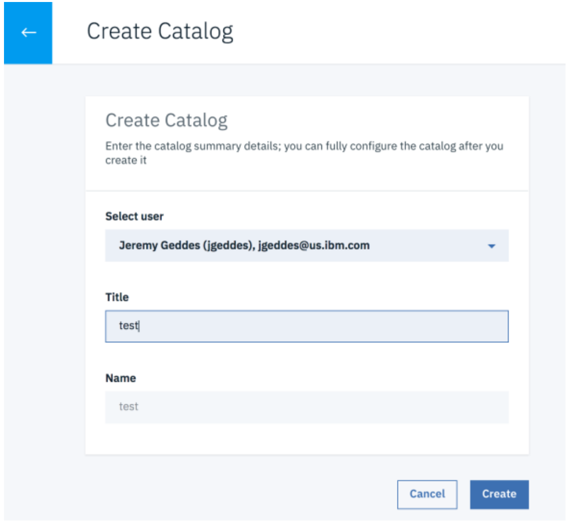


A catalog hosts a collection of API products that are visible in the associated developer portal when published

- These would typically be triggered by user actions taken in the Cloud Manager.
- Examples include “gateway_service_updates” which includes information on configuration analytics, or configuring SNI for the gateway.

Catalog Data

- Catalog Webhooks are events that apply to a particular catalog deployed on the gateway.
  - These would typically be triggered by user actions taken in the API Manager.
    - Examples include webhooks to create catalogs, publish/update/remove products, configure apis, manage subscriptions and applications, among others.



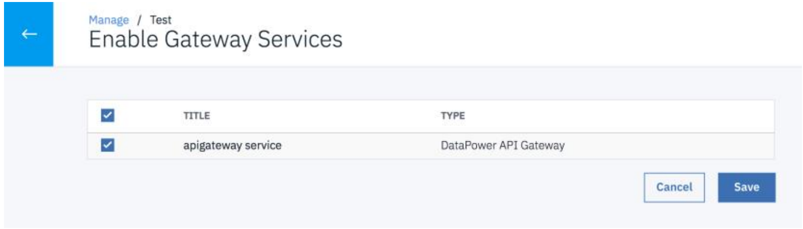
Create Catalog

Enter the catalog summary details; you can fully configure the catalog after you create it

Select user: Jeremy Geddes (jgeddes), jgeddes@us.ibm.com

Title: test

Name: test



Enable Gateway Services

☑	TITLE	TYPE
☑	apigateway service	DataPower API Gateway

Protocol changes v5 to v2018

Event based messages from API Manager to the Gateway Service Object.

- Results in a less chatty protocol and smaller payloads.
- Individual events, such as a new subscription, are sent via webhooks rather than having to pull the full model on each update.

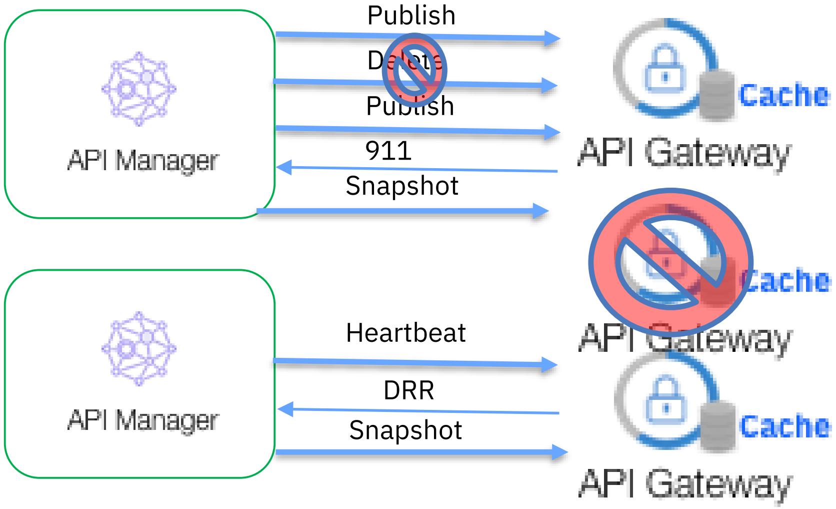
Consistency

- Mechanisms in place to ensure consistency between the gateway service and the Management server.
- A recovery scenario can re-sync the gateways if an inconsistency is detected.

Consistency & Recovery

Consistency problem: 911

HA not maintained: DRR



- Guarantee processing in order before making changes.
- Information stored in gateway, about gateway service and the published catalogs.
- The data is shared between all gateways in the cluster.

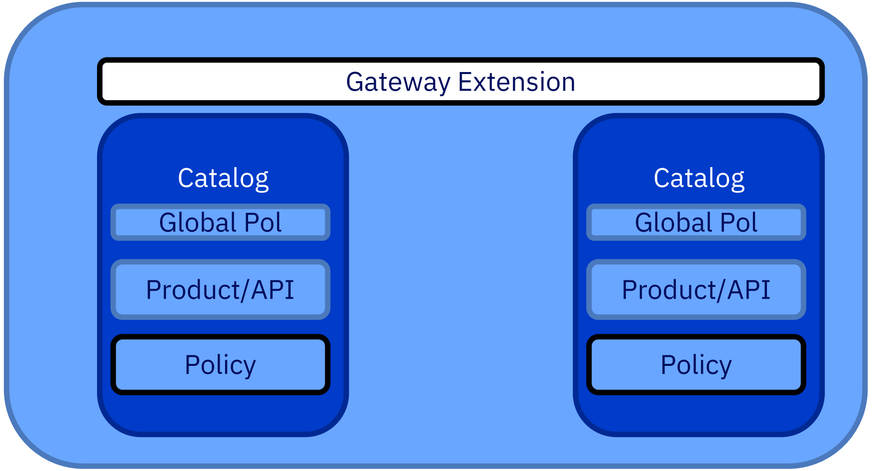
Error Handling and Recovery

- Error handling and recovery is the built in to the Gateway Service Object and is triggered when it detects that it is out of sync with the Management Server.
- The Gateway Service Object will initiate the recovery scenario by sending a REST call back to the Management Server if it encounters the following situations.
  - The Gateway Service Object is unable to find the previous_id for a webhook it is attempting to process.
  - The Gateway Service Object experiences an error while processing a webhook event or while refreshing the gateway.
- In response, the Management Server sends a “snapshot” webhook, which contains all the information needed to get the gateway and the Management Server back in sync.
- “Snapshot” webhooks can be sent at the gateway service level or at the individual catalog level. During this time, the gateway service will continue to operate with its current configuration to prevent downtime.

Extending Gateway Services

- Gateway Extensions
- User Defined Policies
- Global Policies

Gateway Extensions



Scope

- Gateway Service

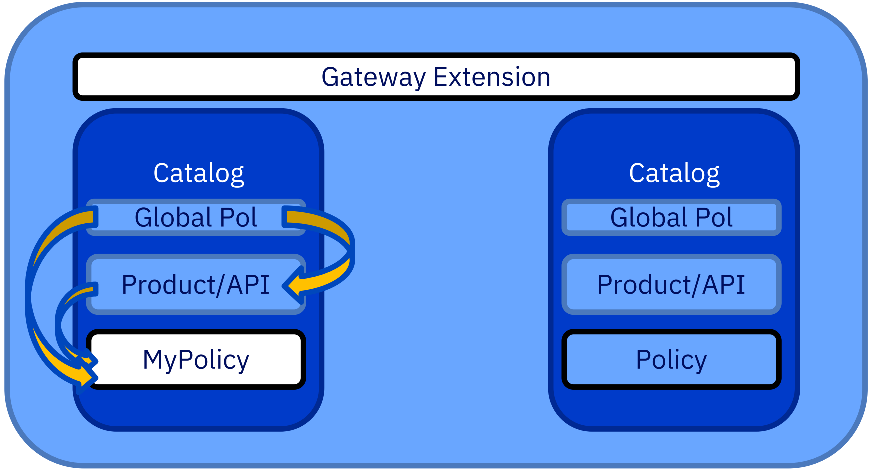
Typical Uses

- Function not exposed by APIm but available on Gateway

Narrative

- Customer wants to customize the gateway to enable gateway function.

User Defined Policies



Scope

- Catalog – V5C Only
- Gateway Service – API Gateway Only

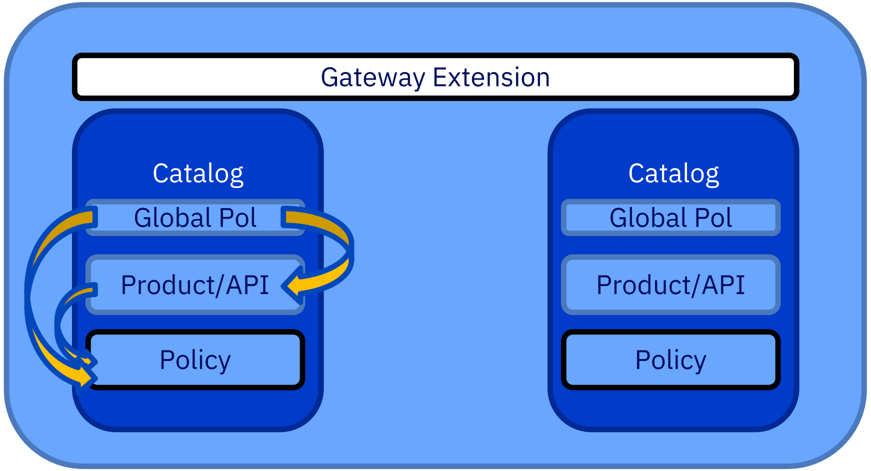
Typical Uses

- Customer specific policy requirement

Narrative

- Customer wants to package up code to be reused on multiple APIs

Global Policies



Scope

- Catalog

Typical Uses

- Normalize use of policies across entire catalog

Narratives

- Customer wants an administrator to define security policy for all APIs on a catalog
- Customer has a set of policies to be applied to every API published to a particular catalog.

Security - Architecture Deep Dive

- Agenda

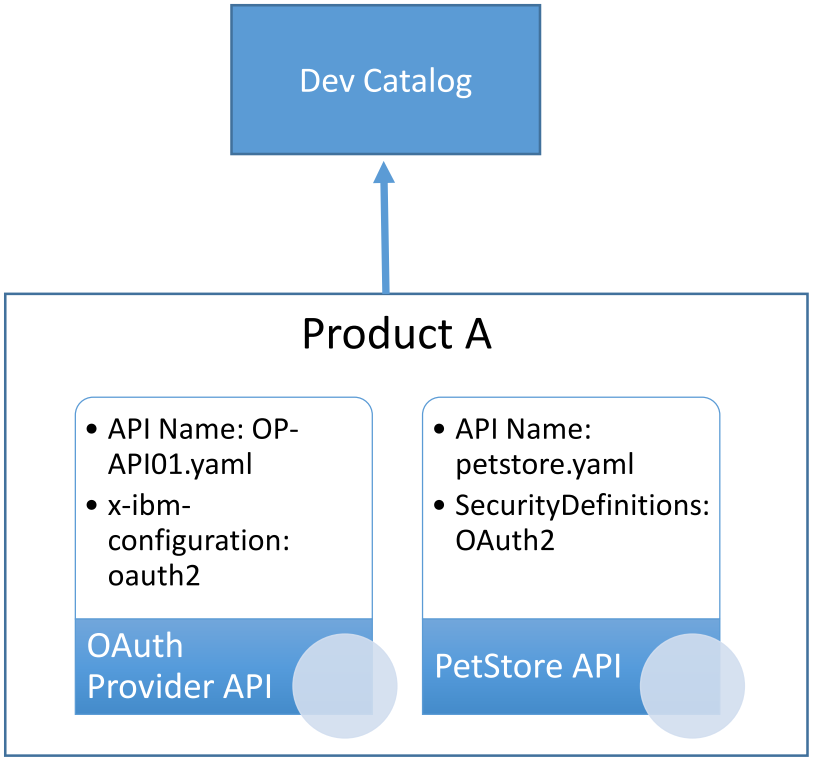
  - Re-architected OAuth Provider
    - First Class Resource Object
    - Native and Third Party Provider
    - Customizable Assembly
    - Out of the box JWT Grant Type Support
    - Out of the box OIDC Support
    - Advanced OAuth Token Management
    - Per Provider Crypto Key Management

  - New Policies
    - User Security
    - Client Security
    - API Rate Limit
    - GraphQL (future)

  - Enhanced Application Authentication Support

Re-Architected OAuth Provider - User Experience

V5



Dev Catalog

Product A

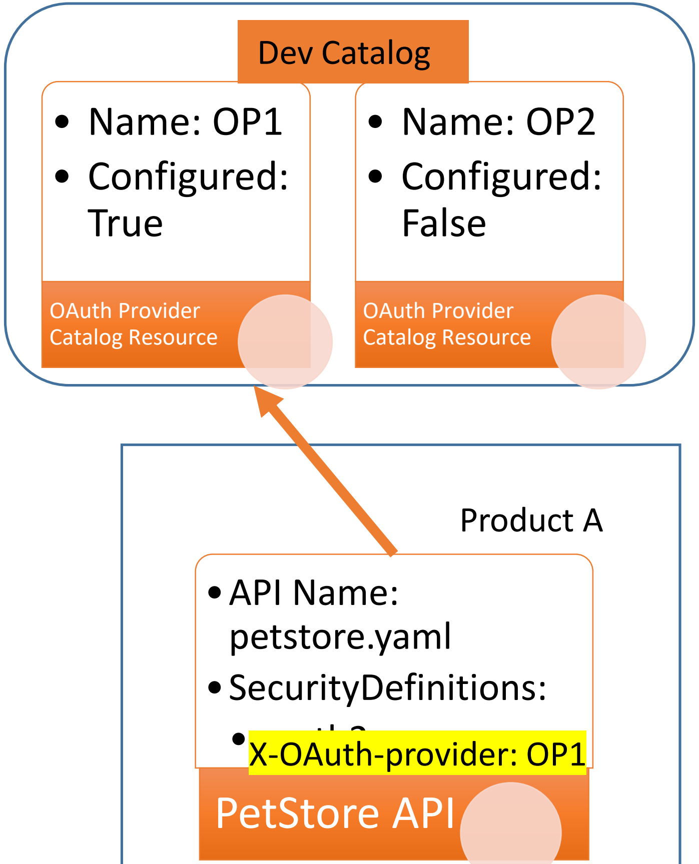
- API Name: OP-API01.yaml
- x-ibm-configuration: oauth2

OAuth Provider API

- API Name: petstore.yaml
- SecurityDefinitions: OAuth2

PetStore API

v2018



Dev Catalog

- Name: OP1
- Configured: True

OAuth Provider Catalog Resource

- Name: OP2
- Configured: False

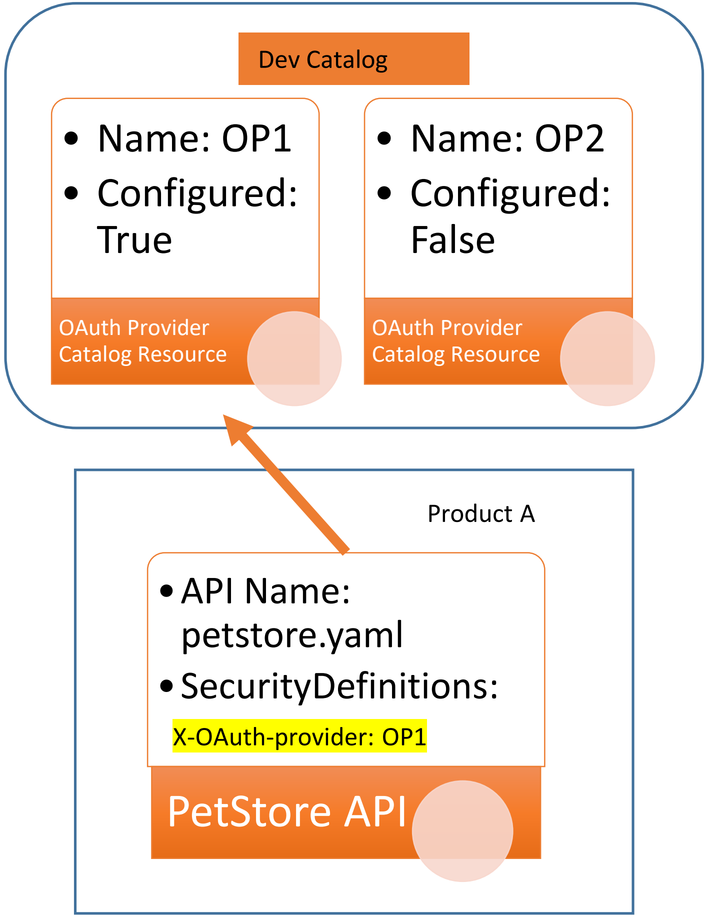
OAuth Provider Catalog Resource

Product A

- API Name: petstore.yaml
- SecurityDefinitions:
  - oauth2
  - X-OAuth-provider: OP1

PetStore API

Re-Architected OAuth Provider - User Experience



Dev Catalog

- Name: OP1
- Configured: True

OAuth Provider Catalog Resource

- Name: OP2
- Configured: False

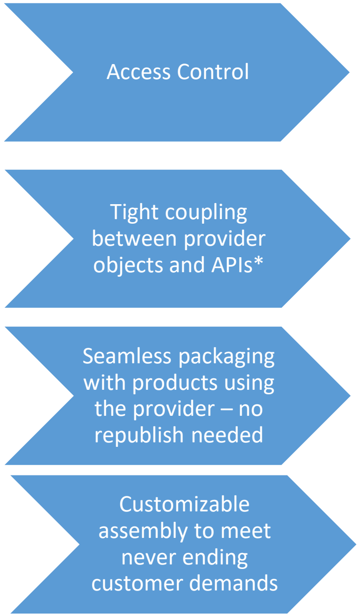
OAuth Provider Catalog Resource

Product A

- API Name: petstore.yaml
- SecurityDefinitions:

X-OAuth-provider: OP1

PetStore API



Access Control

Tight coupling between provider objects and APIs*

Seamless packaging with products using the provider – no republish needed

Customizable assembly to meet never ending customer demands

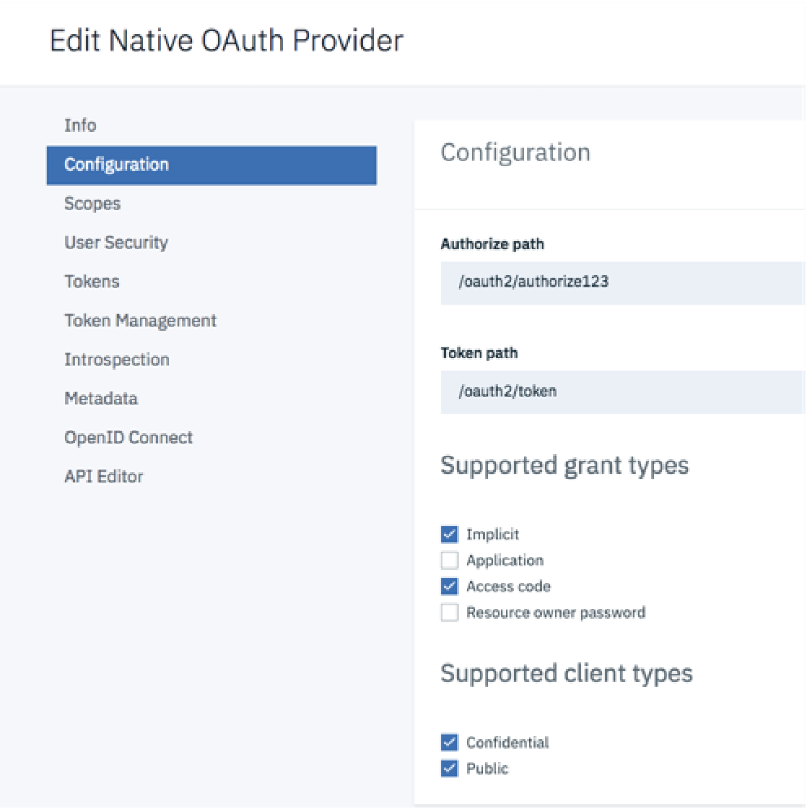
* Note: APIs using OAuth Provider must be published in order for the OAuth endpoint to be accessible

Feature list of OAuth in APIC V5, v2018+V5GW, v2018+APIGW

Features	V4	V5	v2018 + V5 CompatGW	v2018 + APIGW
Basic OAuth Support	☑	☑	☑	☑
Distinct Client ids and Secrets	×	☑	☑	☑
Separate API	×	☑	☑	☑
Access Control	×	×	☑	☑
Seamless packaging within product	☑	×	☑	☑
Tight coupling with Provider	×	×	☑ *	☑
Customize OAuth Assembly	×	×	×	☑
Dynamic configuration updates	×	×	×	☑
Context variable driven	×	×	×	☑
Independent Resource Owner Security	×	×	×	☑
Out of the box OIDC support	×	×	×	☑
Out of the box JWT Grant Type support	×	×	×	☑

* - Tight coupling is only at the APIManager API level, not in the backend V5 Gateway

Native & Third Party OAuth Providers



Edit Native OAuth Provider

Configuration

Authorize path

/oauth2/authorize123

Token path

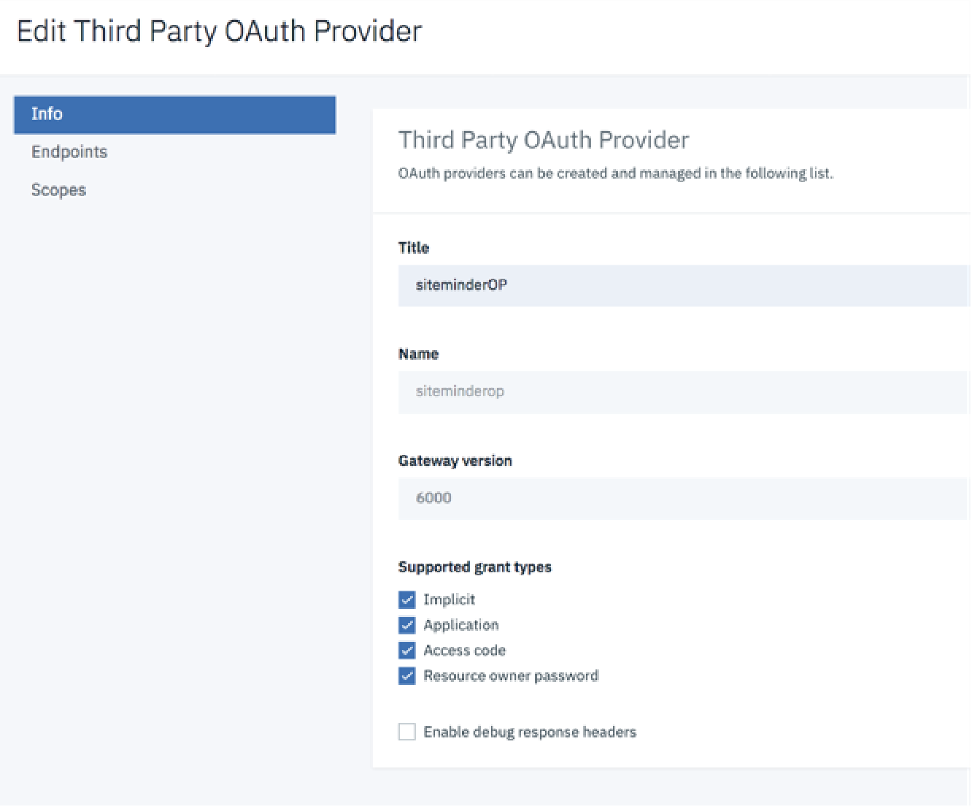
/oauth2/token

Supported grant types

☑ Implicit
□ Application
☑ Access code
□ Resource owner password

Supported client types

☑ Confidential
☑ Public



Edit Third Party OAuth Provider

Third Party OAuth Provider

OAuth providers can be created and managed in the following list.

Title

siteminderOP

Name

siteminderop

Gateway version

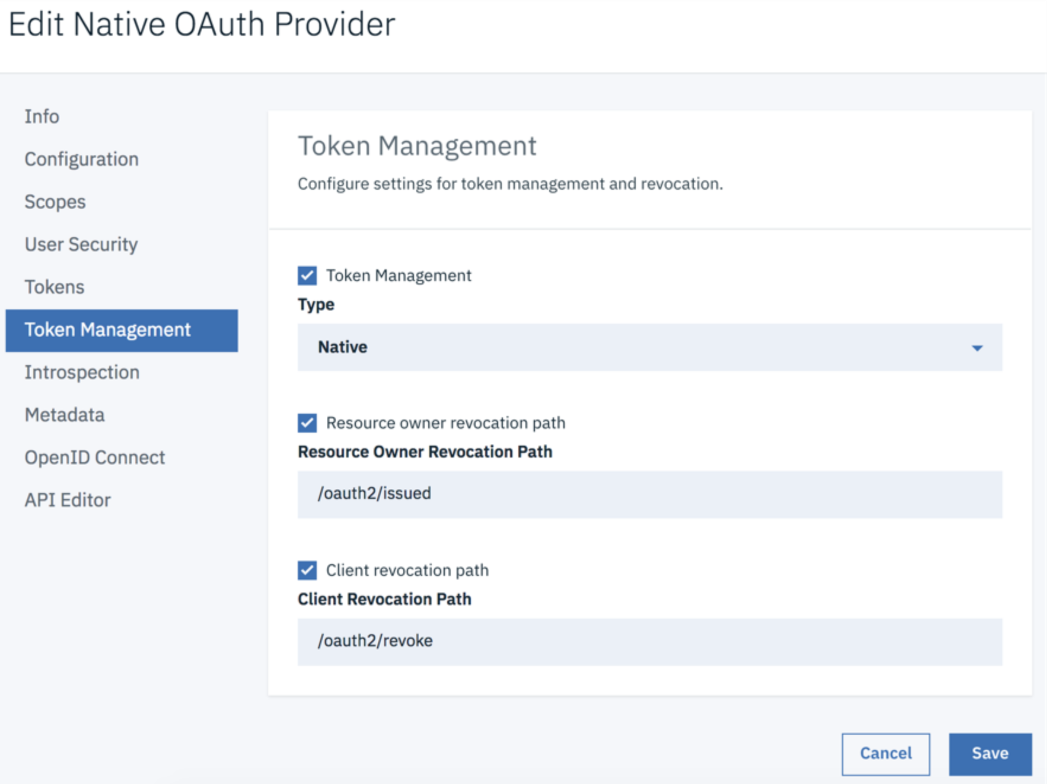
6000

Supported grant types

☑ Implicit
☑ Application
☑ Access code
☑ Resource owner password

□ Enable debug response headers

Advanced OAuth Token Management



Edit Native OAuth Provider

Token Management

Configure settings for token management and revocation.

☑ Token Management

Type

Native

☑ Resource owner revocation path

Resource Owner Revocation Path

/oauth2/issued

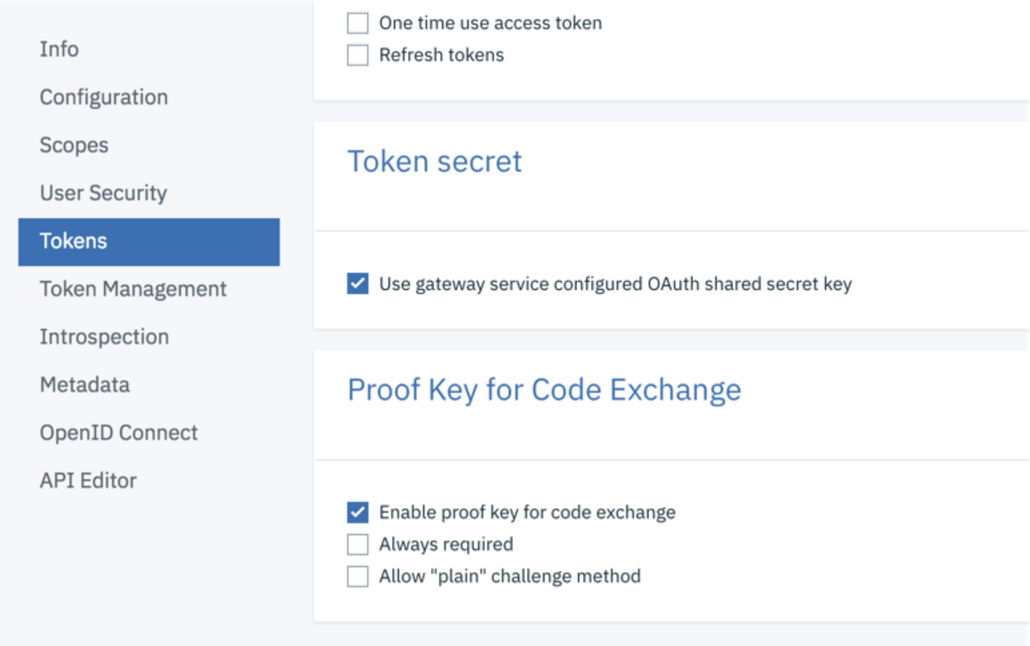
☑ Client revocation path

Client Revocation Path

/oauth2/revoke

- Quota Enforcement is no longer used for Token Management
- Separate database instance exclusive to Security Token Management
- Tokens are whitelisted for increased security

Per Provider Crypto Key Management & PKCE Support



□ One time use access token
□ Refresh tokens

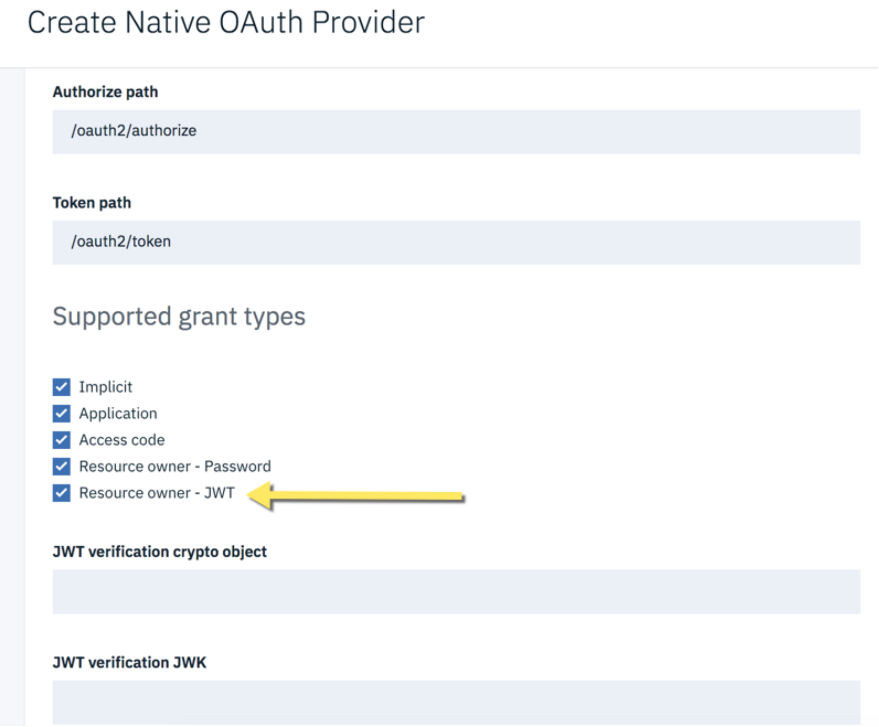
Token secret

☑ Use gateway service configured OAuth shared secret key

Proof Key for Code Exchange

☑ Enable proof key for code exchange
□ Always required
□ Allow "plain" challenge method

Out of the box JWT Grant Type Support



Create Native OAuth Provider

Authorize path

/oauth2/authorize

Token path

/oauth2/token

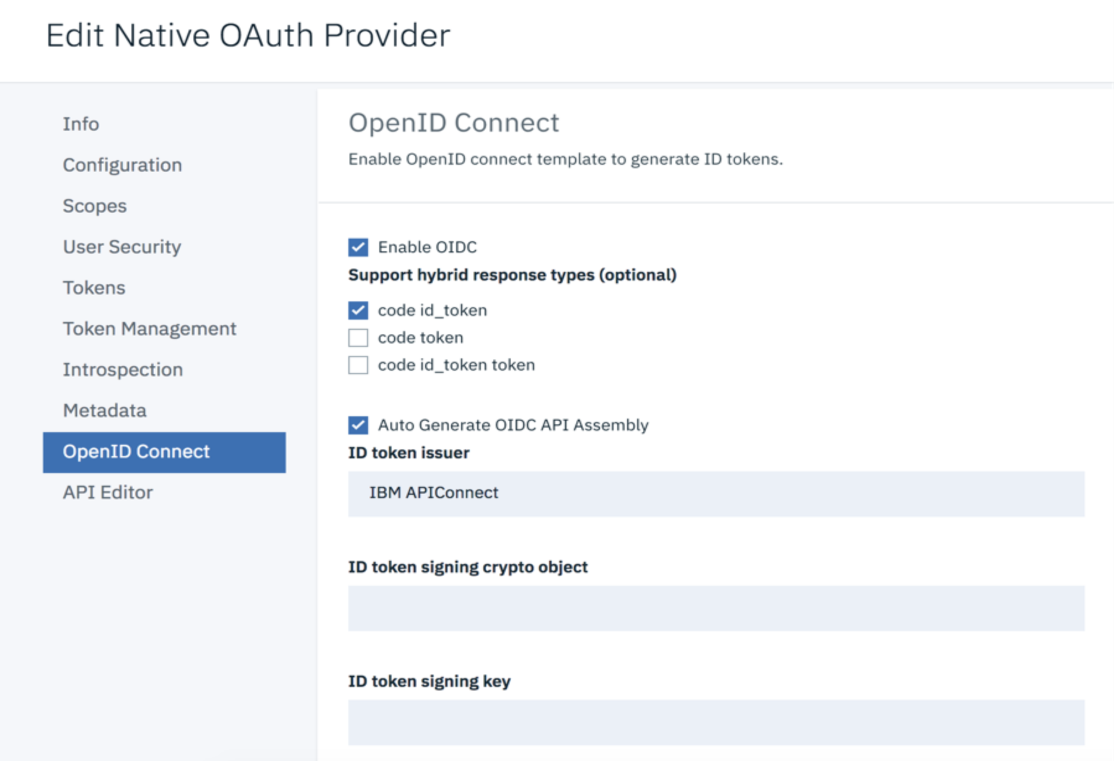
Supported grant types

☑ Implicit
☑ Application
☑ Access code
☑ Resource owner - Password
☑ Resource owner - JWT

JWT verification crypto object

JWT verification JWK

Out of the box OIDC Support



Edit Native OAuth Provider

OpenID Connect

Enable OpenID connect template to generate ID tokens.

☑ Enable OIDC

Support hybrid response types (optional)

☑ code id_token

□ code token

□ code id_token token

☑ Auto Generate OIDC API Assembly

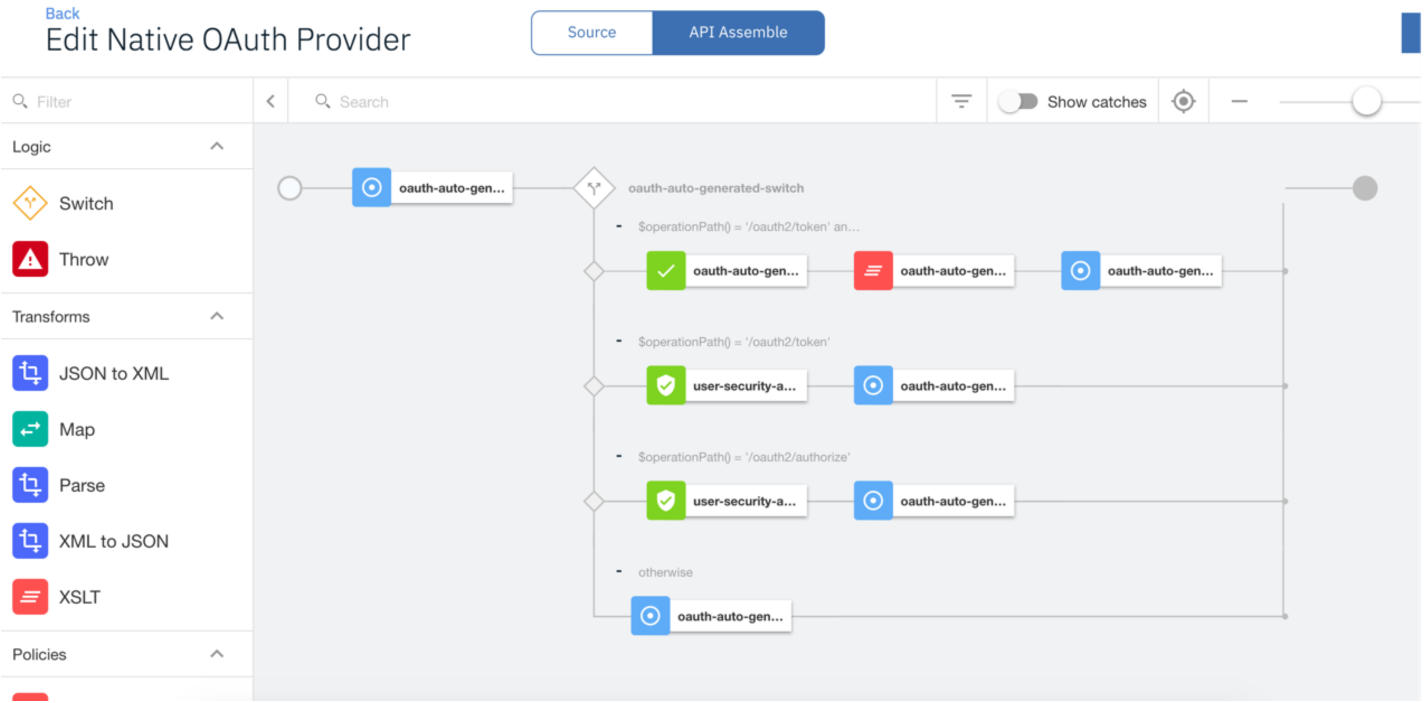
ID token issuer

IBM APIConnect

ID token signing crypto object

ID token signing key

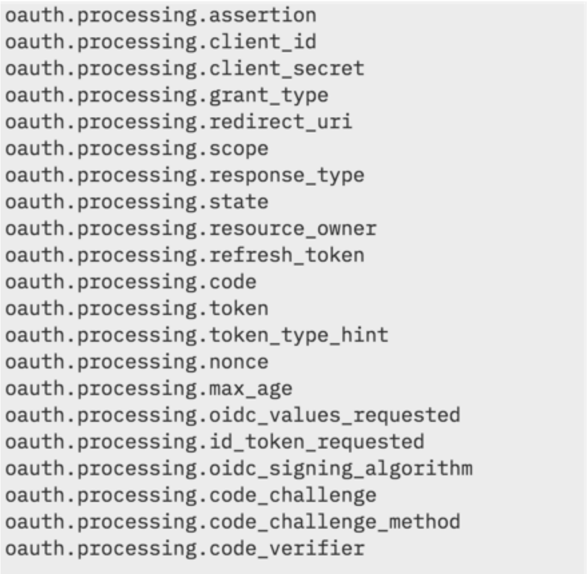
Auto Generated OAuth API & Assembly



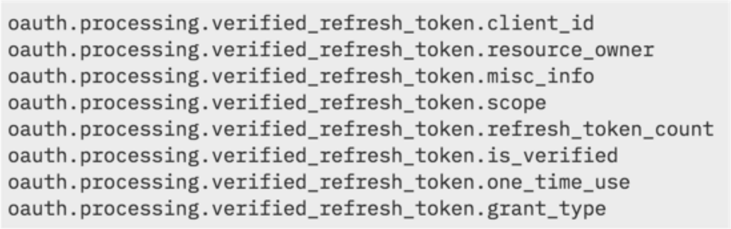
Back

Edit Native OAuth Provider

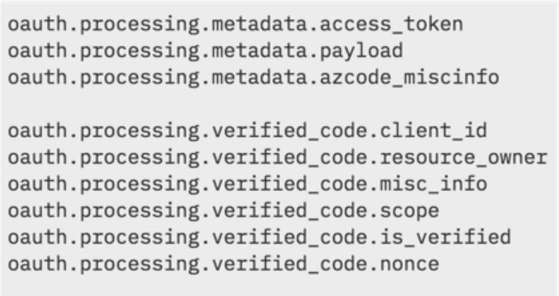
Assembly Fully Driven by Context Variables – Highly Customizable



```
oauth.processing.assertion
oauth.processing.client_id
oauth.processing.client_secret
oauth.processing.grant_type
oauth.processing.redirect_uri
oauth.processing.scope
oauth.processing.response_type
oauth.processing.state
oauth.processing.resource_owner
oauth.processing.refresh_token
oauth.processing.code
oauth.processing.token
oauth.processing.token_type_hint
oauth.processing.nonce
oauth.processing.max_age
oauth.processing.oidc_values_requested
oauth.processing.id_token_requested
oauth.processing.oidc_signing_algorithm
oauth.processing.code_challenge
oauth.processing.code_challenge_method
oauth.processing.code_verifier
```

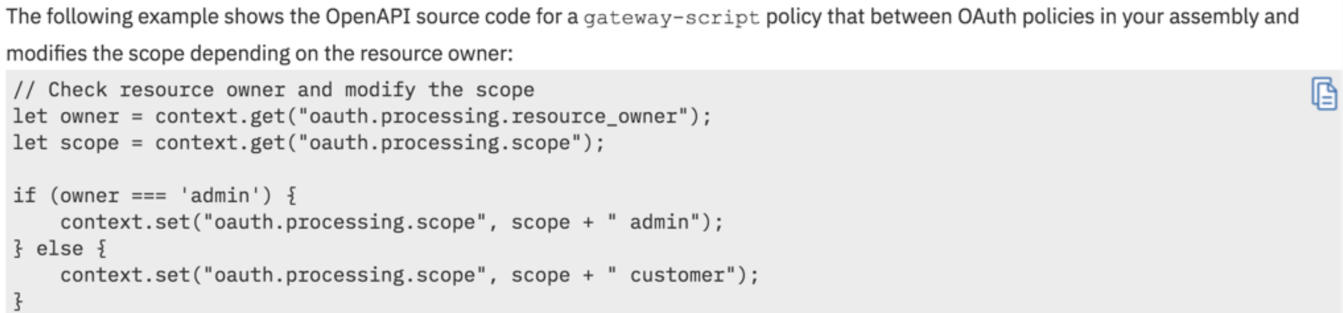


```
oauth.processing.verified_refresh_token.client_id
oauth.processing.verified_refresh_token.resource_owner
oauth.processing.verified_refresh_token.misc_info
oauth.processing.verified_refresh_token.scope
oauth.processing.verified_refresh_token.refresh_token_count
oauth.processing.verified_refresh_token.is_verified
oauth.processing.verified_refresh_token.one_time_use
oauth.processing.verified_refresh_token.grant_type
```



```
oauth.processing.metadata.access_token
oauth.processing.metadata.payload
oauth.processing.metadata.azcode_miscinfo

oauth.processing.verified_code.client_id
oauth.processing.verified_code.resource_owner
oauth.processing.verified_code.misc_info
oauth.processing.verified_code.scope
oauth.processing.verified_code.is_verified
oauth.processing.verified_code.nonce
```

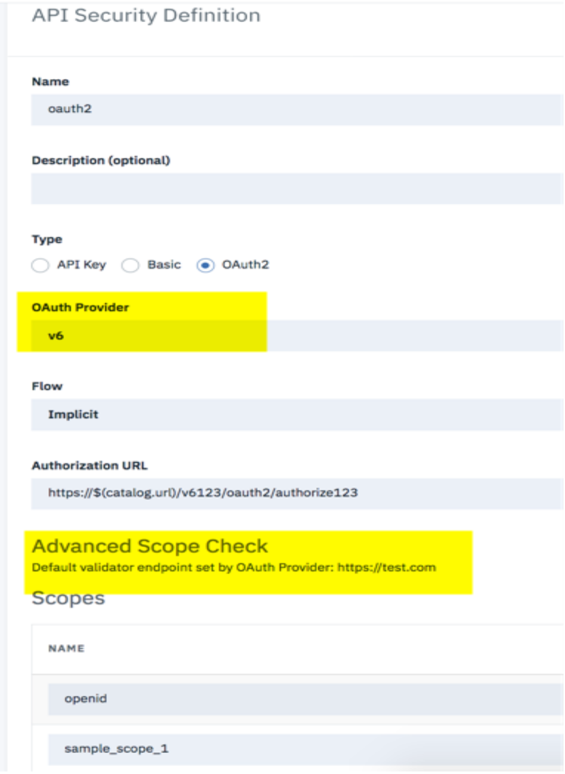


The following example shows the OpenAPI source code for a `gateway-script` policy that between OAuth policies in your assembly and modifies the scope depending on the resource owner:

```
// Check resource owner and modify the scope
let owner = context.get("oauth.processing.resource_owner");
let scope = context.get("oauth.processing.scope");

if (owner === 'admin') {
    context.set("oauth.processing.scope", scope + " admin");
} else {
    context.set("oauth.processing.scope", scope + " customer");
}
```

Link Provider to OAuth Security Definitions



API Security Definition

Name

oauth2

Description (optional)

Type

○ API Key ○ Basic ● OAuth2

OAuth Provider

v6

Flow

Implicit

Authorization URL

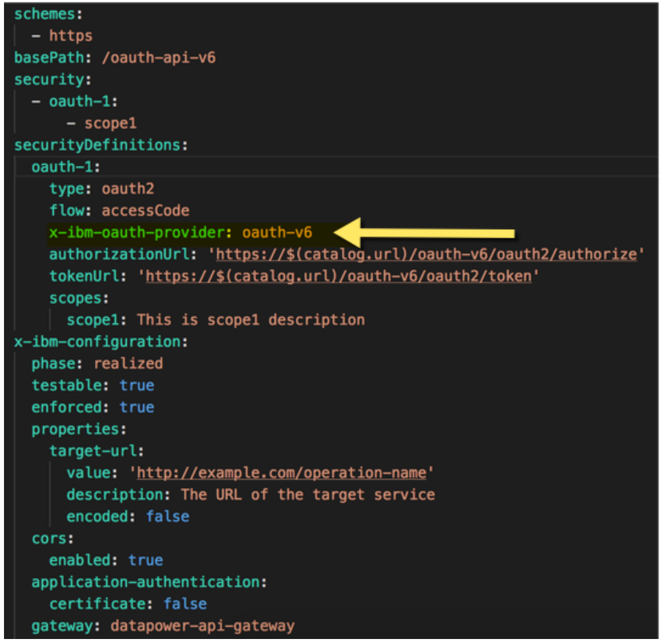
https://$(catalog.url)/v6123/oauth2/authorize123

Advanced Scope Check

Default validator endpoint set by OAuth Provider: https://test.com

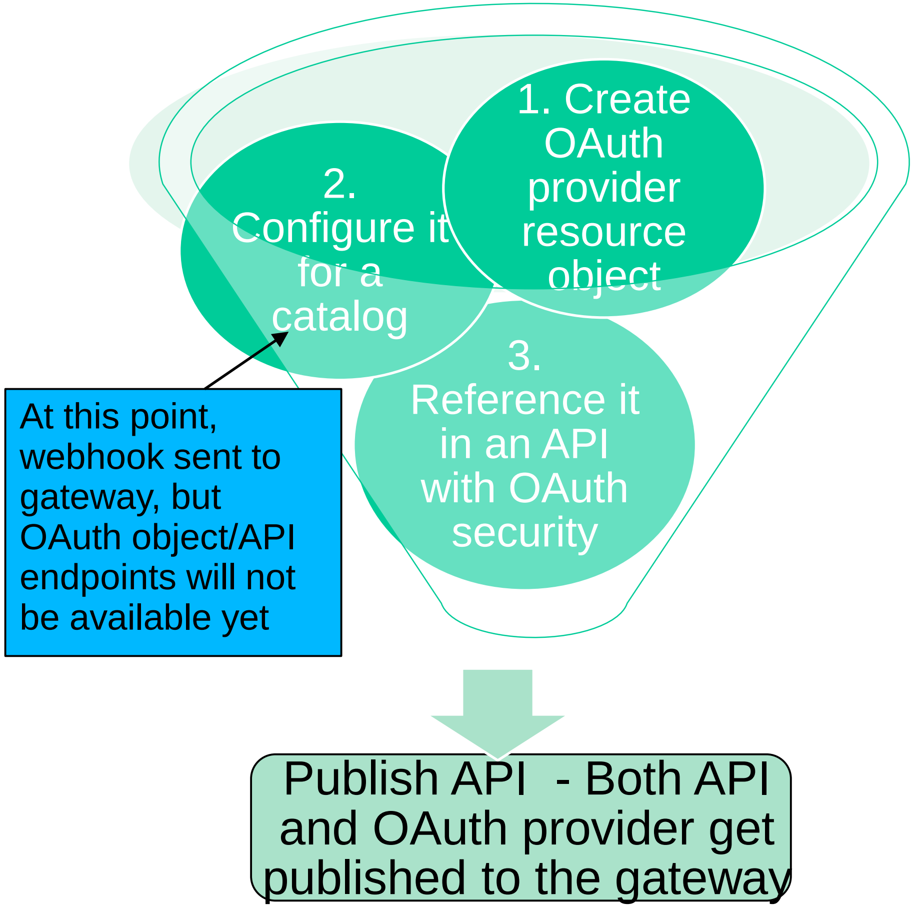
Scopes

NAME
openid
sample_scope_1



```
schemes:
  - https
basePath: /oauth-api-v6
security:
  - oauth-1:
      - scope1
securityDefinitions:
  oauth-1:
    type: oauth2
    flow: accessCode
    x-ibm-oauth-provider: oauth-v6
    authorizationUrl: 'https://$(catalog.url)/oauth-v6/oauth2/authorize'
    tokenUrl: 'https://$(catalog.url)/oauth-v6/oauth2/token'
    scopes:
      scope1: This is scope1 description
x-ibm-configuration:
  phase: realized
  testable: true
  enforced: true
  properties:
    target-url:
      value: 'http://example.com/operation-name'
      description: The URL of the target service
      encoded: false
  cors:
    enabled: true
  application-authentication:
    certificate: false
  gateway: datapower-api-gateway
```

Publish sequence – onto Gateways

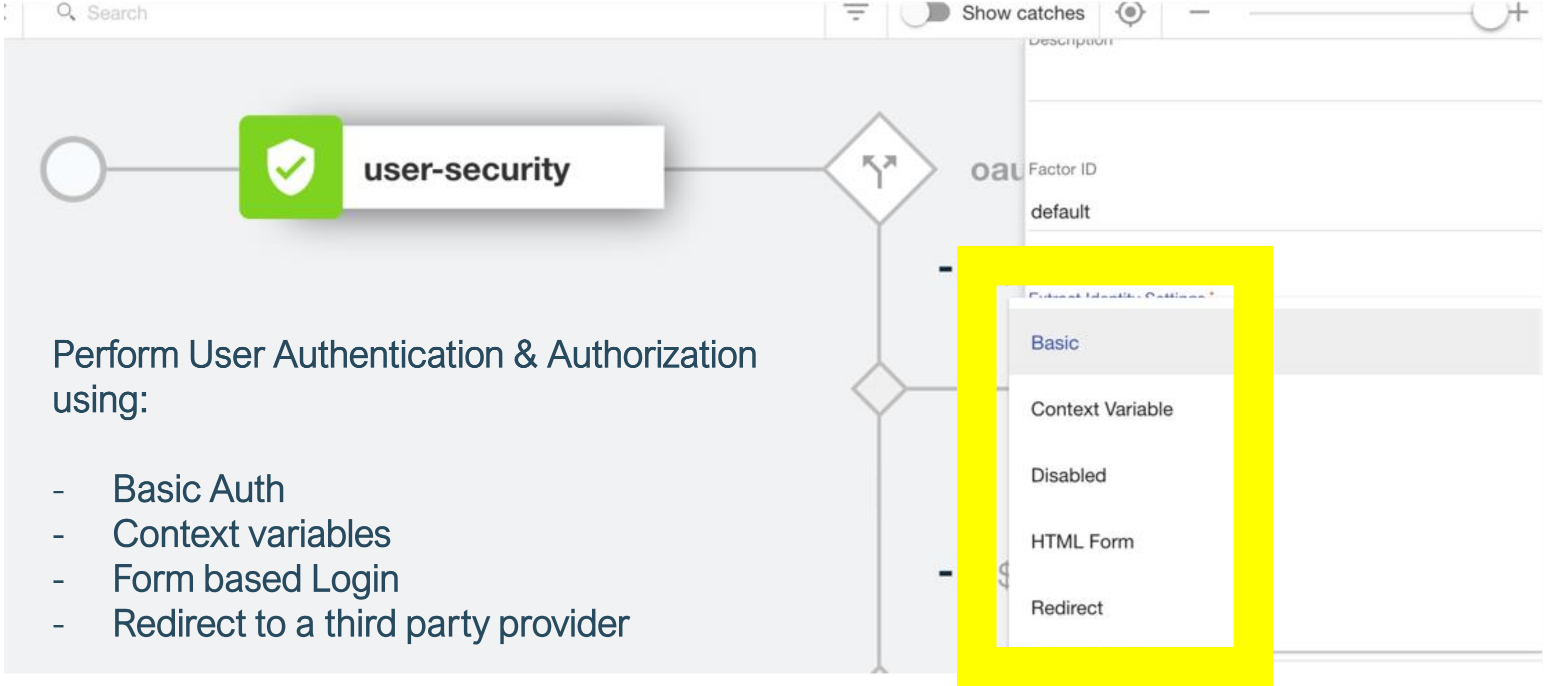

1. Create OAuth provider resource object
2. Configure it for a catalog
3. Reference it in an API with OAuth security

At this point, webhook sent to gateway, but OAuth object/API endpoints will not be available yet

Publish API - Both API and OAuth provider get published to the gateway

- Note : Unless the OAuth provider is used by at least one of the APIs in the Security Definition, the Oauth provider endpoints are not available in Gateway
- Any updates to OAuth provider or its underlying API will take effect immediately on the gateway (no need to republish once already configured in catalog and used by any API)
- Until the last API that uses an OAuth provider is published, the OAuth provider and its API also remain published in the gateway

New Policies – User Security

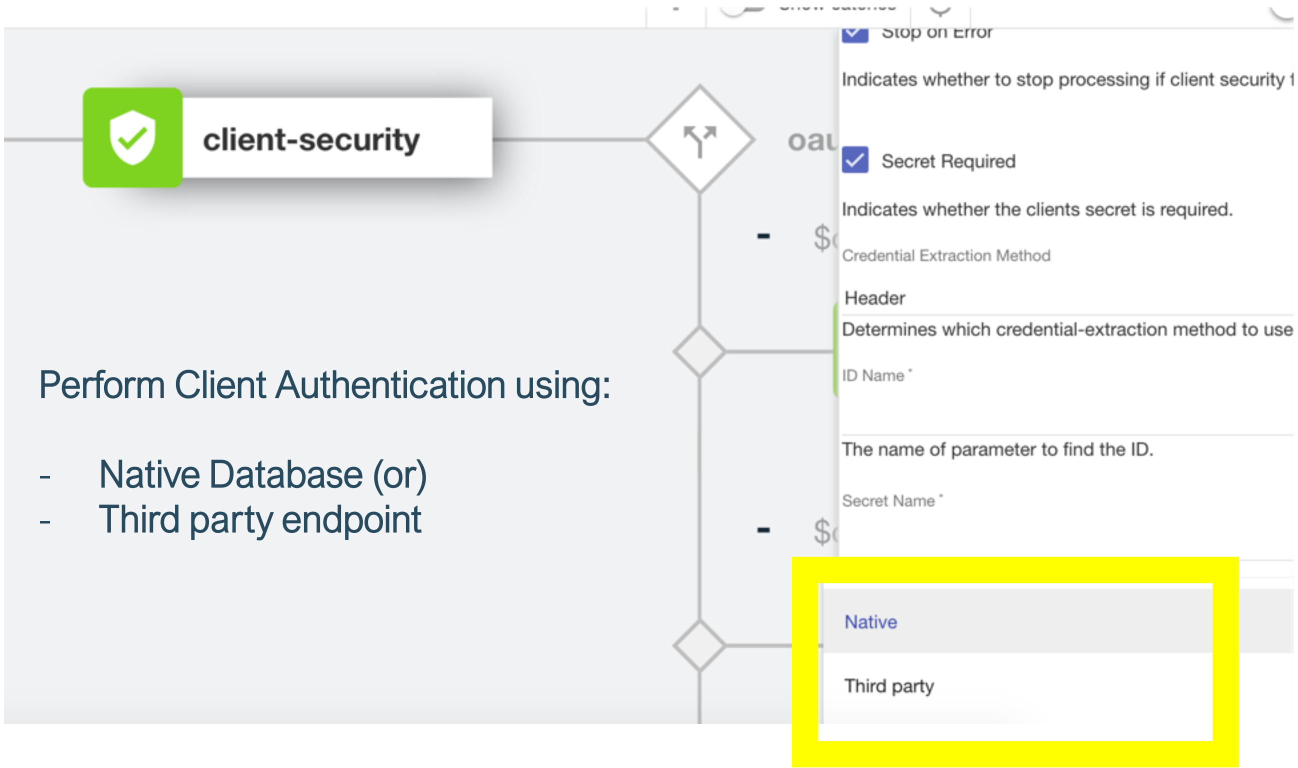


Factor ID
default

Perform User Authentication & Authorization using:

- Basic Auth
- Context variables
- Form based Login
- Redirect to a third party provider

New Policies – Client Security



Perform Client Authentication using:

- Native Database (or)
- Third party endpoint

☑ Stop on Error

Indicates whether to stop processing if client security f

☑ Secret Required

Indicates whether the clients secret is required.

Credential Extraction Method

Header

Determines which credential-extraction method to use

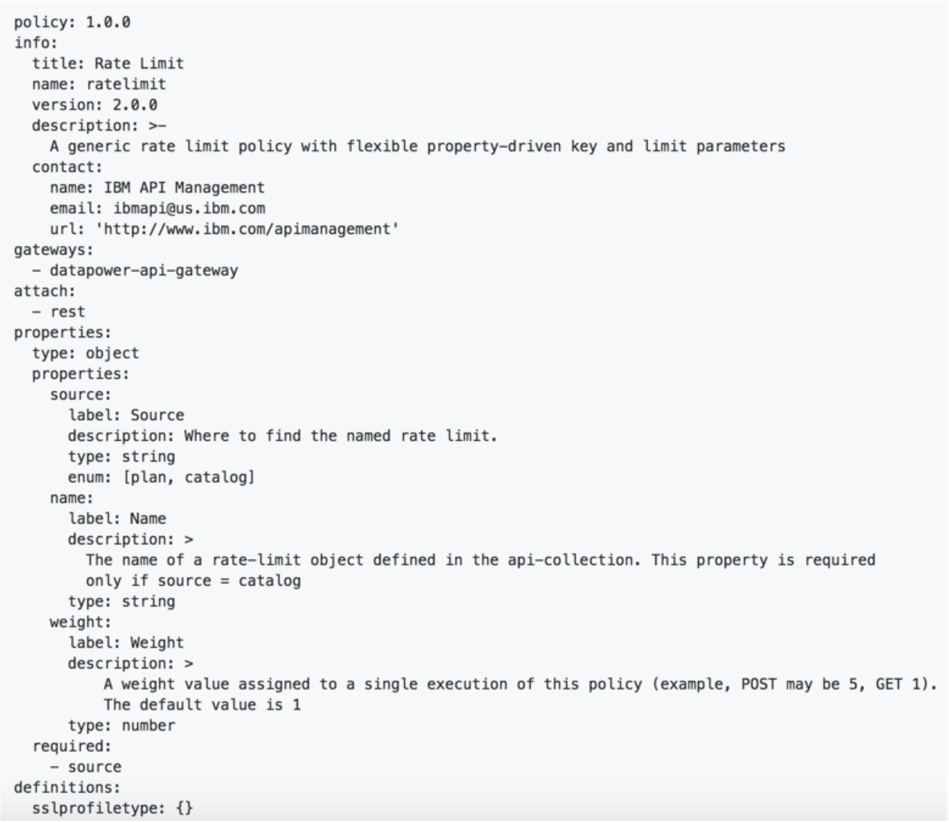
ID Name *

The name of parameter to find the ID.

Secret Name *

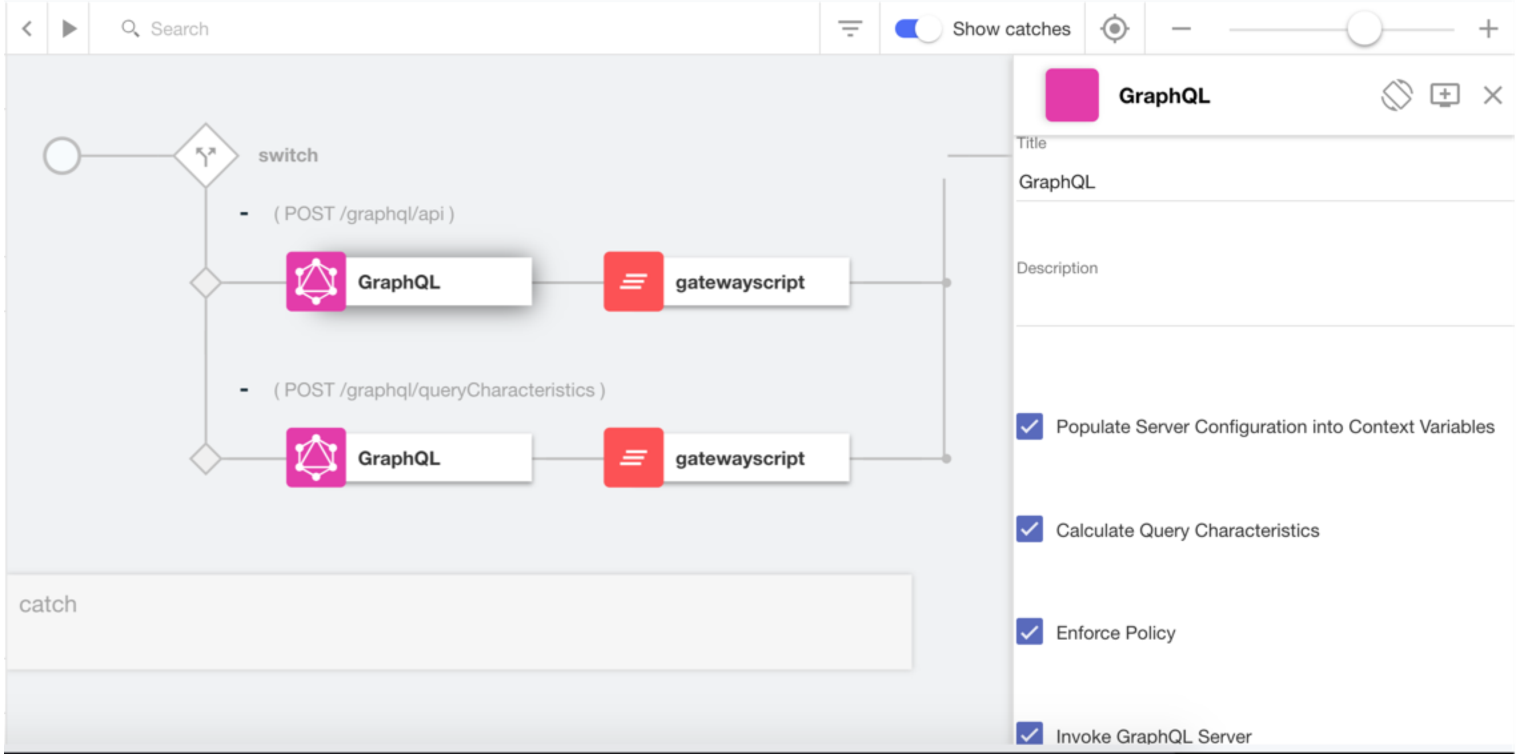
New Policies – API Rate Limit (preview)

- Configurable Rate limit keys
- Use as a circuit breaker to reject transactions exceeding a configurable threshold
- Rate limit based on the backend service limits rather than front end client limits



```
policy: 1.0.0
info:
  title: Rate Limit
  name: ratelimit
  version: 2.0.0
  description: >-
    A generic rate limit policy with flexible property-driven key and limit parameters
  contact:
    name: IBM API Management
    email: ibmapi@us.ibm.com
    url: 'http://www.ibm.com/apimanagement'
gateways:
  - datapower-api-gateway
attach:
  - rest
properties:
  type: object
  properties:
    source:
      label: Source
      description: Where to find the named rate limit.
      type: string
      enum: [plan, catalog]
    name:
      label: Name
      description: >
        The name of a rate-limit object defined in the api-collection. This property is required
        only if source = catalog
      type: string
    weight:
      label: Weight
      description: >
          A weight value assigned to a single execution of this policy (example, POST may be 5, GET 1).
          The default value is 1
      type: number
  required:
    - source
definitions:
  sslprofiletype: {}
```

New Policies – GraphQL (preview)



GraphQL

Title

GraphQL

Description

☑ Populate Server Configuration into Context Variables

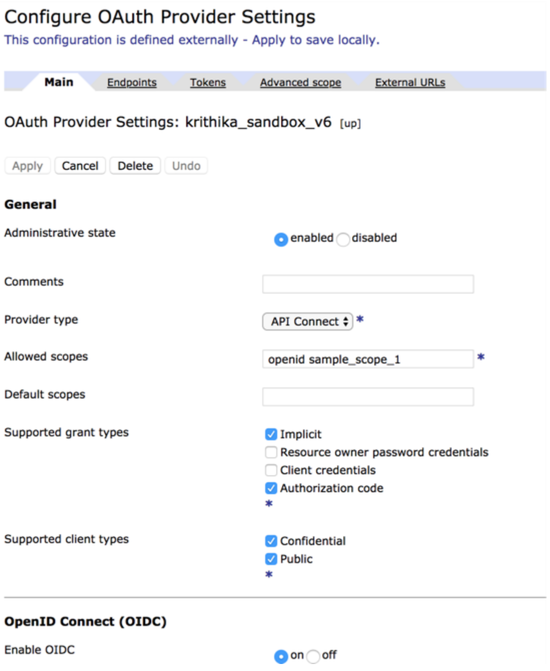
☑ Calculate Query Characteristics

☑ Enforce Policy

☑ Invoke GraphQL Server

Troubleshooting

APIManager



Configure OAuth Provider Settings

This configuration is defined externally - Apply to save locally.

OAuth Provider Settings: krithika_sandbox_v6 [up]

General

Administrative state: ● enabled ○ disabled

Comments:

Provider type: API Connect *

Allowed scopes: openid sample_scope_1 *

Default scopes:

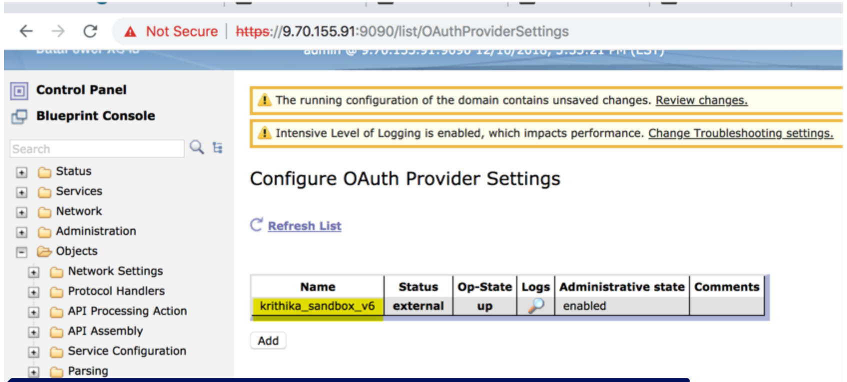
Supported grant types:
☑ Implicit
☐ Resource owner password credentials
☐ Client credentials
☑ Authorization code
*

Supported client types:
☑ Confidential
☑ Public
*

OpenID Connect (OIDC)

Enable OIDC: ● on ○ off

DataPower API Gateway



Not Secure | https://9.70.155.91:9090/list/OAuthProviderSettings

⚠ The running configuration of the domain contains unsaved changes. Review changes.

⚠ Intensive Level of Logging is enabled, which impacts performance. Change Troubleshooting settings.

Configure OAuth Provider Settings

Refresh List

Name	Status	Op-State	Logs	Administrative state	Comments
krithika_sandbox_v6	external	up	🔎	enabled	

Get access to the DataPower API Gateway

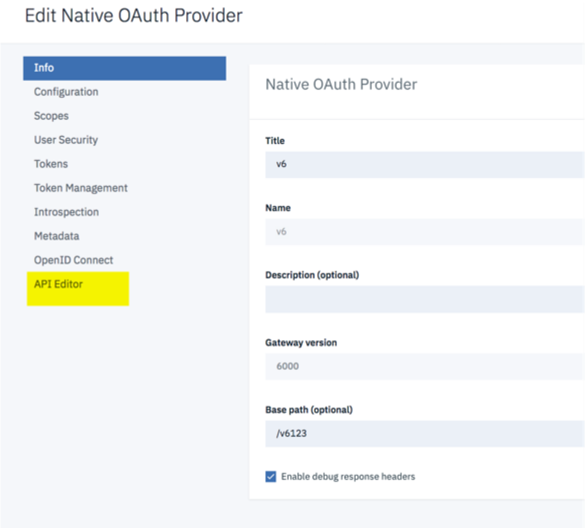
1-1 mapping of OAuth provider objects and API settings in Gateway

Check if OAuth provider settings object got created in Gateway

Troubleshooting - WYSIWYG

What you see in API Manager ….

…. Is what you get in DataPower



Edit Native OAuth Provider

Native OAuth Provider

Title: v6

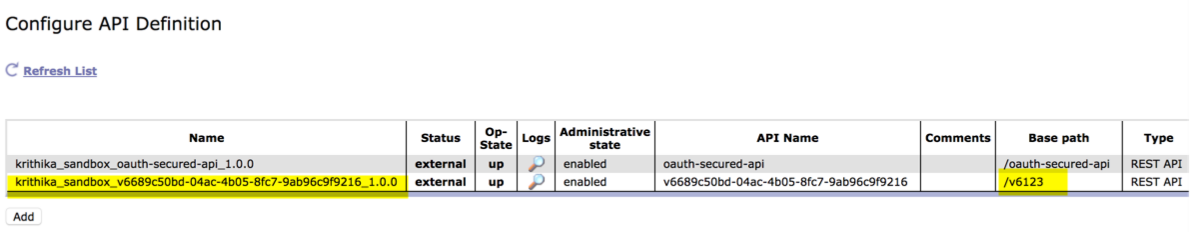
Name: v6

Description (optional):

Gateway version: 6000

Base path (optional): /v6123

☑ Enable debug response headers



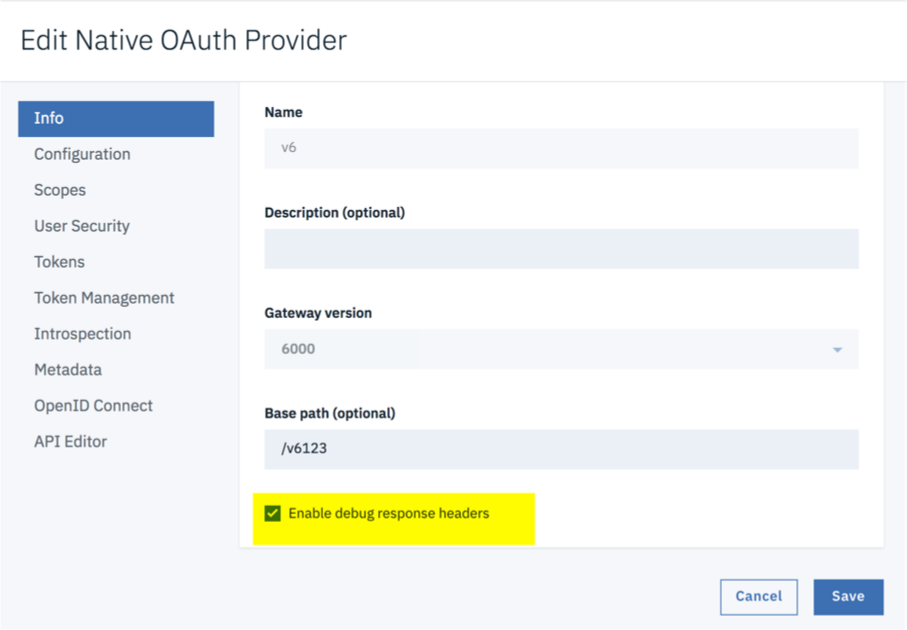
Configure API Definition

Name	Status	Op-State	Logs	Administrative state	API Name	Comments	Base path	Type
krithika_sandbox_oauth-secured-api_1.0.0	**external**	**up**	🔍	enabled	oauth-secured-api		/oauth-secured-api	REST API
krithika_sandbox_v6689c50bd-04ac-4b05-8fc7-9ab96c9f9216_1.0.0	**external**	**up**	🔍	enabled	v6689c50bd-04ac-4b05-8fc7-9ab96c9f9216		/v6123	REST API

- Check if OAuth Provider API is available on APIGateway
- If does not exist, something went wrong with publish. Check gateway service logs

Troubleshooting – Run time

Enable debug response headers on OAuth provider



Edit Native OAuth Provider

Name: v6

Description (optional):

Gateway version: 6000

Base path (optional): /v6123

☑ Enable debug response headers

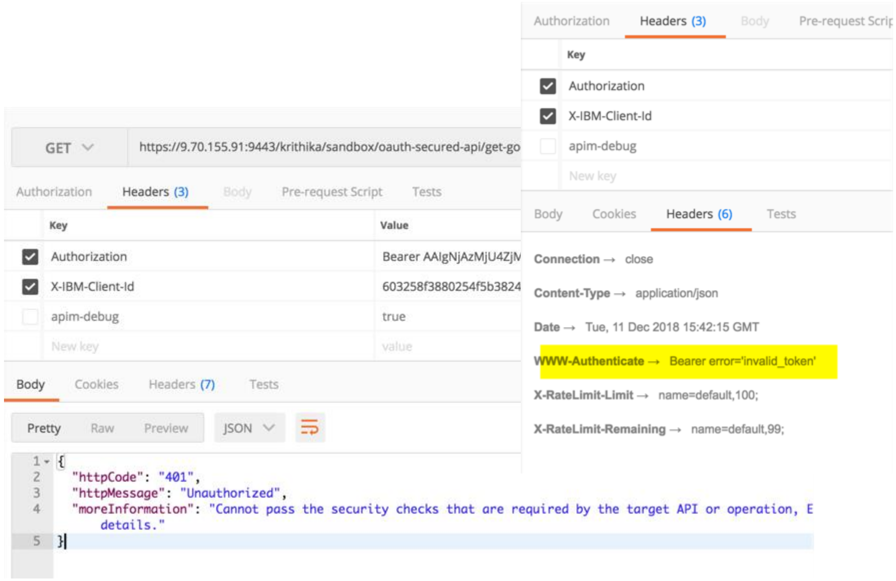
Send request with “api-debug" request header set to true

In the response for OAuth secured APIs, you will see additional error headers

Troubleshooting – Run time

Enable debug response headers on OAuth provider

No “apim-debug=true” header set



GET https://9.70.155.91:9443/krithika/sandbox/oauth-secured-api/get-go

Authorization | Headers (3) | Body | Pre-request Script | Tests

	Key	Value
☑	Authorization	Bearer AAIgNjAzMjU4ZjM
☑	X-IBM-Client-Id	603258f3880254f5b3824
□	apim-debug	true
	New key	value

Body | Cookies | Headers (7) | Tests

Pretty | Raw | Preview | JSON

```
{
  "httpCode": "401",
  "httpMessage": "Unauthorized",
  "moreInformation": "Cannot pass the security checks that are required by the target API or operation, E
     details."
}
```

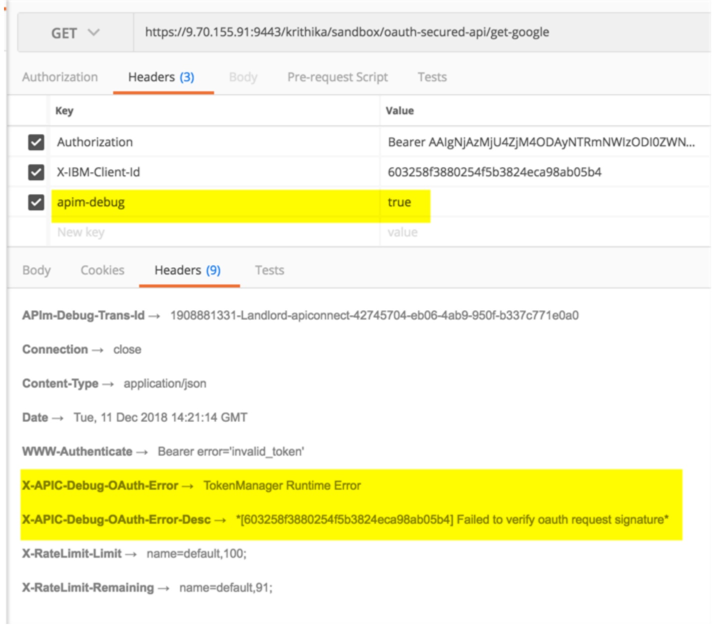
Authorization | Headers (3) | Body | Pre-request Scrip

	Key
☑	Authorization
☑	X-IBM-Client-Id
□	apim-debug
	New key

Body | Cookies | Headers (6) | Tests

```
Connection → close
Content-Type → application/json
Date → Tue, 11 Dec 2018 15:42:15 GMT
WWW-Authenticate → Bearer error='invalid_token'
X-RateLimit-Limit → name=default,100;
X-RateLimit-Remaining → name=default,99;
```

“apim-debug=true” header set



GET https://9.70.155.91:9443/krithika/sandbox/oauth-secured-api/get-google

Authorization | Headers (3) | Body | Pre-request Script | Tests

	Key	Value
☑	Authorization	Bearer AAIgNjAzMjU4ZjM4ODAyNTRmNWIzODI0ZWN...
☑	X-IBM-Client-Id	603258f3880254f5b3824eca98ab05b4
☑	apim-debug	true
	New key	value

Body | Cookies | Headers (9) | Tests

```
APIm-Debug-Trans-Id → 1908881331-Landlord-apiconnect-42745704-eb06-4ab9-950f-b337c771e0a0
Connection → close
Content-Type → application/json
Date → Tue, 11 Dec 2018 14:21:14 GMT
WWW-Authenticate → Bearer error='invalid_token'
X-APIC-Debug-OAuth-Error → TokenManager Runtime Error
X-APIC-Debug-OAuth-Error-Desc → *[603258f3880254f5b3824eca98ab05b4] Failed to verify oauth request signature*
X-RateLimit-Limit → name=default,100;
X-RateLimit-Remaining → name=default,91;
```

Thank You