



Hewlett Packard
Enterprise

NonStop Technical Boot Camp 2023

TBC23-TB61 How to use NonStop API Gateway as the entry point for REST API microservices running on multiple platforms

John Zimsky NonStop Advanced Technology Center

September 2023

Forward-looking statements

This is a rolling (up to three year) Roadmap and is subject to change without notice

This document contains forward looking statements regarding future operations, product development, product capabilities and availability dates. This information is subject to substantial uncertainties and is subject to change at any time without prior notification. Statements contained in this document concerning these matters only reflect Hewlett Packard Enterprise's predictions and / or expectations as of the date of this document and actual results and future plans of Hewlett Packard Enterprise may differ significantly as a result of, among other things, changes in product strategy resulting from technological, internal corporate, market and other changes. This is not a commitment to deliver any material, code or functionality and should not be relied upon in making purchasing decisions.



Agenda

Review of NS API Gateway

Representational State Transfer (REST) microservice

Building the NS API Gateway deployment

Accessing the deployment

Adding additional endpoints off and on platform

Thanks for attending



NonStop Partnership– It's a Beautiful Thing!



NonStop API Gateway

Review of NS API Gateway

- API Gateway is a design pattern that encapsulates the internal system and application architecture and provides a single point of entry into the system. A variety of applications are deployed on NonStop, applications that range from native applications that were developed using DDL defined data structure to the modern applications that conform to established and evolving industry standards. This includes native REST microservices.
- For legacy Pathway applications it provides a way to advertise those applications in the enterprise as REST services
- For native REST microservices it can serve as an entry point for all requests thus limiting the number of ports that need to be exposed.



NonStop API Gateway

Review of NS API Gateway – Prerequisites for running NS API Gateway

- NSJ 11
- NSJ Infrastructure (NSJI)
- NonStop Manageability Elasticity Framework (NSMEF)
- TS/MP 2.8 ACS
- TS/MP Utilities
- Python 3.0
- Python Modules / Toolkit
- OpenSSL
- Middleware Tools



NonStop API Gateway

REST microservice

- We will be using the Spring Boot microservice built as part of the “Build a REpresentational State Transfer (REST) API microservice on NonStop in 15 minutes”
 - Talk TBC23-TB60
- The service provides 2 REST entry points, “/hello” which is a simple GET request and “/doWork” which is a POST request and expects to receive a JSON payload :

```
{  
  "empid" : 1  
}
```

- It returns a JSON payload:

```
{"empid":1,"firstname":"John","lastname":"<platform OS>","departid":1}
```



NonStop API Gateway

Building the NS API Gateway deployment

- If you attended my talk in TBC 2022 on how to build and deploy an NS API Gateway, you will remember there are three sections to a deployment, that remains the same for a deployment of a REST microservice. The sections we will review are:
 1. Global definitions for the Pathway environment along with the IP and port information is defined
 2. Global endpoint definitions for all entry points
 3. Endpoint definitions for each target server
- As these have changed with the new release, we will review all three sections. See the NonStop API Gateway User Guide or the samples in `/usr/tandem/gway/<release>/yml_samples/2.0` for details. For this presentation that is `/usr/tandem/gway/T1199L01_04AUG2023_AAQ`



NonStop API Gateway

Building the NS API Gateway deployment - globals

config_data:

 webservice:

 type: builtin

 logging:

 level: "INFO"

 connection:

 default_ports:

 http: [12030]

 https: [12031]

 network:

 #Please mention the ip address on which the gateway server should bind

 - address: 172.17.199.73

 ports:

 http: [12030]

 https: [12031]

 transport_provider: auto



NonStop API Gateway

Building the NS API Gateway deployment - globals

runtime:

pathmon_name: \$JZMPM

pathway_subvolume: \$OSS.JZMPM

pathctl : GWAYCTL

min_instances: 2

max_instances: 4

max_connections: 250

linkdepth: 2

maxlinks: 8

java:

args: ["-Xmx1024m", "-Xms1024m"]

elasticity:

enable: false

collection_interval: 30

expand_threshold: 70

contract_threshold: 30

advanced: null



NonStop API Gateway

Building the NS API Gateway deployment - globals

endpoint_global_configuration:

context_path: gway

url_pattern: /rest/*

include_custom_directories: None

secure:

csrf: disable

transaction:

mode: NEVER

timeout: 300

encoding:

charset: default



NonStop API Gateway

Building the NS API Gateway deployment - endpoint

```
oss_endpoint:  
  commons:  
    encoding:  
      charset: default  
  ingress:  
    http:  
      path: "/oss"  
      description: "call OSS microservice"  
      post:  
        url: "/doWork/"  
  egress:  
    http:  
      post:  
        # The target URL  
        url: "http://TBC1:12012/doWork"  
      options:  
        connectionsPerRoute: 4
```



NonStop API Gateway

Building the NS API Gateway deployment – create deployment

- Create script:

```
#!/bin/ksh
```

```
./envs.sh
```

```
# purge existing CTL file
```

```
gtacI -c 'purge $OSS.JZMPM.GWAYCTL'
```

```
rm -rf $PWD/gateway
```

```
gwaycli create $PWD/gateway 1.0 $PWD/config/microDeployment.yml
```

- Note the scripts (xxxx.sh) are just written to save typing in the full gwaycli command



NonStop API Gateway

Building the NS API Gateway deployment – create deployment

- Script output:

```
$OSS.JZMPM.GWAYCTL Purged
```

```
Operation in progress...
```

```
{
```

```
  "message": "Gway deployment version 1.0 successfully created on  
/home/hp/johnz/TBC_2023/tbcGWAY/gateway",
```

```
  "result": "success"
```

```
}
```



NonStop API Gateway

Building the NS API Gateway deployment – start deployment

- Start deployment script:

```
#!/bin/ksh
```

```
../envs.sh
```

```
gwaycli start $PWD/gateway $1 (input 1.0 for the initial deployment)
```



NonStop API Gateway

Building the NS API Gateway deployment – access endpoint

- Curl script to access microservice:

```
/usr/bin/curl -X 'POST' \  
"http://TBC1:12030/gway/rest/oss/doWork/" \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d @doWork.json
```

- Note the entry point is /gway/rest/oss/doWork and that is redirected to the REST microservice as configured above with url: "http://TBC1:12012/doWork"
- Output:

```
{"empid":1,"firstname":"John","lastname":"NONSTOP_KERNEL","departid":1}
```



NonStop API Gateway

Building the NS API Gateway deployment – add an endpoint that is off platform

- To add another endpoint you can use the script:

```
gwaycli update $PWD/gateway 1.0 2.0 -a $PWD/config/addLinux.yml
```

- The yml file will contain the definition of the new end point.

Note the target URL is off platform thus allowing you to provide a single entry point to REST microservices running on any platform that is accessible via a TCP/IP connection.



NonStop API Gateway

Building the NS API Gateway deployment – endpoint configuration file

```
version: 2.0.0
config_data:
  endpoints:
    linux_endpoint:
      commons:
        encoding:
          charset: default
      ingress:
        http:
          path: "/linux"
          description: "call Linux microservice"
          post:
            url: "/doWork/"
      egress:
        http:
          post:
            # The target URL in this case on a Linux platform
            url: "http://linux:12012/doWork"
          options:
            connectionsPerRoute: 4
```



NonStop API Gateway

Building the NS API Gateway deployment – access the new endpoint

- Curl script to access microservice:

```
/usr/bin/curl -X 'POST' \  
"http://TBC1:12030/gway/rest/linux/doWork/" \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d @doWork.json
```

- Note the entry point is /gway/rest/linux/doWork and that is redirected to the REST microservice as configured above with url: "http://linux:12012/doWork"
- Output:
{"empid":1,"firstname":"John","lastname":"Linux","departid":1}



NonStop API Gateway

Building the NS API Gateway deployment – add an endpoint that uses NSJI

- NSJI will transform socket applications into applications that can use the file system, for example READUPDATEX_(\$RECEIVE), to do work.
- To add this endpoint you can use the command:

```
gwaycli update $PWD/gateway 2.0 3.0 -a $PWD/config/addJI.yml
```

- In the yml configuration note the change in the target URL to “url: nonstop://doWork” that denotes the connection will be made via a SERVERCLASS_SEND_() request.



NonStop API Gateway

Building the NS API Gateway deployment – endpoint configuration file

```
version: 2.0.0
config_data:
  endpoints:
    ji_endpoint:
      commons:
        encoding:
          charset: default
      ingress:
        http:
          path: "/ji"
          description: "call JI enabled microservice"
          post:
            url: "/doWork/"
    egress:
      http:
        post:
          #Note the URL is "nonstop" and includes a Pathmon and Server Class
          url: nonstop://doWork
          nonstop:
            pathmon: $SBJI
            serverclass: SPRINGBOOT
          options:
            connectionsPerRoute: 4
```



NonStop API Gateway

Building the NS API Gateway deployment – access the new endpoint

- Curl script to access microservice:

```
/usr/bin/curl -X 'POST' \  
"http://TBC1:12030/gway/rest/ji/doWork/" \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d @doWork.json
```

- Note the entry point is /gway/rest/ji/doWork and that is redirected to the REST microservice as configured above with url: " url: nonstop://doWork"

- Output:

```
{"empid":1,"firstname":"John","lastname":"NONSTOP_KERNEL","departid":1}
```



Thank you for attending this talk TBC23-TB61 How to use NonStop API Gateway as the entry point for REST API microservices running on multiple platforms

John Zimsky
John.Zimsky@HPE.com



HPE Slides and Materials Usage

This content is protected

This presentation is the property of Hewlett Packard Enterprise and protected by copyright laws of the United States. The material in this presentation is provided to attendees of the NonStop Technical Boot Camp 2023 as part of their registration and attendance at the event. Attendees are free to use this material and share it with others within their own company.

This material may not be quoted, copied, communicated or shared with third parties or mutual customers without permission from HPE. To request permission to share material in this presentation outside of your company, send an email to mark.pollans@hpe.com explaining the usage you are intending and your request will be considered.

